Modeling and Simulation in Science,
Engineering and Technology

Tayfun E. Tezduyar
Editor

Frontiers in Computational Fluid-Structure Interaction and Flow Simulation

Research from Lead Investigators under Forty – 2018

Birkhäuser

Modeling and Simulation in Science, Engineering and Technology

Series Editors

Editorial Advisory Board

More information about this series at http://www.springer.com/series/4960

Tayfun E. Tezduyar
Editor

Frontiers in Computational Fluid-Structure Interaction and Flow Simulation

Research from Lead Investigators under Forty – 2018

Birkhäuser

Editor
Tayfun E. Tezduyar
Mechanical Engineering
Rice University
Houston, TX, USA

Faculty of Science and Engineering
Waseda University
Shinjuku-ku, Tokyo, Japan

ISSN 2164-3679 ISSN 2164-3725 (electronic)
Modeling and Simulation in Science, Engineering and Technology
ISBN 978-3-319-96468-3 ISBN 978-3-319-96469-0 (eBook)
https://doi.org/10.1007/978-3-319-96469-0

Library of Congress Control Number: 2018958319

Mathematics Subject Classification (2010): 37N10, 74Bxx, 74F10, 76-06, 76Dxx, 76D05, 76F10, 76Nxx

This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Dedicated to Yuri Bazilevs and Kenji Takizawa in celebration of their 40th birthday

Preface

Computational fluid–structure interaction and flow simulation are challenging research areas that bring solution and analysis to many classes of problems in science, engineering, and technology. They have been attractive areas for investigators under 40, with many doing frontier research. This book presents that frontier research. Each chapter has a lead author who is under 40, and that author took the lead role in the work presented in the chapter.

The idea of putting together such a book came from the intention of celebrating the 40th birthday of Yuri Bazilevs and Kenji Takizawa, who will become 40 later this year with a record of outstanding research accomplishments in computational fluid–structure interaction and flow simulation. The book starts with the chapters from Bazilevs and Takizawa and continues with the chapters from the other lead investigators, in alphabetical order. I am confident that the readers will agree that the research presented in all the chapters are impressive and a testament to how much all the lead authors accomplished at such early stages of their careers.

The book is intended to be a valuable resource for early-career readers—not only those interested in computational fluid–structure interaction and flow simulation but also other fields of engineering and science, including fluid mechanics, solid mechanics, and computational mathematics. They will have an opportunity to know more about successful research conducted by researchers under 40 and think about successful research directions and strategies in their own career paths. The book is intended to be a valuable resource also for senior researchers who are interested in knowing more about successful research led by those under 40. They can explore offering collaboration to these researchers under 40 or just enjoy encouraging and supporting the future leaders in these crucial research areas. Graduate students doing or starting research in fluid–structure interaction or flow simulation, and also undergraduate students considering research in these areas, will be able to learn from the book about some of the frontier work. This will help them make better early-stage decisions related to research directions and framework.

This book is a window to some of the state-of-the-art computational methods and analysis in fluid–structure interaction and flow simulation and also a window

to computational analysis of some of the most challenging problems in science, engineering, and technology. It is comforting to know that what we see through the window was all led by researchers under 40, and we expect to see even more from them in the future.

Houston, TX, USA
Tokyo, Japan
May 2018

Tayfun E. Tezduyar

Contents

Simulating Free-Surface FSI and Fatigue Damage in Wind-Turbine Structural Systems

Y. Bazilevs, J. Yan, X. Deng, and A. Korobenko

Abstract This article reviews state-of-the-art numerical techniques for fluid–structure interaction (FSI) of full-scale wind-turbine systems. Simulation of floating wind turbines subjected to combined wind-flow and ocean-wave forcing, and modeling of high-cycle fatigue failure of blades due to long-term cyclic aerodynamic loading are the focal points of this article. Computational techniques including advanced structural modeling based on isogeometric analysis (IGA), free-surface FSI, and fatigue-damage modeling are presented. Representative computational examples involving land-based and floating offshore wind-turbine designs illustrate the versatility and power of the computational methods developed.

1 Introduction

1.1 Offshore Wind and the Need for Advanced Simulation

Offshore wind is an indigenous, clean, and inexhaustible source of energy. It is also an emerging industry, with strong potential for job creation. According to the prediction from National Renewable Energy Laboratory (NREL) in 2010, offshore wind could produce electricity for almost 39 million households by 2020 [51].

Y. Bazilevs (✉)
School of Engineering, Brown University, Providence, RI, USA
e-mail: yuri_bazilevs@brown.edu

J. Yan
Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA

X. Deng
Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong

A. Korobenko
Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, AB, Canada

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_1

This figure could grow even faster beyond 2020, which would require advances in offshore wind-turbine technology.

The current trend in wind energy is to go from land-based to offshore designs, where the latter include wind turbines with bottom-fixed foundation installed along the shoreline in relatively shallow depths, and wind turbines mounted on a floating structure in deeper waters. Compared with land-based wind turbines, the floating offshore wind turbines have the following advantages: (a) The winds are stronger and more constant far from the shore, thus more energy can be generated; (b) The size of turbines is not limited by land transportation, provided the turbines can be assembled along the coastline and safely towed to their operating locations; (c) The visual and noise impact of wind turbines can be avoided due to operation far away from the shore; (d) Vast, open sea/ocean space is available. We believe that performing leading-edge wind energy research and development, which includes advanced fluid–structure interaction (FSI) modeling and simulation, will be essential in order to better exploit the above advantages of offshore wind in the future.

The current practice in simulating floating wind turbines makes use of either steady (time-independent), lumped-parameter aerodynamics and hydrodynamics models that are coupled with scaled down and/or simplified floating-wind-turbine structure models. These models are simple to implement and easy to execute, which makes them attractive, especially since they are routinely used as part of large parametric studies. However, because floating wind turbines are subjected to high wind speeds and violent sea states, and exhibit relatively complex mechanical response to wave loading, including mechanical components in relative motion superposed on elastic deformation of the blades and tower, these simplified models are often unable to adequately describe this behavior, especially in more extreme situations. Nevertheless, it is precisely these more extreme events that cause failures and reduce the remaining useful life of these machines, leading to premature maintenance and repair, and, as a result, to increased cost of offshore wind energy.

1.2 Role of Simulation and Experimental Measurements in Wind-Turbine Damage Prediction

Wind turbines operate for hundreds of millions of cycles during their lifetime. Due to cyclic loading, fatigue damage in blades, as well as in other wind-turbine structural and mechanical components, becomes an important issue. A thorough overview of the early techniques for wind-turbine fatigue-damage estimation may be found in [72], where it was noted that rigorous blade testing in a laboratory environment, as well as in real operating conditions, is necessary to develop realistic fatigue-damage forecasts. However, using experimental and field measurements from sensor data alone is not sufficient to accurately describe the structure damage initiation and progression. This is because sensor arrays typically produce signals from a limited number of spatial points, and, more importantly, the quantities measured do not directly quantify damage.

As the computational methods mature, high-fidelity models based on these methods are starting to be adopted for damage prognosis in large-scale structures [18, 38]. However, computational modeling alone is also not sufficient for making prediction about damage evolution because even the more sophisticated approaches contain many assumptions about geometry, material composition, constitutive modeling, boundary and initial conditions, etc., which do not always reflect physical reality.

Ideally, the computational model should be enriched, as much as possible, with sensor and measurement data that is used to periodically update the model inputs to maintain consistency with the measured data, and, thus, physical reality. Conversely, the appropriately updated computational model is, in turn, able to produce higher-fidelity outputs for the quantities directly linked to structural damage (e.g., the spatial damage-variable distribution), for which direct measurements are not available. This conceptual framework, in which sensor and measurement data for a given physical system co-exist in a symbiotic relationship with an advanced computational model of that system, is referred to as the Dynamic Data-Driven Application System (DDDAS) [35]. As shown in recent work, DDDAS provides a pathway for accurate damage prediction in large-scale composite structures [18], and presents a new, important paradigm in other engineering applications [5, 6, 19, 61].

1.3 Fatigue-Damage Modeling and Integration with FSI and DDDAS

Structures made of laminated composites, such as the wind-turbine blades, exhibit complex mechanical behavior [36] and, when it comes to fatigue-damage modeling, present significant challenges stemming from the multiscale nature of the damage process [66, 73, 74]. The fatigue-damage model, proposed in [64, 65] and presented in this article, falls in the class of continuum damage models (CDMs) for fiber-reinforced composites, in which space- and time-dependent damage variables are evolved to quantitatively predict damage growth as a function of loading cycles [64, 65]. The model was deployed as part of the DDDAS framework to predict failure of a full-scale wind-turbine blade in a laboratory fatigue test in [18].

Using standalone structural mechanics simulations, even in the presence of sensor data, is also insufficient for predicting fatigue-damage growth in wind-turbine blades. As shown in [60], aerodynamic loading and the structure response to that loading contribute significantly to blade fatigue damage. However, it is virtually impossible to rigorously and accurately account for aerodynamic loading in laboratory fatigue tests. To increase the realism of the studies, one may attempt to study fatigue damage "in the field" as the turbines operate. However, wind-turbine blades are built to last hundreds of millions of cycles (here one cycle is assumed to be one full revolution of the rotor), and, unless premature blade failure occurs due to external factors, one needs to wait for upwards of a decade to see significant effects of fatigue damage, which is not practical. On the other hand, the blade fatigue-damage problem may be approached numerically, with the help of DDDAS, and using appropriate coupling of advanced FSI and CDM, which is what we present in this article.

1.4 FSI Modeling Techniques Employed

The FSI framework employed in the present paper was originally developed by the senior author in [12], and successfully applied to the simulation of horizontal- and vertical-axis wind turbines at full-scale in [12, 12, 15, 17, 42, 45, 59]. For the fluid dynamics and turbulence modeling we make use of the variational multiscale (VMS) formulation [10] posed on a moving spatial domain using the Arbitrary Lagrangian–Eulerian (ALE) technique [47] with the addition of weakly enforced no-slip boundary conditions [10, 14, 20] to improve coarse boundary-layer mesh accuracy. This methodology was named ALE-VMS in [76].

It should be noted that VMS-based methods for fluid mechanics showed excellent results in many engineering applications involving turbulent flows, complex geometry, and moving domains—see references [13, 18, 21, 22, 24, 25, 52, 57, 75, 77, 78, 86–92, 95] for the examples of challenging computations performed using the VMS methodology in the context of ALE, space–time (ST), and fixed-grid techniques.

For FSI modeling of the floating offshore wind turbines, the level-set method [2–4, 53, 62, 63] is adopted to track the evolution of the free surface. The aerodynamics and hydrodynamics are governed by the Navier–Stokes equations of incompressible two-fluid flow, in which the fluid density and viscosity are evaluated with the aid of a level-set function.

The structural mechanics of wind turbines is modeled using isogeometric analysis (IGA) [30, 46], which is beneficial for the present application from the standpoint of both geometry modeling and accuracy. Since its conception, IGA has been widely used in many areas of computational mechanics, engineering, and sciences, showing improved performance over the standard FEM [30]. The main structural components of on-land and offshore wind turbines, including blades, rotor, nacelle, tower, and platform, are modeled as Kirchhoff–Love shells [26, 37, 54, 55] with the aid of the bending strip method [54], while the main shaft and mooring lines are modeled using the recently introduced rotation-free beam/cable formulation [67]. The IGA shell is used in the modeling of the wind-turbine composite blades, both as part of the FSI and fatigue-damage evolution simulations. For recent development in, and applications of, IGA, the reader is referred to [7, 27, 39–41, 43, 56, 67, 71, 110–112].

The FSI methodology presented in this article assumes a non-matching discretization at the fluid–structure interface, which is handled by means of a recently developed augmented Lagrangian FSI formulation with formal elimination of the Lagrange multiplier variable [12]. In the FSI formulation, the fluid–structure interface is tracked by the moving fluid mechanics mesh, while for the floating offshore wind turbines the air–water interface is captured (i.e., not tracked) on that mesh. In that sense, the method falls in the class of Mixed Interface-Tracking/Interface-Capturing Techniques (MITICT) [99, 102], which were primarily introduced for fluid–object interaction (FOI) with multiple fluids [81, 82]. To handle the added mass effect, which presents a challenge in the case of floating turbines, quasi-direct FSI coupling strategy [23, 79, 83, 85, 99, 104, 105, 107] is employed to solve the discrete FSI equations at each nonlinear iteration within a time step.

1.5 Outline

Governing equations of free-surface flows at the continuum and discrete levels are presented in Sects. 2 and 3, respectively. IGA-based structural mechanics formulation including a fatigue-damage model is presented in Sect. 4. Methods for time integration and FSI coupling for free-surface flows, and for the case of fatigue damage, are discussed in Sect. 5. FSI simulations employing the methods developed and reviewed in this article are shown in Sect. 6. In Sect. 7 we draw conclusions.

2 Continuum Formulation of Free-Surface Flows

In this section, the governing differential equations of free-surface flow on a moving domain are summarized. Let $\Omega_t \in \mathbb{R}^d, d = 2, 3$ denote the combined air–water domain at time t and let Γ_t denote its boundary. The domain Ω_t is decomposed into the water and air subdomains, denoted by Ω_t^w and Ω_t^a, respectively, and Γ_t^{aw} denotes the air–water interface. (See Fig. 1 for an illustration.)

In the present work, the level-set method is adopted to capture the air–water interface. For this, we introduce a scalar function $\phi(\mathbf{x}, t)$ and define the subdomains and their interface as follows:

$$\Omega_t^a = \{\mathbf{x} \mid \phi(\mathbf{x}, t) < 0, \forall \mathbf{x} \in \Omega_t\}, \tag{1}$$

$$\Omega_t^w = \{\mathbf{x} \mid \phi(\mathbf{x}, t) > 0, \forall \mathbf{x} \in \Omega_t\}, \tag{2}$$

$$\Gamma_t^{aw} = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0, \forall \mathbf{x} \in \Omega_t\}. \tag{3}$$

In each subdomain, the fluid density ρ and viscosity μ are defined as

$$\rho = \rho_w H(\phi) + \rho_a (1 - H(\phi)), \tag{4}$$

$$\mu = \mu_w H(\phi) + \mu_a (1 - H(\phi)), \tag{5}$$

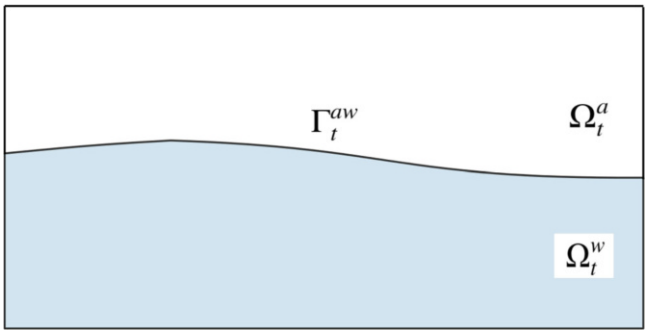


Fig. 1 Fluid mechanics spatial domain with air and water subdomains, and interface between them

where ρ_a and ρ_w are the constant densities of air and water, respectively, μ_a and μ_w are the constant viscosities of air and water, respectively, and $H(\phi)$ is the Heaviside function expressed as

$$H(\phi) = \begin{cases} 0 & \text{if } \phi \leq 0, \\ 1/2 & \text{if } \phi = 0, \\ 1 & \text{if } \phi > 0. \end{cases} \tag{6}$$

With the fluid properties defined as above, the Navier–Stokes equations of incompressible two-fluid flow in the ALE description [47] may be written as follows:

$$\rho \left(\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\hat{x}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \tag{7}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{8}$$

where the Cauchy stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu \nabla^s \mathbf{u}, \tag{9}$$

the fluid velocity and pressure are denoted by $\mathbf{u}$ and p, respectively, $\mathbf{f}$ is the body force per unit mass, $\hat{\mathbf{u}}$ is the velocity of the fluid domain, ∇^s is the symmetric gradient operator, and $|_{\hat{x}}$ is used to denote the fact that the time derivative is taken with respect to a fixed referential domain (see [23] for more details).

The air–water interface is assumed to be convected by the flow, which is modeled by means of an additional convection equation of the level-set function ϕ posed on a moving domain using the ALE description as follows:

$$\left. \frac{\partial \phi}{\partial t} \right|_{\hat{x}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \phi = 0. \tag{10}$$

The above equations, together with the suitably chosen initial and boundary conditions, constitute the free-surface flow formulation on a moving domain at the continuous level.

3 Discrete Formulation of Free-Surface Flows

In this section we present the space-discrete version of the free-surface flow equations. We adopt the ALE-VMS formulation [13, 76–78], which has been applied to a variety of challenging fluid mechanics and FSI problems in [8, 15, 17, 18, 58, 59, 113], and also presented in the context of free-surface flow in [2–4, 53].

3.1 ALE-VMS Formulation of Free-Surface Flow

Let $\mathscr{V}^h$ denote the discrete trial space for the velocity-pressure-level set triple $\{\mathbf{u}^h, p^h, \phi^h\}$, and let $\mathscr{W}^h$ denote the discrete test space for the linear-momentum, continuity and level-set equations $\{\mathbf{w}^h, q^h, \eta^h\}$. The ALE-VMS formulation is stated as follows: Find $\{\mathbf{u}^h, p^h, \phi^h\} \in \mathscr{V}^h$, such that $\forall \{\mathbf{w}^h, q^h, \eta^h\} \in \mathscr{W}^h$:

$$
\begin{aligned}
&\int_{\Omega_t} \mathbf{w}^h \cdot \rho \left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega + \int_{\Omega_t} \boldsymbol{\nabla} \mathbf{w}^h : \boldsymbol{\sigma}(\mathbf{u}^h, p^h)\, \mathrm{d}\Omega \\
&\quad - \int_{\Gamma_t^h} \mathbf{w}^h \cdot \mathbf{h}\, \mathrm{d}\Gamma + \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \tau_M \left((\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \boldsymbol{\nabla} \mathbf{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho} \right) \cdot \mathbf{r}_M(\mathbf{u}^h, p^h)\, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \rho \tau_C \boldsymbol{\nabla} \cdot \mathbf{w}^h r_C(\mathbf{u}^h, p^h)\, \mathrm{d}\Omega \\
&\quad - \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \tau_M \mathbf{w}^h \cdot \left(\mathbf{r}_M(\mathbf{u}^h, p^h) \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \mathrm{d}\Omega \\
&\quad - \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \frac{\boldsymbol{\nabla} \mathbf{w}^h}{\rho} : \left(\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h) \right) \otimes \left(\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h) \right) \mathrm{d}\Omega \\
&\quad + \int_{\Omega_t} \eta^h \left(\left.\frac{\partial \phi^h}{\partial t}\right|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \boldsymbol{\nabla} \phi^h \right) \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \tau_\phi (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \boldsymbol{\nabla} \eta^h \left(\left.\frac{\partial \phi^h}{\partial t}\right|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \boldsymbol{\nabla} \phi^h \right) \mathrm{d}\Omega = 0. \qquad (11)
\end{aligned}
$$

In Eq. (11), n_{el} is the number of elements in the domain, $\mathbf{r}_M(\mathbf{u}^h, p^h)$ and $r_C(\mathbf{u}^h, p^h)$ are element-interior residuals of the strong-form momentum and continuity equations, and τ_M, τ_C, and τ_ϕ are the stabilization parameters [28, 44, 92, 94, 98, 100, 104].

To alleviate the high computational costs of detailed resolution of viscous turbulent boundary layers near solid surfaces without sacrificing the accuracy of hydrodynamic loading on structures, the ALE-VMS formulation of the level-set equations is augmented with weakly enforced no-slip boundary conditions. In this case, the following terms are added to the left-hand side of Eq. (11):

$$
\begin{aligned}
&- \sum_{b=1}^{n_{eb}} \int_{\Gamma_t^b \cap \Gamma_t^g} \mathbf{w}^h \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h)\mathbf{n} \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{n_{eb}} \int_{\Gamma_t^b \cap \Gamma_t^g} \left(2\mu \boldsymbol{\nabla}^s \mathbf{w}^h \mathbf{n} + q^h \mathbf{n} \right) \cdot (\mathbf{u}^h - \mathbf{g}^h) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{n_{eb}} \int_{\Gamma_t^b \cap (\Gamma_t^g)^-} \mathbf{w}^h \cdot \rho((\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \mathbf{n})(\mathbf{u}^h - \mathbf{g}^h) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{n_{eb}} \int_{\Gamma_t^b \cap \Gamma_t^g} \mathbf{w}^h \cdot \tau_B (\mathbf{u}^h - \mathbf{g}^h) \, \mathrm{d}\Gamma. \qquad (12)
\end{aligned}
$$

In Eq. (12), $\mathbf{g}^h$ is the prescribed fluid velocity on the no-slip moving boundary Γ_t^g, which decomposed into n_{eb} surface elements denoted by Γ_t^b, $(\Gamma_t^g)^-$ denotes the inflow part of Γ_t^g and τ_B is a boundary stabilization parameter. See [14, 20] for more details on weakly enforced essential boundary conditions, as well as more recent work in [93, 95] on weakly enforced essential boundary conditions in the context of space–time methods.

3.2 Additional Level-Set Computational Technology

In discrete setting, the fluid density and viscosity are computed as

$$\rho = \rho_w H_\epsilon(\phi^h) + \rho_a(1 - H_\epsilon(\phi^h)), \qquad (13)$$

$$\mu = \mu_w H_\epsilon(\phi^h) + \mu_a(1 - H_\epsilon(\phi^h)). \qquad (14)$$

Here $H_\epsilon(\phi)$ is a regularized version of the Heaviside function, namely,

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \sin\left(\frac{\phi\pi}{\epsilon}\right)\right) & \text{if } |\phi| \le \epsilon, \\ 1 & \text{if } \phi > \epsilon, \end{cases} \qquad (15)$$

where ϵ, assumed to scale with the local mesh size h, defines the interface width between the air and water subdomains.

While the regularized Heaviside function in Eq. (15) gives a smooth transition from zero to unity within a small band of elements around the interface, and is numerically more favorable to the sharp discontinuity, this regularization places a requirement on the level-set field ϕ^h to satisfy the so-called signed-distance property in the transition layer between the two fluids. For this, we define an additional field, ϕ_d^h, which satisfies the Eikonal partial differential equation, namely,

$$\left|\left|\boldsymbol{\nabla}_x \phi_d^h\right|\right| = 1 \text{ in } \Omega_t, \tag{16}$$

subject to the constraint that the interface defined by the zero level set of ϕ^h is preserved, namely,

$$\phi_d^h = \phi^h = 0 \text{ on } \Gamma_t^{aw} \tag{17}$$

In order to satisfy Eqs. (16) and (17), we make the Eikonal equation "pseudo-time"-dependent (we denote pseudo-time by $\tilde{t}$), discretize it using a VMS technique, and add a suitably constructed penalty term to enforce the interior constraint on ϕ_d^h given by Eq. (17). The resulting semi-discrete form of the governing equations may be stated as: Given ϕ^h, find ϕ_d^h, such that, $\forall \eta_d^h$,

$$\begin{aligned}
&\int_{\Omega_t} \eta_d^h \left(\frac{\partial \phi_d^h}{\partial \tilde{t}} + \mathbf{a} \cdot \boldsymbol{\nabla} \phi_d^h - S_\epsilon(\phi^h) \right) \mathrm{d}\Omega \\
&\quad + \int_{\Omega_t} \tau_{\phi_d} \mathbf{a} \cdot \boldsymbol{\nabla} \eta_d^h \left(\frac{\partial \phi_d^h}{\partial \tilde{t}} + \mathbf{a} \cdot \boldsymbol{\nabla} \phi_d^h - S_\epsilon(\phi^h) \right) \mathrm{d}\Omega \\
&\quad + \int_{\Omega_t} \eta_d^h \lambda_{pen} \frac{dH_\epsilon(\phi^h)}{d\phi^h} (\phi_d^h - \phi^h) \, \mathrm{d}\Omega = 0.
\end{aligned} \tag{18}$$

Here, $S_\epsilon(\phi^h) = 2H_\epsilon(\phi^h) - 1$ is the regularized sign function, $\mathbf{a} = S_\epsilon(\phi^h) \boldsymbol{\nabla} \phi_d^h \Big/ \left|\left|\boldsymbol{\nabla} \phi_d^h\right|\right|$ is the effective "convective" velocity, τ_{ϕ_d} is the stabilization parameter, and λ_{pen} is the interface penalty parameter (see [3] for details). At each time step, Eq. (18) is integrated in pseudo-time, which gives a new level-set field ϕ_d^h with the signed-distance property and zero level set coincident with that of ϕ^h. After this "re-distancing" process is done, we set $\phi^h = \phi_d^h$ at the end of the time step.

Remark It is important to note that $\frac{dH_\epsilon(\phi^h)}{d\phi^h}$ in Eq. (18) is only nonzero in a band of elements around the air–water interface, and thus the penalty term is only active near the air–water interface, which is the desired construction. Also note that the presence of $\frac{dH_\epsilon(\phi^h)}{d\phi^h}$ produces the correct scaling of the penalty term and makes the penalty parameter λ_{pen} independent of the mesh size.

Remark Level set convection and re-distancing introduce mass balance errors, which accumulate as the equations are integrated in time. As a result, in the computations, a simple procedure is employed where the resultant re-distanced level-set field is perturbed by a constant to restore the mass balance. See [3] for details as well as additional references [1, 31–33, 53, 99, 101] where other mass correction techniques were proposed for free-surface flows.

4 Structural Mechanics Formulation

The structural mechanics formulation is based on the principle of virtual work. Let $\mathscr{V}_s$ and $\mathscr{W}_s$ denote the trial and test function sets for the structural mechanics problem. The principle of virtual work leads to the weak form of the structural mechanics problem, which may be stated as follows: Find $\mathbf{d} \in \mathscr{V}_s$, such that, $\forall \mathbf{w}_s \in \mathscr{W}_s$,

$$\int_{\Omega_0^s} \mathbf{w}_s \cdot \rho_s \left(\frac{d^2\mathbf{d}}{dt^2} - \mathbf{f}_s \right) \mathrm{d}\Omega + \int_{\Omega_0^s} \delta\mathbf{E} : \mathbf{S} \, \mathrm{d}\Omega - \int_{\Gamma_0^s} \mathbf{w}_s \cdot \mathbf{h}_s \, \mathrm{d}\Gamma = 0, \tag{19}$$

where $\mathbf{d}$ is the structural displacement, ρ_s is the density, $\mathbf{E}$ is the Green–Lagrange strain tensor, $\delta\mathbf{E}$ is its variation, $\mathbf{S}$ is the second Piola–Kirchhoff stress, $\mathbf{f}_s$ is the body force per unit mass, and $\mathbf{h}_s$ is the applied traction.

Since the structures simulated in this paper are essentially thin shells, we introduce the thin shell kinematics into the above weak formulation. Furthermore, in order to simulate laminated composite structures, we assume there are multiple plies through the shell thickness, each modeled as the St. Venant–Kirchhoff material. As a result, we obtain a Kirchhoff–Love shell formulation [55], which is written purely in terms of the shell midsurface displacements (i.e., the formulation is "rotation-free"), and which we discretize using IGA based on Non-Uniform Rational B-Splines (NURBS) [30, 46]. The weak formulation of the rotation-free isogeometric shell can be stated as: Find the shell midsurface displacement $\mathbf{d}^h \in \mathscr{V}_s^h$, such that, $\forall \mathbf{w}_s^h \in \mathscr{W}_s^h$,

$$\begin{aligned}
&\int_{\Gamma_t^s} \mathbf{w}_s^h \cdot \overline{\rho}_s h_{th} \left(\frac{d^2\mathbf{d}^h}{dt^2} - \mathbf{f}_s \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\,\overline{\boldsymbol{\epsilon}}^h \cdot \left(\mathbf{K}_{exte} \overline{\boldsymbol{\epsilon}}^h + \mathbf{K}_{coup} \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^s} \delta\,\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{K}_{coup} \overline{\boldsymbol{\epsilon}}^h + \mathbf{K}_{bend} \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\
&+ \int_{\Gamma_0^b} \delta\,\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{K}_{bend}^b \overline{\boldsymbol{\kappa}}^h \, \mathrm{d}\Gamma \\
&- \int_{\Gamma_0^s} \mathbf{w}_s^h \cdot \mathbf{h}_s \, \mathrm{d}\Gamma = 0.
\end{aligned} \tag{20}$$

In the above, $\mathscr{V}_s^h$ and $\mathscr{W}_s^h$ denote the trial and test function sets for the isogeometric shell formulation, Γ_0^s and Γ_t^s denote the shell midsurface in reference and current configuration, respectively, h_{th} is the local shell thickness, $\overline{\boldsymbol{\epsilon}}^h$ and $\overline{\boldsymbol{\kappa}}^h$ are the tensors of shell membrane strains and curvature changes expressed in Voigt notation and written with respect to the local basis oriented on the first covariant basis vector of the shell midsurface, $\delta\,\overline{\boldsymbol{\epsilon}}^h$ and $\delta\,\overline{\boldsymbol{\kappa}}^h$ are the corresponding variations.

Matrices $\mathbf{K}_{exte}$, $\mathbf{K}_{coup}$, and $\mathbf{K}_{bend}$ are the extensional, coupling, and bending stiffnesses, respectively, which, using the classical laminated plate theory [68], may be computed as follows:

$$\mathbf{K}_{exte} = \int_{h_{th}} \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k t_k, \tag{21}$$

$$\mathbf{K}_{coup} = \int_{h_{th}} \xi_3 \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k t_k \overline{z_k}, \tag{22}$$

$$\mathbf{K}_{bend} = \int_{h_{th}} {\xi_3}^2 \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k \left(t_k \overline{z_k^2} + \frac{t_k^3}{12} \right). \tag{23}$$

Here, ξ_3 is the through-thickness coordinate, t_k denotes the thickness of the kth ply, $\overline{z}_k$ denotes its centroid, and $\overline{\mathbf{C}}_k$ is a constitutive material matrix for the kth ply in the local coordinate system computed as

$$\overline{\mathbf{C}}_k = \mathbf{T}^T(\phi_k)\, \tilde{\mathbf{C}}_k\, \mathbf{T}(\phi_k), \tag{24}$$

$$\mathbf{T}(\phi_k) = \begin{bmatrix} \cos^2\phi_k & \sin^2\phi_k & \sin\phi_k \cos\phi_k \\ \sin^2\phi_k & \cos^2\phi_k & -\sin\phi_k \cos\phi_k \\ -2\sin\phi_k \cos\phi_k & 2\sin\phi_k \cos\phi_k & \cos^2\phi_k - \sin^2\phi_k \end{bmatrix}, \tag{25}$$

where $\mathbf{T}(\phi_k)$ is a transformation matrix defined by the fiber orientation angle in the ply ϕ_k, and $\tilde{\mathbf{C}}_k$ is the constitutive matrix for the orthotropic material written with respect to the principal material axes (or lamina axes) of the ply.

To enable modeling of structures comprised of multiple NURBS patches, and with regions of reduced continuity (e.g., sharp edges or non-manifold surfaces), the bending-strip technique is employed [54], which is expressed by means of the second-to-last term on the left-hand side of Eq. (20). Here, Γ_0^b and $\mathbf{K}_{bend}^b$ are the bending-strip domain in the shell reference configuration and the directional bending stiffness, respectively.

The shell model is augmented with a newly proposed isogeometric rotation-free bending-stabilized cable formulation [67], which may also be obtained by introducing the appropriate kinematics into the virtual-work principle given by Eq. (19), and reducing the formulation to the cable middle-curve displacement. The resulting weak formulation may be stated as: Find the middle curve displacement $\mathbf{d}^h$, such that, $\forall \mathbf{w}_c^h$,

$$\begin{aligned} &\int_{S_0} \mathbf{w}_c^h \cdot \overline{\rho}_s A_0 \left(\frac{d^2\mathbf{d}^h}{dt^2} - \mathbf{f}_s \right) \mathrm{d}S \\ &\quad + \int_{S_0} \delta\, \overline{\epsilon}_c^h [E_c A_0 \left|\left|\mathbf{G}'\right|\right|^4] \overline{\epsilon}_c^h \, \mathrm{d}S \\ &\quad + \int_{S_0} \delta\, \overline{\kappa}_c^h [E_c I_0 \left|\left|\mathbf{G}'\right|\right|^4] \overline{\kappa}_c^h \, \mathrm{d}S = 0 \end{aligned} \tag{26}$$

In the above, S_0 is the middle curve in reference configuration, A_0 and I_0 are the cable cross-section area and area moment of inertia, respectively, $\overline{\epsilon}_c^h$ and $\overline{\kappa}_c^h$ are the membrane strains and curvature changes written with respect to the basis vector oriented on the tangent vector to the middle curve, denoted by $\mathbf{G}'$, $\delta\overline{\epsilon}_c^h$ and $\delta\overline{\kappa}_c^h$ are the corresponding variations, and E_c is the Young's modulus.

4.1 Fatigue-Damage Model

The fatigue-damage model, proposed in [64, 65] and adapted to IGA-based Kirchhoff–Love shells in [18] is briefly recalled here. The ply-level constitutive matrix $\tilde{\mathbf{C}}_k$ takes on the following form:

$$\tilde{\mathbf{C}}_k = \begin{bmatrix} \frac{E_1(1-D_{11})}{1-\nu_{21}\nu_{12}} & \frac{\nu_{21}E_1\sqrt{(1-D_{11})(1-D_{22})}}{1-\nu_{21}\nu_{12}} & 0 \\ \frac{\nu_{12}E_2\sqrt{(1-D_{11})(1-D_{22})}}{1-\nu_{21}\nu_{12}} & \frac{E_2(1-D_{22})}{1-\nu_{21}\nu_{12}} & 0 \\ 0 & 0 & \frac{G_{12}(1-D_{12})}{1-\nu_{21}\nu_{12}} \end{bmatrix}, \tag{27}$$

where ν's are the Poisson ratios, E_1 and E_2 are the Young's moduli in the fiber and matrix directions, respectively, G_{12} is the shear modulus, and D_{11}, D_{22}, and D_{12} are, respectively, the fiber, matrix, and shear damage indices expressed as

$$D_{ij} = d_{ij}^t + d_{ij}^c \quad (i, j = 1, 2), \tag{28}$$

where d_{ij}^t and d_{ij}^c are the corresponding damage variables with the superscripts 't' and 'c' used to distinguish between tensile and compressive damage modes. For the case of high-cycle fatigue, assuming no damage growth occurs within the cycle, the evolution law for the damage variables is defined in terms of damage growth rate per cycle as

$$\frac{d(d_{ij}^{t,c})}{dN} = f\left(c_k, d_{ij}^{t,c}, D_{12}, \Sigma_{ij}\right) \quad (i, j = 1, 2), \tag{29}$$

where N denotes the cycle number, and reference [18] provides the detailed expressions for f. The damage growth rate is, in particular, an empirical function of material parameters c_k (total of nine parameters for fully reversible cyclic loading) that govern damage initiation and propagation, and failure indices

$$\Sigma_{ij} = \frac{\Sigma_{ij}^{2D}}{1 + (\Sigma_{ij}^{2D} - \Sigma_{ij}^{1D})} \quad (i, j = 1, 2), \tag{30}$$

which can be seen as a combination of indices based on decoupled longitudinal, transverse, and shear failure modes (Σ_{ij}^{1D}'s), and indices based on the Tsai–Wu failure surface (Σ_{ij}^{2D}'s) wherein the failure modes are coupled [34].

5 FSI Coupling and Time Integration

5.1 Framework for Free-Surface Flow

The free-surface flow and IGA-based structural mechanics formulations are coupled as follows. In Eq. (12), which details the weak no-slip boundary conditions for free-surface flow, the prescribed solid-wall velocity $\mathbf{g}^h$ is replaced by the unknown structural displacement rate $d\mathbf{d}^h/dt$ computed from the shell formulation given by Eq. (20). At the same time, the prescribed traction $\mathbf{h}_s$ in Eq. (20) is replaced with the unknown traction field computed from the free-surface flow equations. Conservative fluid traction that is consistent with the weak enforcement of no-slip boundary conditions (see [9] for a definition) is employed in this work. Additional fluid-traction accuracy may be gained by employing a recently developed separated stress projection (SSP) technique [80, 84, 96, 106, 108], where the pressure is projected as a scalar and viscous traction is projected as a vector quantity.

Application of the Generalized-α time integration technique [11, 29, 48] to the coupled free-surface FSI formulation leads to a coupled, nonlinear equation systems that need to be solved at every time step. To solve the nonlinear equation system, we employ the Newton–Raphson method, which requires solving a large linear-equation system that couples the different components of the free-surface FSI formulation. To increase the computational efficiency, we separate the linear system into "physics" and "mesh" subsystems, and solve them sequentially as follows. Increments of the fluid, level-set, and structure solution are obtained by solving a reduced linear system that couples these three fields. The new structure displacement is then used to update the fluid mechanics mesh configuration. The mesh deformation is governed by the equations of elastostatics with jacobian-based stiffening [49, 97, 99, 103, 104] to preserve the mesh quality. The overall approach may be classified as a quasi-direct FSI coupling technique [79, 83, 85, 99, 104, 105, 107], which is used here to handle the large fluid added mass present in the application.

In quasi-direct coupling, the off-diagonal tangent matrices are needed to solve the coupled linear system. In order to avoid assembling the off-diagonal terms, a flexible-GMRES (FGMRES) technique [69] with block-preconditioning is adopted. Application of the FGMRES technique to the solution of the coupled equation system requires computation of matrix-vector products, which are performed using a matrix-free technique. To precondition the FGMRES, we make use of the left-hand-side matrices of the individual fluid, level-set, and structure problems. The linear systems associated with preconditioning of the fluid and level-set equations are solved using a diagonally scaled GMRES technique [70], while the structure problem is solved using a diagonally scaled conjugate gradient method. Other, more sophisticated preconditioning options may also be considered (see, e.g., [23]).

Remark While level-set convection is performed inside the Newton-iteration loop, level-set re-distancing and mass balance are performed once per time step. This is done from considerations of computational efficiency.

Remark We assume no coupling between the cables and surrounding fluid. Although the cable cross-section is relatively small in the applications simulated, hydrodynamic loading on cables may be an important factor in the overall dynamics of the floating-wind-turbine structures considered in this article. For this, a simple modification to the FSI formulation may consist in assuming one-way dependence, as in [109], wherein the cable is subjected to fluid drag forces, while the fluid does not "feel" the presence of the cable.

5.2 Algorithms for FSI Coupling with Fatigue Damage

Since we are interested in high-cycle fatigue for wind-turbine blades, and use a CDM formulation that assumes that damage growth occurs on a time scale that is much slower than a single rotor revolution, we feel it is sufficient to use a "frozen" damage state during FSI computations of the spinning rotor. This motivated the following overall algorithm:

1. We integrate the damage evolution law given by Eq. (29) using an explicit Euler method. Damage evolution is computed *at every Gaussian quadrature point of the blade surface and at each composite ply*.
   To enhance the computational efficiency, a "cycle-jump" technique is employed. Here the stress state that drives the damage model is obtained by solving the coupled FSI system every NJUMP cycles of the damage evolution, where NJUMP is a user-controlled parameter.
2. Every time a new stress state is needed to integrate the damage-evolution equations, we perform a full dynamic FSI simulation for a time interval corresponding to a full (or partial) loading cycle, and extract the required stress data. The FSI equations are integrated in time using the generalized-α method. As was reported in prior work on wind turbines, in the absence of free-surface flow, block-iterative FSI coupling strategy [23, 99, 104, 105, 107] is the most efficient choice for the present application [15].

While using the cycle-jump technique makes the problem computable, further reduction in computational time may be achieved by making the following observation: One rotor revolution may be decomposed into three 120° segments (see Fig. 2). As Blade 1 moves through zone I, Blade 2 moves through zone II, and Blade III moves through zone III. Because gravity and rotor–stator interaction are taken into account, loads experienced by the blades in each zone differ in a significant manner. Nevertheless, the three blades moving through their respective zones collectively experience the same loading as a single blade undergoing a full revolution. As a result, we perform the FSI computation of a full machine with three blades for only 1/3 of a revolution every NJUMP cycles. However, when collecting blade-stress data to integrate the damage-evolution equations, time history of the stress from all three blades is employed.

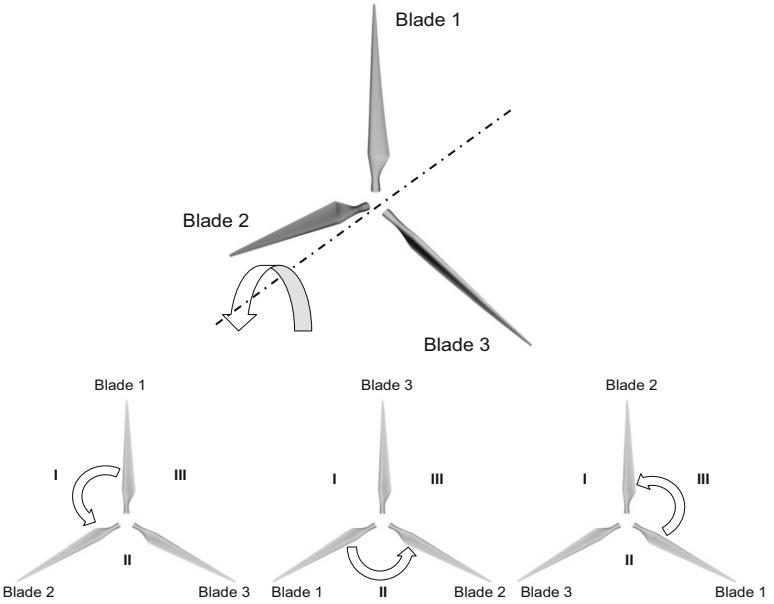


Fig. 2 Decomposition of the rotor motion into three 120° segments. Simulation of the full machine with three blades for only 1/3 of the revolution is equivalent to simulating a single blade for a full revolution, from the standpoint of obtaining a full blade stress time history for driving the fatigue-damage model

Remark In the FSI calculations, the initial conditions chosen are often such that the coupled system undergoes unphysical transients, which eventually settle to produce a physical time-dependent response. In the present computations, in order to preclude this unphysical transient response from affecting the blade-damage evolution, in the beginning of the simulation, as well as after each NJUMP cycle, the FSI simulation is carried out for 1/3 of the revolution to ensure that the undesired transients settle.

6 Applications

6.1 Free-Surface FSI Simulations of an Offshore Floating Wind Turbine

In this section, we present a free-surface FSI simulation of an offshore floating wind turbine. The whole floating wind turbine consists of a supporting spar buoy called "Hywind," developed by Statoil, and the NREL 5 MW baseline turbine. This design

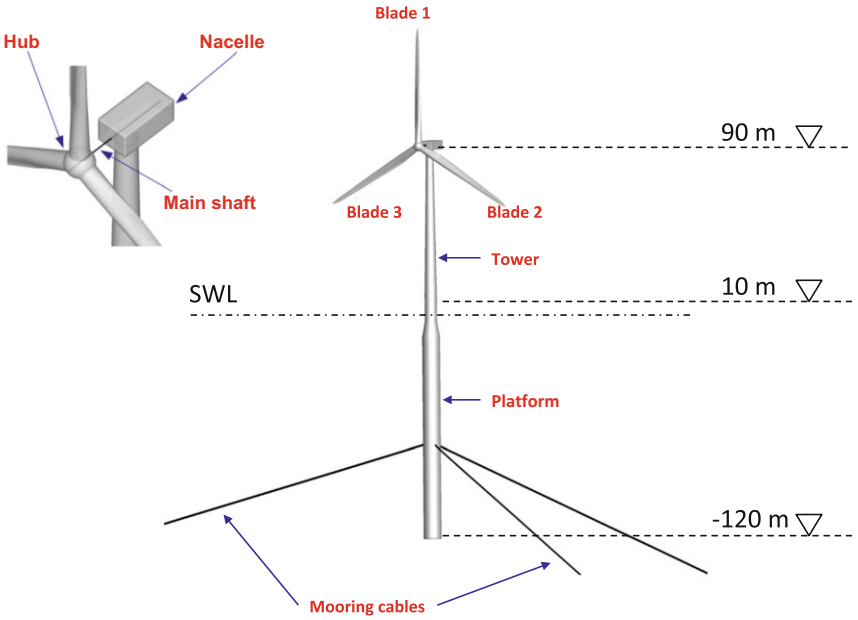


Fig. 3 Structural model of the offshore floating wind turbine with zoom on the rotor

concept was chosen for its suitability for modeling and existence of a full-scale prototype [50, 51]. The geometry of the complete floating structure is defined as follows. The rotor has a diameter of 126 m. The base of the tower is located at an elevation of 10 m above the still water level (SWL), while its top, which is coincident with the bottom of nacelle, is located at 87.6 m above the SWL. The centerline of the shaft passing through the center of the nacelle and hub corresponds to an elevation of 90 m. The tower is 3.87 m in diameter at the top and 6.5 m in diameter at the base. The latter dimension matches the diameter of the top of the floating platform. The platform consists of two cylindrical regions with depths of 14 and 108 m, respectively, connected by a linearly tapered conical region with a depth of 8 m. The linearly tapered conical region extends from 4 to 12 m below the SWL. The base of the platform has a diameter of 9.4 m and is located at 120 m below the SWL. Three mooring cables are attached to the platform at 70 m below the SWL, and are anchored at the seabed at 320 m below the SWL. From the top view, the three cables, with the original length of 902 m, are distributed 120° apart. The whole floating system is depicted in Fig. 3. Additional geometry and material data for this turbine may be found in [114], which is the main source of the material presented in this section.

The main wind-turbine structural components, including the rotor, nacelle, tower, and platform, are modeled using IGA Kirchhoff–Love shells. The bending

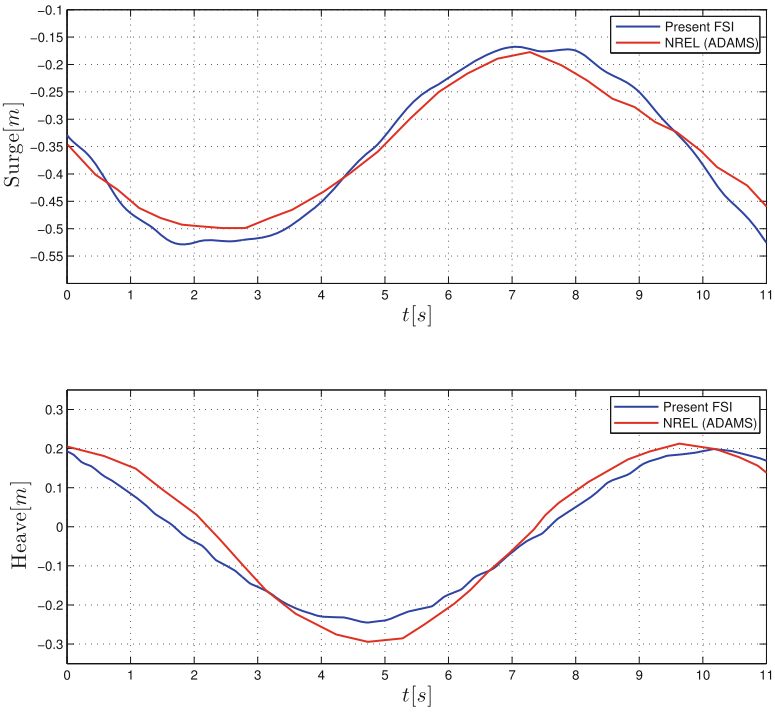


Fig. 4 Platform center-of-mass displacement for Airy-wave conditions

strip technique [54] is adopted to deal with the multiple-patch discretization. The main shaft is modeled as a collection of beams, while mooring lines are modeled as cables. Quadratic NURBS are employed for both shell, beam and cable discretization. The total number of shell elements is 14,709, and the total number of beam/cable elements is 33. The fluid domain is meshed with tetrahedra and triangular prisms, and the number of nodes and elements in the simulations are 4,216,201 and 24,817,979, respectively. In addition, while the blades are flexible, the rotor is not allowed to spin in the FSI simulations presented.

First the simulation is performed by using the Airy-wave inflow conditions with zero mean flow, 6 m wave height and 156.13 m wave length. The time history of average platform displacement is plotted in Fig. 4. The simulation results obtained by NREL using the ADAMS code [50] are also plotted in Fig. 4 as a reference. Although the NREL data is obtained from a simulation of the no-rotor configuration, good agreement between the two simulations is nevertheless achieved. This is not surprising, since the rotor mass is low compared to the overall mass of the floating turbine. Furthermore, wind loading on the rotor is also very low due to zero-wind-speed conditions employed in the simulation.

Although linear wave theory like Airy wave is often used to model sea wave, it is insufficient to model more "violent" sea conditions. To generate some more realistic

waves, a piston-type wave generation concept, which is widely used in offshore engineering laboratories to generate irregular waves, is utilized numerically in this work. Considering that the free-surface flow formulation is proposed in a moving mesh, periodic horizontal motion of the inlet boundary is prescribed to mimic a piston motion. After the waves are generated, the wave profile is transferred from the numerical-wave-tank domain to the domain used for the FSI simulation of the floating wind turbine to carry out the FSI computation.

Figure 5 shows the wave surface colored by streamwise velocity and the corresponding velocity vector field in the water domain at different time instances. The wave peak height is about 18 m in this simulation. The configurations of the floating wind turbine are also shown in these two figures. Compared with Airy wave conditions, more complex free-surface phenomena, such as splashing, is observed when the waves impact the platform. In these stronger waves, the platform displacement is much bigger that can be observed in Fig. 4. We also plot the time history of the blade tip displacement for all three blades in Fig. 6 (See Fig. 3 for blade numbering). While the displacement time histories of blades 2 and 3 are very similar and relatively low in magnitude, blade 1, whose tip is at the highest point on the wind-turbine structure (over 90 m higher than the tips of blades 2 and 3) undergoes displacement with a more complex time history and much larger magnitude. The tip displacement time histories reveal the following behavior. When the first wave peak reaches the turbine, the spar-buoy changes its direction of motion at a time instant of 3.7 s. The change in the direction of motion of the tips of blades 2 and 3 occurs at about 4.5 s, while blade 1 changes direction at about 4.8 s. This pattern repeats for other wave peaks.

6.2 FSI Modeling of Fatigue Damage in CX-100 Wind-Turbine Blades

We first deploy the DDDAS fatigue-damage framework on a full-scale cantilever composite blade CX-100, which is calibrated by the measured acceleration data collected from the indoor test [18]. The test blade was clamped at the root and driven by a hydraulic forcing with a natural frequency of 1.82 Hz of the first flap-wise bending mode. It was cyclically loaded until a fatigue-induced crack formed in the blade root region after about 8.0 M loading cycles. The accelerometers installed on the blade surface recorded the acceleration history. The blade mesh has 4647 quadratic NURBS elements. A time-periodic vertical displacement is applied at the blade root to mimic the hydraulic system forcing in the fatigue experiment. Two-DDDAS-loop scheme is devised to make use of the dynamic accelerometer data to update the computational model of the system. Figure 7 shows the predicted and measured accelerations in both time and frequency domains have good match. Excellent agreement between the location of the zone with most damage predicted by the simulation and the crack location observed in the fatigue test of the CX-100 composite blade is achieved, which may be seen in Fig. 7.

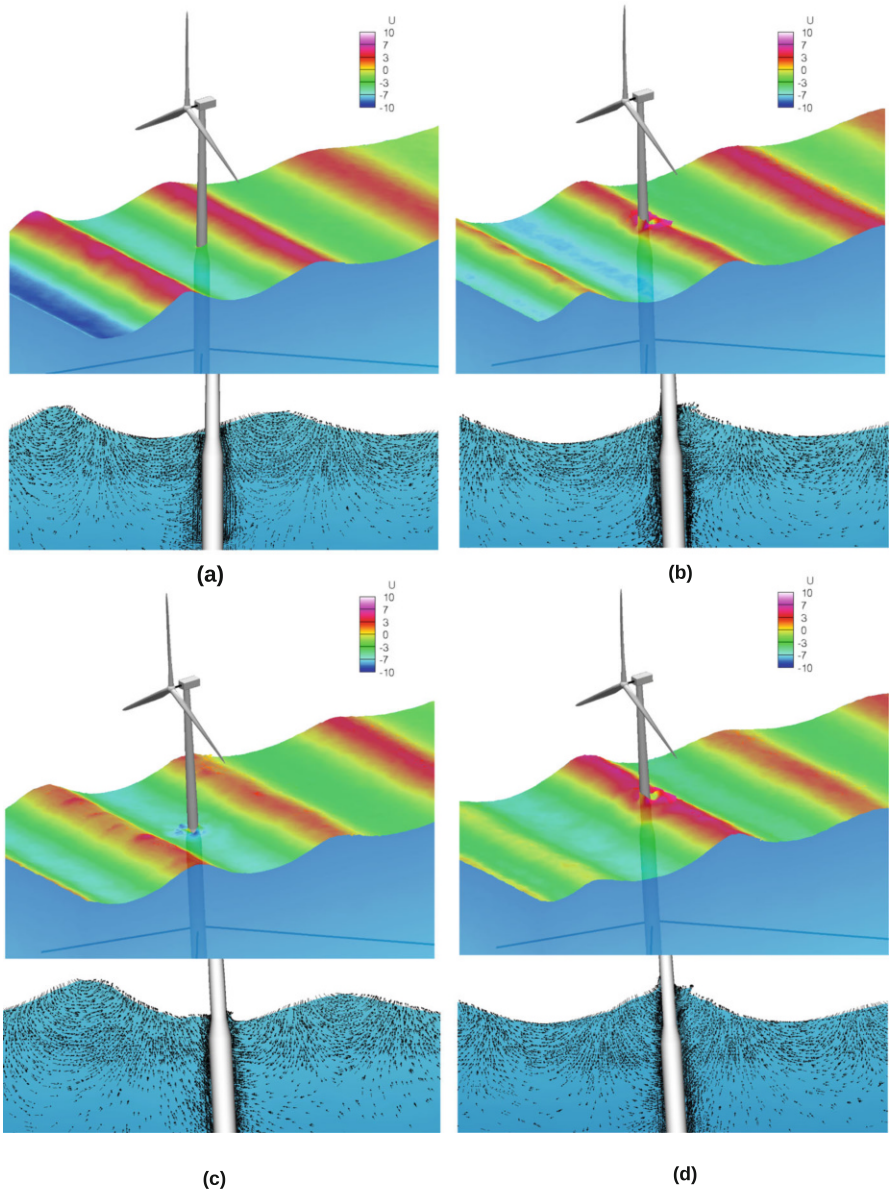


Fig. 5 Snapshots of free surface colored by streamwise velocity (in m/s) and the corresponding velocity filed in the water domain at different time instants for piston-generated wave conditions (**a**)–(**d**) designate snapshots taken at different times during the FSI simulation

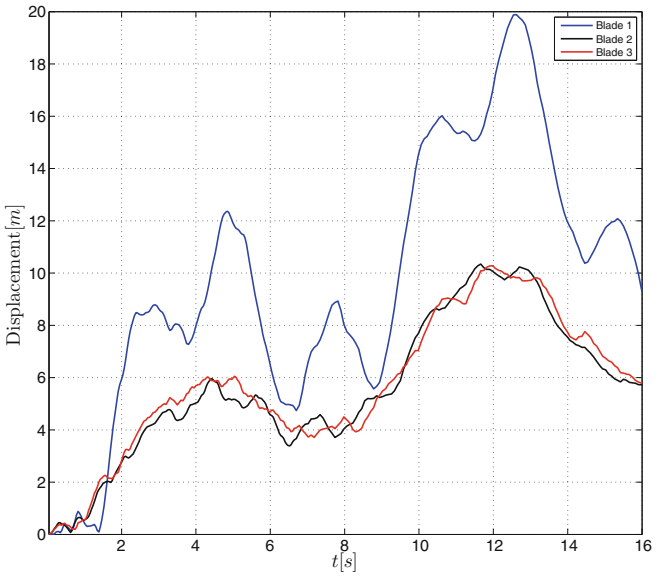


Fig. 6 Magnitude of tip displacement of three blades for piston-generated wave conditions

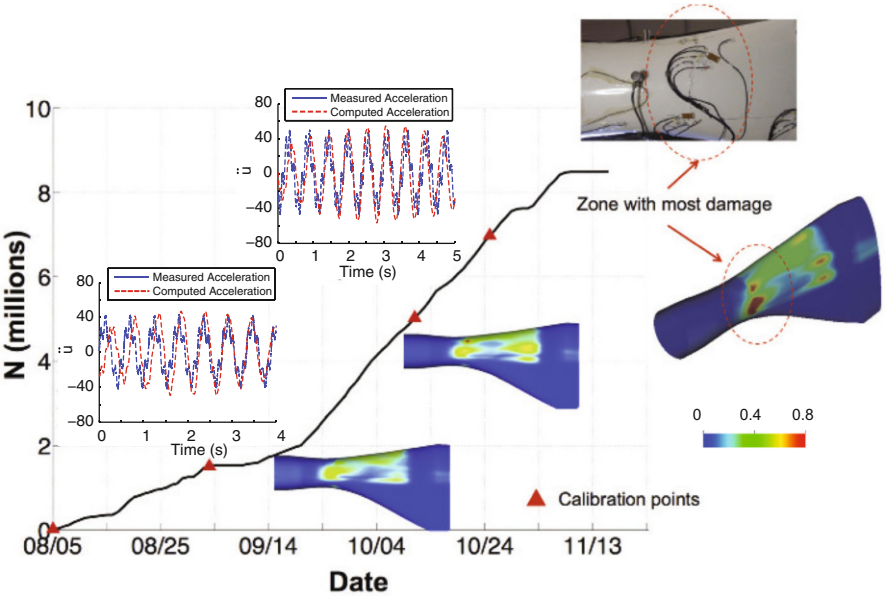


Fig. 7 Cycle count versus date for the fatigue test of the CX-100 blade. Triangular points indicate the calibration stations at which the simulation results for damage growth and acceleration history were compared to fatigue test data

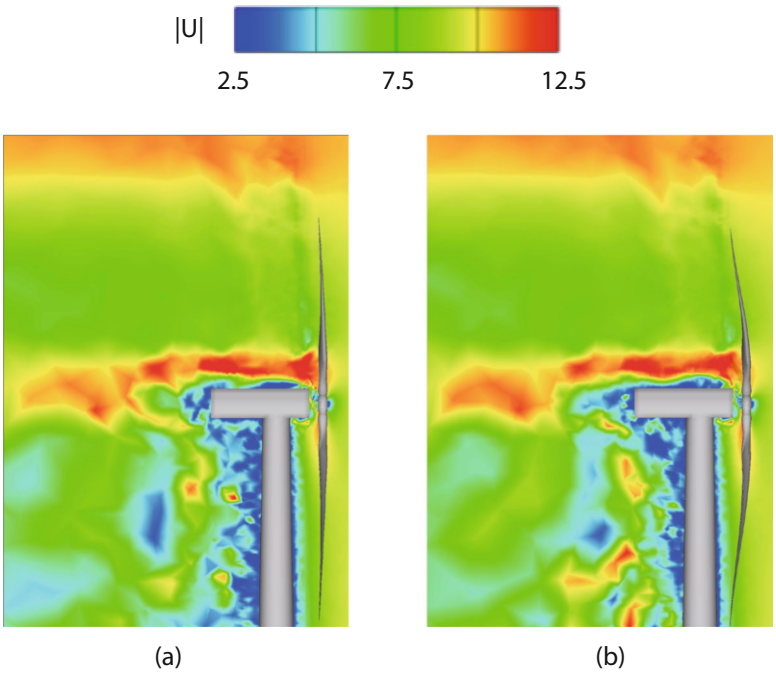


Fig. 8 Isocontours of air speed (in m/s) on a planar cut after 100,000 (**a**) and 150,000,000 (**b**) cycles. (**b**) Corresponds to the cycle right before the blade failure. Large blade tip deflection is due to significant loss of blade stiffness

For the next step, we simulate fatigue damage "in the field" as the wind turbine operates. A coupled FSI and CDM simulation is carried out for the Micon 65/13M wind turbine mounted with Sandia CX-100 blades under realistic wind and rotor speed conditions [16]. This is a fixed-pitch, upwind horizontal-axis turbine with rated power of 100 kW. The wind-turbine rotor spins at 55 rpm and is subjected to a wind speed of 10.5 m/s. The fluid mesh consists of 2,877,958 linear elements, which are triangular prisms in the rotor boundary layers and tetrahedra elsewhere in the domain. The blade boundary-layer mesh has 15 layers of prismatic elements generated with a growth ratio of 1.2. The size of the first boundary-layer element in the wall-normal direction is 0.002 m. The time-step size is set to 6.0×10^{-5} s. Figure 8 shows the large blade tip deflection due to the overall softening of the blade by the time damage propagates through the blade, when the cycle count reaches 150 M. Compared to the laboratory fatigue test, significant differences in damage patterns are observed for simulated realistic blade operating conditions (compare Figs. 7 and 9.) This result underscores the importance of considering realistic structure loading when studying blade fatigue damage, and employing advanced FSI in obtaining such loading data.

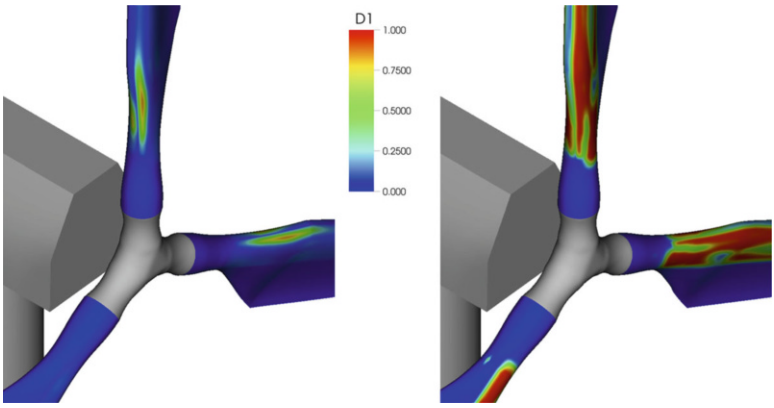


Fig. 9 Damage index D_1 in the DBM-1708 layer near the blade aerodynamic zone after 100,000,000 (left) and 150,000,000 (right) cycles

7 Conclusions

This article focused on FSI simulation of floating wind turbines subjected to combined wind-flow and ocean-wave forcing, and on modeling of high-cycle fatigue failure of blades due to long-term cyclic aerodynamic loading. Computational techniques, including advanced structural modeling based on IGA, free-surface FSI, and fatigue-damage modeling, were reviewed. A DDDAS framework for steering FSI simulations of high-cycle fatigue damage in composite turbine blades was also presented. Representative computational examples involving land-based and floating offshore wind-turbine designs illustrate the versatility and power of the computational methods developed.

Acknowledgements The authors gratefully acknowledge the support of AFOSR Award FA9550-16-1-0131 and NSF Award CBET-1306869.

References

1. J. E. Akin, T. E. Tezduyar, and M. Ungor. Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT). *Computers & Fluids*, 36:2–11, 2007.
2. 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*, 79:010905, 2012.
3. I. Akkerman, Y. Bazilevs, C. E. Kees, and M. W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230:4137–4152, 2011.
4. 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*, 50:719–727, 2012.

5. D. Allaire, G. Biros, J. Chambers, D. Kordonowy, O. Ghattas, and K. Wilcox. Dynamic data driven methods for self-aware aerospace vehicles. *Procedia Computer Science*, 9:1206–1210, 2012.
6. D. Allaire, J. Chambers, R. Cowlagi, D. Kordonowy, M. Lecerf, L. Mainini, D. Ulker, and K. Wilcox. An offline/online DDDAS capability for self-aware aerospace vehicles. *Procedia Computer Science*, 18:1959–1968, 2013.
7. C. Anitescu, Y.J. Yongjie Jessica Zhang, and T. Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015.
8. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs. Experimental and numerical FSI study of compliant hydrofoils. *Computational Mechanics*, 55:1079–1090, 2015.
9. 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*, 229:3402–3414, 2010.
10. 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*, 197:173–201, 2007.
11. Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.
12. Y. Bazilevs, Ming-Chen Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.
13. Y. Bazilevs, Ming-Chen 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(supp02):1230002, 2012.
14. Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.
15. Y. Bazilevs, A Korobenko, X. Deng, and J. Yan. Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines. *International Journal for Numerical Methods in Engineering*, 102:766–783, 2015.
16. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan. FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades. *Journal of Applied Mechanics*, 83(6):061010, 2016.
17. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel, and J. O. Dabiri. FSI modeling of vertical-axis wind turbines. *Journal of Applied Mechanics*, 81:081006, 2014.
18. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S. M. I. Gohari, and S. Sarkar. ALE–VMS formulation for stratified turbulent incompressible flows with applications. *Mathematical Models and Methods in Applied Sciences*, 25:2349–2375, 2015.
19. Y. Bazilevs, A.L. Marsden, F. Lanza di Scalea, A. Majumdar, and M. Tatineni. Toward a computational steering framework for large-scale composite structures based on continually and dynamically injected sensor data. *Procedia Computer Science*, 9:1149–1158, 2012.
20. 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*, 199:780–790, 2010.
21. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Challenges and directions in computational fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 23: 215–221, 2013.
22. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods. *Mathematical Models and Methods in Applied Sciences*, 25:2217–2226, 2015.
23. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013.

24. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre. Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Archives of Computational Methods in Engineering*, 21:359–398, 2014.
25. Y. Bazilevs, J. Yan, M. de Stadler, and S. Sarkar. Computation of the flow over a sphere at Re = 3700: A comparison of uniform and turbulent inflow conditions. *Journal of Applied Mechanics*, 81(12):121003, 2014.
26. D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200: 1367–1378, 2011.
27. M.J. Borden, T.J.R Hughes, C.M. Landis, and C.V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 273:100–118, 2014.
28. 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*, 32:199–259, 1982.
29. 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*, 60:371–75, 1993.
30. J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis. Toward Integration of CAD and FEA*. Wiley, 2009.
31. 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*, 191:525–543, 2001.
32. M. Cruchaga, D. Celentano, and T. Tezduyar. Computation of mould filling processes with a moving Lagrangian interface technique. *Communications in Numerical Methods in Engineering*, 18:483–493, 2002.
33. 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*, 47:451–469, 2005.
34. I. M. Daniel and O. Ishai. *Engineering Mechanics of Composite Materials*. Oxford University Press, 2006.
35. F. Darema. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. *in proceedings of ICCS 2004 4th International Conference on Computational Science*, pages 662–669, 2004.
36. J. Degrieck and W. V. Paepegem. Fatigue damage modelling of fiber-reinforced composite materials: Review. *Applied Mechanics Reviews*, 54(4):279–300, 2001.
37. R. Echter, B. Oesterle, and M. Bischoff. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 254:170–180, 2013.
38. T.D. Griffith. Structural health and prognostics management for offshore wind plants: Final report of Sandia research and developments activities. 2015. Report of the Sandia National Laboratory.
39. Y. Guo and M. Ruess. A layerwise isogeometric approach for NURBS-derived laminate composite shells. *Composite Structures*, 124:300–309, 2015.
40. M. Hillman, J.S. Chen, and Y. Bazilevs. Variationally consistent domain integration for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284: 521–540, 2015.
41. S. Hosseini, J.J.C. Remmers, C.V. Verhoosel, and R. de Borst. Propagation of delamination in composite materials with isogeometric continuum shell elements. *International Journal for Numerical Methods in Engineering*, 102:159–179, 2015.
42. M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: Simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.
43. M.-C. Hsu, C. Wang, A.J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers and Mathematics with Applications*, 70:1481–1500, 2015.

44. Ming-Chen Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.
45. Ming-Chen Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54:1055–1071, 2014.
46. 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*, 194:4135–4195, 2005.
47. 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*, 29:329–349, 1981.
48. 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*, 190:305–319, 2000.
49. 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*, 119:73–94, 1994.
50. J. Jonkman and W. Musial. Offshore code comparison collaboration (OC3) for IEA task 23 offshore wind technology and deployment. *NREL Technical Report*, 2010.
51. J. M. Jonkman. *Definition of the Floating System for Phase IV of OC3*. National Renewable Energy Laboratory Golden, CO, USA, 2010.
52. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
53. 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*, 230:4536–4558, 2011.
54. J. Kiendl, Y. Bazilevs, Ming-Chen Hsu, R. Wüchner, and Kai-Uwe Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199: 2403–2416, 2010.
55. J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
56. J. Kiendl, M.-C. Hsu, M.C.H. Wu, and A. Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, 2015.
57. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Aerodynamic simulation of vertical-axis wind turbines. *Journal of Applied Mechanics*, 81(2), 021011, 2013.
58. A. Korobenko, Ming-Chen Hsu, I. Akkerman, and Y. Bazilevs. Aerodynamic simulation of vertical-axis wind turbines. *Journal of Applied Mechanics*, 81:021011, 2013.
59. A. Korobenko, Ming-Chen Hsu, I. Akkerman, J. Tippmann, and Y. Bazilevs. Structural mechanics modeling and FSI simulation of wind turbines. *Mathematical Models and Methods in Applied Sciences*, 23:249–272, 2013.
60. S. Lee, M. Churchfield, P. Moriarty, J. Jonkman, and J. Michalakes. Atmospheric and wake turbulence impacts on wind turbine fatigue loading. 2012. Conference Paper presented at $50th$ AIAA Aerospace Sciences Meeting, Nashville, Tennessee, 2012.
61. J. T. Oden, K. R. Diller, C. Bajaj, J. C. Browne, J. Hazle, I. Babuska, J. Bass, L. Demkowicz, Y. Feng, D. Fuentes, S. Prudhomme, M. N. Rylander, R. J. Stafford, and Y. Zhang. Dynamic data-driven finite element models for laser treatment of prostate cancer. *Num. Meth. PDE*, 23:904–922, 2007.

62. S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science and Business Media, 2006.
63. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
64. W. V. Paepegem and J. Degrieck. A new coupled approach of residual stiffness and strength for fatigue of fiber-reinforced composites. *International Journal of Fatigue*, 24:747–762, 2002.
65. W. V. Paepegem and J. Degrieck. Simulating in-plane fatigue damage in woven glass fibre-reinforced composites subject to fully reversed cyclic loading. *Fatigue Fract Engng Mater Struct*, 27:1197–1208, 2004.
66. P. Raghavan, S. Li, and S. Ghosh. Two scale response and damage modeling of composite materials. *Finite Element in Analysis and Design*, 40:1619–1640, 2004.
67. S.B. Raknes, X. Deng, Y. Bazilevs, D.J. Benson, K.M. Mathisen, and T. Kvamsdal. Isogeometric rotation-free bending-stabilized cables: Statics, dynamics, bending strips and coupling with shells. *Comput. Methods Appl. Mech. Engrg.*, 263:127–143, 2013.
68. J. N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, 2nd ed.* CRC Press, Boca Raton, FL, 2004.
69. Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.
70. Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.
71. D. Schillinger, M.J. Borden, and H.K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.
72. H.J. Sutherland. On the fatigue analysis of wind turbines. 1999. Report of the Sandia National Laboratory.
73. S. Swaminathan and S. Ghosh. Statistically equivalent representative volume elements for composite microstructures, part ii: With evolving damage. *Journal of Composite Materials*, 40:605–621, 2006.
74. S. Swaminathan, S. Ghosh, and N. J. Pagano. Statistically equivalent representative volume elements for composite microstructures, part i: Without damage. *Journal of Composite Materials*, 40:583–604, 2006.
75. K. Takizawa. Computational engineering analysis with the new-generation space–time methods. *Computational Mechanics*, 54:193–211, 2014.
76. 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*, 19:171–225, 2012.
77. K. Takizawa, Y. Bazilevs, T. E. Tezduyar, Ming-Chen Hsu, O. Øiseth, K. M. Mathisen, N. Kostov, and S. McIntyre. Engineering analysis and design with ALE-VMS and space–time methods. *Archives of Computational Methods in Engineering*, 21:481–508, 2014.
78. K. Takizawa, Y. Bazilevs, T. E. Tezduyar, C. C. Long, A. L. Marsden, and K. Schjodt. ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling. *Mathematical Models and Methods in Applied Sciences*, 24:2437–2486, 2014.
79. 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*, 23:307–338, 2013.
80. 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*, 65:271–285, 2011.
81. K. Takizawa, K. Tanizawa, T. Yabe, and T. E. Tezduyar. Computational ship hydrodynamics with the CIP method. In E. Onate, J. Garcia, P. Bergan, and T. Kvamsdal, editors, *Marine 2007*, Barcelona, Spain, 2007. CIMNE.

82. 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*, 54:1011–1019, 2007.
83. K. Takizawa and T. E. Tezduyar. Multiscale space–time fluid–structure interaction techniques. *Computational Mechanics*, 48:247–267, 2011.
84. K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.
85. K. Takizawa and T. E. Tezduyar. Space–time fluid–structure interaction methods. *Mathematical Models and Methods in Applied Sciences*, 22(supp02):1230001, 2012.
86. 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*, 52:1351–1364, 2013.
87. K. Takizawa, T. E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel. FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes. *Computational Mechanics*, 54:1203–1220, 2014.
88. K. Takizawa, T. E. Tezduyar, and A. Buscher. Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping. *Computational Mechanics*, 55:1131–1141, 2015.
89. K. Takizawa, T. E. Tezduyar, and R. Kolesar. FSI modeling of the Orion spacecraft drogue parachutes. *Computational Mechanics*, 55:1167–1179, 2015.
90. K. Takizawa, T. E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel. Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes. *Computational Mechanics*, 54:1461–1476, 2014.
91. K. Takizawa, T. E. Tezduyar, and N. Kostov. Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV. *Computational Mechanics*, 54:213–233, 2014.
92. K. Takizawa, T. E. Tezduyar, and T. Kuraishi. Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. *Mathematical Models and Methods in Applied Sciences*, 25:2227–2255, 2015.
93. K. Takizawa, T. E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi. Computational thermo-fluid analysis of a disk brake. *Computational Mechanics*, 57:965–977, 2016.
94. 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*, 53:1–15, 2014.
95. K. Takizawa, T. E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel. Space–time VMS method for flow computations with slip interfaces (ST-SI). *Mathematical Models and Methods in Applied Sciences*, 25:2377–2406, 2015.
96. K. Takizawa, S. Wright, C. Moorman, and T. E. Tezduyar. Fluid–structure interaction modeling of parachute clusters. *International Journal for Numerical Methods in Fluids*, 65:286–307, 2011.
97. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.
98. T. Tezduyar and S. Sathe. Stabilization parameters in SUPG and PSPG formulations. *Journal of Computational and Applied Mechanics*, 4:71–88, 2003.
99. T. E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.
100. T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.
101. T. E. Tezduyar. Finite elements in fluids: Special methods and enhanced solution techniques. *Computers & Fluids*, 36:207–223, 2007.
102. T. E. Tezduyar. Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces. *Computers & Fluids*, 36:191–206, 2007.

103. 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, pages 7–24, New York, 1992. ASME.
104. 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*, 54:855–900, 2007.
105. 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*, 195:2002–2027, 2006.
106. 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*, 43:39–49, 2008.
107. 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*, 195:5743–5753, 2006.
108. 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*, 27:1665–1710, 2011.
109. 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*, 64:1201–1218, 2010.
110. N. Valizadeh, Y. Bazilevs, J.S. Chen, and T. Rabczuk. A coupled IGA-Meshfree discretization of arbitrary order of accuracy and without global geometry parametrization. *Computer Methods in Applied Mechanics and Engineering*, 293:20–37, 2015.
111. T. van Opstal, E. Fonn, R. Holdahl, T. Kvamsdal, A.M. Kvarving, K.M. Mathisen, K. Nordanger, K.M. Okstad, A. Rasheed, and M. Tabib. Isogeometric methods for CFD and FSI-simulation of flow around turbine blades. *Energy Procedia*, 80:442–449, 2015.
112. L. Beio Da Veiga, T.J.R. Hughes, J. Kiendl, C. Lovadina, J. Niiranen, A. Reali, and H. Speleers. A locking-free model for Reissner–Mindlin plates: Analysis and isogeometric implementation via NURBS and triangular NURPS. *Mathematical Models and Methods in Applied Sciences*, 28:1519–1551, 2015.
113. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs. FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration. *Computers and Fluids*, 141:201–211, 2016.
114. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers and Fluids*, 141:155–174, 2016.

Aorta Flow Analysis and Heart Valve Flow and Structure Analysis

Kenji Takizawa, Tayfun E. Tezduyar, Hiroaki Uchikawa, Takuya Terahara, Takafumi Sasaki, Kensuke Shiozaki, Ayaka Yoshida, Kenji Komiya, and Gaku Inoue

Abstract We present our computational methods for and results from aorta flow analysis and heart valve flow and structure analysis. In flow analysis, the core method is the space–time Variational Multiscale (ST-VMS) method. The other key methods are the ST Slip Interface (ST-SI) and ST Topology Change (ST-TC) methods and the ST Isogeometric Analysis (ST-IGA). The ST framework, in a general context, provides higher-order accuracy. The VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flows in the aorta and heart valve. The moving-mesh feature of the ST framework enables high-resolution computation near the valve leaflets. The ST-SI connects the sectors of meshes containing the leaflets, enabling a more effective mesh moving. The ST-TC enables moving-mesh computation even with the TC created by the contact between the leaflets. It deals with the contact while maintaining high-resolution representation near the leaflets. Integration of the ST-SI and ST-TC enables high-resolution representation even though parts of the SI are coinciding with the leaflet surfaces. It also enables dealing with leaflet–leaflet contact location change and contact sliding. The ST-IGA provides smoother representation of aorta and valve surfaces and increased accuracy in the flow

K. Takizawa (✉) · H. Uchikawa · T. Terahara · T. Sasaki · K. Shiozaki · A. Yoshida · K. Komiya · G. Inoue
Department of Modern Mechanical Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: Kenji.Takizawa@tafsm.org

T. E. Tezduyar
Mechanical Engineering, Rice University, Houston, TX, USA

Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: tezduyar@tafsm.org

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_2

solution. With the integration of the ST-IGA with the ST-SI and ST-TC, the element density in the narrow spaces near the contact areas is kept at a reasonable level. In structure analysis, we use a Kirchhoff–Love shell model, where we take the stretch in the third direction into account in calculating the curvature term. The computations presented demonstrate the scope and effectiveness of the methods.

1 Introduction

Aorta flow analysis and heart valve flow analysis are two of the computationally challenging cases in cardiovascular fluid mechanics. The challenges in aorta flow analysis include unsteady flow through a curved geometry with multiple outlets, relatively high Reynolds numbers, and multiscale flow behavior. In heart valve flow analysis, the challenges include unsteady flow through a complex geometry, solid surfaces with large motion, and contact between the valve leaflets. The precursors of the work presented in this article were reported for aorta flow analysis in [1, 2] and for heart valve flow analysis in [3–5]. In this article, in the aorta flow analysis, we target a smoother representation of the geometry, increased accuracy in the flow solution, systematic determination of the mesh refinement influence, and an assessment of flow periodicity in the cardiac cycles. In the heart valve analysis, we do the flow computation with the leaflet deformation coming from a structure analysis with a Kirchhoff–Love shell model. In the structure computation, we use a prescribed, time-periodic, spatially uniform pressure difference between the upper and lower surfaces of the leaflets.

1.1 Flow Analysis Methods

In flow analysis, the core component of our computational technology is the space–time Variational Multiscale (ST-VMS) method [6–8]. The other key components are the ST Slip Interface (ST-SI) [9, 10] and ST Topology Change (ST-TC) [3, 11] methods and the ST Isogeometric Analysis (ST-IGA) [6, 12, 13]. Integration of these components resulted in the ST-SI-TC [14] and ST-SI-TC-IGA [4, 5] methods, with a significant increase in scope and accuracy.

1.1.1 ST-VMS

The ST-VMS is the VMS version of the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [15–17]. The DSD/SST was introduced for computation of flows with moving boundaries and interfaces (MBI), including fluid–structure interaction (FSI). In MBI computations the DSD/SST functions as a moving-mesh

method. Moving the fluid mechanics mesh to track a fluid–solid interface enables mesh-resolution control near the interface and, consequently, high-resolution representation of the boundary layer. Because the stabilization components of the DSD/SST are the Streamline-Upwind/Petrov-Galerkin (SUPG) [18] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [15] stabilizations, the method is also called "ST-SUPS." The VMS components of the ST-VMS are from the residual-based VMS (RBVMS) method [19–22]. The ST-VMS has two more stabilization terms beyond those the ST-SUPS has, and these additional terms give the method better turbulence modeling features. Conversely, we can see the ST-SUPS as a reduced version of the ST-VMS. The ST-SUPS and ST-VMS, because of the higher-order accuracy of the ST framework (see [6, 7]), are desirable also in computations that do not involve MBI.

The Arbitrary Lagrangian–Eulerian (ALE) method is an earlier and more commonly used moving-mesh method. The ALE finite element method was introduced in 1981 [23]. The ALE-VMS method [24–29] is the VMS version of the ALE. It was introduced after the ST-SUPS [15] and ALE-SUPS [30] methods and preceded the ST-VMS. The ALE-VMS and RBVMS are often supplemented with special methods, such as those for weakly enforced no-slip boundary condition [31–33], "sliding interfaces" [34, 35], and backflow stabilization [36]. They have been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include wind-turbine aerodynamics and FSI [37–44], more specifically, vertical-axis wind turbines [45, 46], floating wind turbines [47], wind turbines in atmospheric boundary layers [48], and fatigue damage in wind-turbine blades [49], patient-specific cardiovascular fluid mechanics and FSI [24, 50–55], biomedical-device FSI [56–61], ship hydrodynamics with free-surface flow and fluid–object interaction [62, 63], hydrodynamics and FSI of a hydraulic arresting gear [64, 65], hydrodynamics of tidal-stream turbines with free-surface flow [66], and bioinspired FSI for marine propulsion [67, 68].

The ST-SUPS and ST-VMS have also been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include spacecraft parachute analysis for the main parachutes [27, 69–72], cover-separation parachutes [73] and the drogue parachutes [74–76], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [27, 37, 77, 78], full horizontal-axis wind-turbines [43, 79–81] and vertical-axis wind-turbines [9], flapping-wing aerodynamics for an actual locust [12, 27, 82, 83], bioinspired MAVs [80, 81, 84, 85] and wing-clapping [11, 86], blood flow analysis of cerebral aneurysms [80, 87], stent-blocked aneurysms [87–89], aortas [1] and heart valves [3–5, 11, 81], spacecraft aerodynamics [73, 90], thermo-fluid analysis of ground vehicles and their tires [8], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [91], flow analysis of turbocharger turbines [13, 92, 93], flow around tires with road contact and deformation [14, 94], ram-air parachutes [95], and compressible-flow parachute aerodynamics [96].

In the flow analyses presented here, the ST framework provides higher-order accuracy in a general context. The VMS feature of the ST-VMS addresses the

computational challenges associated with the multiscale nature of the unsteady flows in the aorta and heart valve. The moving-mesh feature of the ST framework enables high-resolution computation near the valve leaflets as they move.

1.1.2 ST-SI

The ST-SI was introduced in [9], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS when we have spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers. The SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the flow field. The starting point in the development of the ST-SI was the ALE-VMS version for "sliding interfaces" [34, 35]. In the ST-SI, interface terms similar to those in the ALE-VMS version are added to the ST-VMS formulation to account for the compatibility conditions for the velocity and stress. An ST-SI version where the SI is between fluid and solid domains with weakly enforced Dirichlet boundary conditions for the fluid was also presented in [9]. The SI in this case is a "fluid–solid SI" rather than a standard "fluid–fluid SI." The ST-SI method introduced in [10] for the coupled incompressible-flow and thermal-transport equations retain the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been successfully applied to aerodynamic analysis of vertical-axis wind turbines [9], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [91], flow analysis of turbocharger turbines [13, 92, 93], flow around tires with road contact and deformation [14, 94], aerodynamic analysis of ram-air parachutes [95], and heart valve flow analysis [4, 5].

In another version of the ST-SI method presented in [9], the SI is between a thin porous structure and the fluid on its two sides. This enables dealing with the fabric porosity in a fashion consistent with how the standard fluid–fluid SIs are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid SIs. Furthermore, this version enables handling thin structures that have T-junctions. This method has been successfully used in incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [95]. The compressible-flow ST-SI methods were introduced in [96], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [96]. These, together with the compressible-flow ST SUPG method [97], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled successful ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [96]. The computations were in the context of finite element discretization.

1.1.3 ST-TC

The ST-TC [3, 11] was introduced for moving-mesh computation of flow problems with TC, such as contact between solid surfaces. Even before the ST-TC, the ST-SUPS and ST-VMS, when used with robust mesh update methods, have proven effective in flow computations where the solid surfaces are in near contact or create other near TC, if the nearness is sufficiently near for the purpose of solving the problem. Many classes of problems can be solved that way with sufficient accuracy. For examples of such computations, see the references mentioned in [11]. The ST-TC made moving-mesh computations possible even when there is an actual contact between solid surfaces or other TC. By collapsing elements as needed, without changing the connectivity of the "parent" mesh, the ST-TC can handle an actual TC while maintaining high-resolution boundary layer representation near solid surfaces. This enabled successful moving-mesh computation of heart valve flows [3–5], wing clapping [86], and flow around a flow around tires with road contact and deformation [14, 94].

In the heart valve flow analysis, the ST-TC enables moving-mesh computation even with the TC created by the actual contact between the valve leaflets. It deals with the contact while maintaining high-resolution representation near the leaflets.

1.1.4 ST-SI-TC

The ST-SI-TC is the integration of the ST-SI and ST-TC. A fluid–fluid SI requires elements on both sides of the SI. When part of an SI needs to coincide with a solid surface, which happens, for example, when the solid surfaces on two sides of an SI come into contact or when an SI reaches a solid surface, the elements between the coinciding SI part and the solid surface need to collapse with the ST-TC mechanism. The collapse switches the SI from fluid–fluid SI to fluid–solid SI. With that, an SI can be a mixture of fluid–fluid and fluid–solid SIs. With the ST-SI-TC, the elements collapse and are reborn independent of the nodes representing a solid surface. The ST-SI-TC enables high-resolution representation even when parts of the SI are coinciding with a solid surface. It also enables dealing with contact location change and contact sliding. This was used in heart valve flow analysis [4, 5] and flow around tires with road contact and deformation [14, 94].

In the heart valve flow analysis, the ST-SI-TC enables high-resolution representation even though parts of the SI are coinciding with the leaflet surfaces. It also enables dealing with leaflet–leaflet contact location change and contact sliding.

1.1.5 ST-IGA

The ST-IGA was introduced in [6]. It is the integration of the ST framework with isogeometric discretization. First computations with the ST-VMS and ST-IGA were

reported in [6] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. The stability and accuracy analysis given [6] for the advection equation showed that using higher-order basis functions in time would be essential in getting full benefit out of using higher-order basis functions in space.

In the early stages of the ST-IGA, the emphasis was on IGA basis functions in time. As pointed out in [6, 7] and demonstrated in [12, 82, 84], higher-order NURBS basis functions in time provide a more accurate representation of the motion of the solid surfaces and a mesh motion consistent with that. They also provide more efficiency in temporal representation of the motion and deformation of the volume meshes, and better efficiency in remeshing. That is how the ST/NURBS Mesh Update Method (STNMUM) was introduced and demonstrated in [12, 82, 84]. The name "STNMUM" was given in [79]. The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS basis functions in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly, and a "secondary mapping" [6, 7, 12, 27] enables also specifying a constant angular velocity for invariant speeds along the paths. The ST framework and NURBS in time also enable, with the "ST-C" method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [8, 10, 91, 98]. The STNMUM and desirable features of the ST-IGA with IGA basis functions in time have been demonstrated in many 3D computations. The classes of problems solved are flapping-wing aerodynamics for an actual locust [12, 27, 82, 83], bioinspired MAVs [80, 81, 84, 85] and wing-clapping [11, 86], separation aerodynamics of spacecraft [73], aerodynamics of horizontal-axis [43, 79–81] and vertical-axis [9] wind-turbines, thermo-fluid analysis of ground vehicles and their tires [8], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [91], and flow analysis of turbocharger turbines [13, 92, 93].

The ST-IGA with IGA basis functions in space provides more accurate representation of the geometry and increased accuracy in the flow solution. It has been utilized in ST computational flow analysis of turbocharger turbines [13, 92, 93], ram-air parachutes [95], tires with road contact and deformation [94], and heart valves [4, 5]. In the flow analyses presented here, the ST-IGA enables smoother representation of aorta and valve surfaces and increased flow solution accuracy.

1.1.6 ST-SI-TC-IGA

The turbocharger turbine, ram-air parachute, tire, and heart valve computations mentioned in the last paragraph of Sect. 1.1.5 were accomplished with the integration of the ST-IGA and ST-SI or ST-IGA, ST-SI and ST-TC. The turbocharger turbine analysis was based on the integration of the ST-SI and ST-IGA. The IGA basis functions were used in the spatial discretization of the fluid mechanics

equations and also in the temporal representation of the rotor and spinning-mesh motion. That enabled accurate representation of the turbine surfaces and rotor motion and increased accuracy in the flow solution. The ram-air parachute analysis was based on the integration of the ST-IGA, the ST-SI version that weakly enforces the Dirichlet conditions, and the ST-SI version that accounts for the porosity of a thin structure. The ST-IGA with IGA basis functions in space enabled, with relatively few number of unknowns, accurate representation of the parafoil geometry and increased accuracy in the flow solution. The volume mesh needed to be generated both inside and outside the parafoil. Mesh generation inside was challenging near the trailing edge because of the narrowing space. Using IGA basis functions addressed that challenge and still kept the element density near the trailing edge at a reasonable level. The tire flow analysis with road contact and deformation was based on the integration of the ST-SI, ST-TC, and ST-IGA. The ST-SI-TC-IGA enabled a more accurate representation of the geometry and motion of the tire surfaces, a mesh motion consistent with that, and increased accuracy in the flow solution. It also kept the element density in the tire grooves and in the narrow spaces near the contact areas at a reasonable level. When solid surfaces come into contact, the elements between the surface and the SI collapse. Before the elements collapse, the boundaries could be curved and complex, and the narrow spaces might have high-aspect-ratio elements. With NURBS elements, it is possible to deal with such adverse conditions rather effectively.

The heart valve flow analysis is also based on the integration of the ST-SI, ST-TC, and ST-IGA. The ST-SI-TC-IGA, beyond enabling a smoother representation of the arterial surfaces and increased accuracy in the flow solution, keeps the element density in the narrow spaces near the leaflet–leaflet contact areas at a reasonable level. Also, the adverse conditions created by the high-aspect-ratio elements in the narrow spaces between the curved and complex boundaries are again dealt with rather effectively.

An SI also provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used in the turbocharger turbine analysis. It is also used in the heart valve flow analysis, for the purpose of independent meshing in the inlet and outlet regions of the computational domain.

1.2 Structure Analysis Methods

A shell formulation based on the Kirchhoff–Love shell theory was implemented using isogeometric discretization [38, 99, 100], which does not require rotational

degrees of freedom. The formulation has been successfully used in a number of computations, including wind-turbine FSI [26, 27, 40, 100, 101] and bioinspired flapping-wing aerodynamics [85]. The extension to a more general constitutive law can be found in [102]. The formulation has been used in bioprosthetic heart valve computations with neo-Hookean [59] and Fung-type [60] material models.

In this article, we start with the formulation from [102], and take the stretch in the third direction into account in calculating the curvature term. Fung's model has different versions. In the version used in [60], the first invariant of the Cauchy–Green deformation tensor appears in a squared form. In the version we use in this article, it appears without being squared, and this version has been used in a number of arterial FSI computations [103–110] with continuum elements.

1.3 Computations Presented

In the aorta flow analysis, we use a geometry obtained from medical images and conduct studies to determine the mesh refinement influence and to assess flow periodicity in the cardiac cycles. To evaluate the performance of the shell model presented, we do structural mechanics test computations with cylindrical and spherical geometries and compare the results to near-analytical reference solutions. In the heart valve analysis, we first do a structural mechanics computation with a prescribed, time-periodic, spatially uniform pressure difference between the upper and lower surfaces of the valve leaflets. Based on the leaflet deformation coming from that computation, we next do a flow computation.

1.4 Outline of the Remaining Sections

In Sect. 2 we provide the governing equations. The hyperelastic shell model is presented in Sect. 3. The aorta flow analysis is presented in Sect. 4, test computations with cylindrical and spherical geometries in Sect. 5, heart valve analysis in Sect. 6, and the concluding remarks in Sect. 7. In the Appendix, we provide the ST-VMS and ST-SI methods, some mathematical derivations used in Sect. 3, and the constitutive laws.

2 Governing Equations

2.1 Incompressible Flow

Let $\Omega_t \subset \mathbb{R}^{n_{\mathrm{sd}}}$ be the spatial domain with boundary Γ_t at time $t \in (0, T)$, where n_{sd} is the number of space dimensions. The subscript t indicates the time-dependence of

the domain in general. The symbols ρ, $\mathbf{u}$, and p will represent the density, velocity, and pressure, respectively, and the stress tensor is defined as $\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, where $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u}) = \left((\boldsymbol{\nabla}\mathbf{u}) + (\boldsymbol{\nabla}\mathbf{u})^T\right)/2$ is the strain-rate tensor. The Navier–Stokes equations of incompressible flows are written on Ω_t and $\forall t \in (0, T)$ as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} - \mathbf{f}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \mathbf{0}, \tag{1}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0, \tag{2}$$

where $\mathbf{f}$ is the body force. The essential and natural boundary conditions for Eq. (1) are represented as $\mathbf{u} = \mathbf{g}$ on $(\Gamma_t)_{\mathrm{g}}$ and $\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}$ on $(\Gamma_t)_{\mathrm{h}}$, where $(\Gamma_t)_{\mathrm{g}}$ and $(\Gamma_t)_{\mathrm{h}}$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

2.2 Structural Mechanics

In this article, we will not have a formulation requiring fluid and structure definitions simultaneously. Therefore, for notation simplicity, we will reuse many of the symbols used in the fluid mechanics equations to represent their counterparts in the structural mechanics equations. To begin with, $\Omega_t \subset \mathbb{R}^{n_{\mathrm{sd}}}$ and Γ_t will represent the structure domain and its boundary. The structural mechanics equations are then written, on Ω_t and $\forall t \in (0, T)$, as

$$\rho\left(\frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}t^2} - \mathbf{f}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \mathbf{0}, \tag{3}$$

where $\mathbf{y}$ and $\boldsymbol{\sigma}$ are the displacement and Cauchy stress tensor. The essential and natural boundary conditions for Eq. (3) are represented as $\mathbf{y} = \mathbf{g}$ on $(\Gamma_t)_{\mathrm{g}}$ and $\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}$ on $(\Gamma_t)_{\mathrm{h}}$. The Cauchy stress tensor can be obtained from

$$\boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^T, \tag{4}$$

where $\mathbf{F}$ and J are the deformation gradient tensor and its determinant, and $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor. It is obtained from the strain-energy density function φ as follows:

$$\mathbf{S} \equiv \frac{\partial \varphi}{\partial \mathbf{E}}, \tag{5}$$

where $\mathbf{E}$ is the Green–Lagrange strain tensor:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right), \tag{6}$$

and $\mathbf{C}$ is the Cauchy–Green deformation tensor:

$$\mathbf{C} \equiv \mathbf{F}^T \cdot \mathbf{F}. \tag{7}$$

From Eqs. (5) and (6),

$$\mathbf{S} = 2\frac{\partial \varphi}{\partial \mathbf{C}}. \tag{8}$$

3 Hyperelastic Shell Model

We split the domain as $\Omega_t = \overline{\Gamma}_t \times (h_{\text{th}})_t$, where $\overline{\Gamma}_t$ represents the midsurface of the domain, which is parametrized by $n_{\text{pd}} = n_{\text{sd}} - 1$, where n_{pd} is the number of parametric dimensions. With the position $\overline{\mathbf{x}} \in \overline{\Gamma}_t$, we define a natural coordinate system:

$$\overline{\mathbf{g}}_\alpha \equiv \frac{\partial \overline{\mathbf{x}}}{\partial \xi^\alpha} \tag{9}$$

$$= \overline{\mathbf{x}}_{,\alpha}, \tag{10}$$

where $\alpha = 1, \ldots, n_{\text{pd}}$, and the third direction is based on

$$\mathbf{n} \equiv \overline{\mathbf{g}}_3 \tag{11}$$

$$= \frac{\overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2}{\left\| \overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2 \right\|}. \tag{12}$$

The components of the metric tensor are

$$\overline{g}_{\alpha\beta} = \overline{\mathbf{g}}_\alpha \cdot \overline{\mathbf{g}}_\beta, \tag{13}$$

and this is known as the first fundamental form. Similarly, we define the components of the metric tensor for the contravariant basis vectors as

$$\overline{g}^{\alpha\beta} = \overline{\mathbf{g}}^\alpha \cdot \overline{\mathbf{g}}^\beta, \tag{14}$$

and obtain the tensor components and contravariant basis vectors from

$$\left[\overline{g}^{\alpha\beta}\right] = \left[\overline{g}_{\alpha\beta}\right]^{-1} \tag{15}$$

and

$$\overline{\mathbf{g}}^{\alpha} = \overline{g}^{\alpha\beta}\overline{\mathbf{g}}_{\beta}. \tag{16}$$

We define

$$\overline{\boldsymbol{\Gamma}}_{\alpha\beta} = \frac{\partial^2 \overline{\mathbf{x}}}{\partial \xi^{\alpha} \partial \xi^{\beta}}, \tag{17}$$

and with that, components of the covariant curvature tensor are

$$\overline{b}_{\alpha\beta} = \overline{\boldsymbol{\Gamma}}_{\alpha\beta} \cdot \mathbf{n} \tag{18}$$

$$= \overline{\mathbf{g}}_{\beta,\alpha} \cdot \mathbf{n}, \tag{19}$$

and this is known as the second fundamental form.

A position $\mathbf{x} \in \Omega_t$ is represented as

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{n}\xi^3, \tag{20}$$

where $-1 \leq \xi^{\alpha} \leq 1$ and $\xi^3 \in (h_{\text{th}})_t$. The basis vectors are represented as

$$\mathbf{g}_{\alpha} \equiv \mathbf{x}_{,\alpha} \tag{21}$$

$$= \overline{\mathbf{g}}_{\alpha} + \mathbf{n}_{,\alpha}\xi^3 \tag{22}$$

$$= \overline{\mathbf{g}}_{\alpha} - \overline{b}_{\alpha\gamma}\overline{\mathbf{g}}^{\gamma}\xi^3. \tag{23}$$

See section "Derivative of the Normal Vector" in Appendix 2 for the lines between Eqs. (22) and (23). Because $\mathbf{g}_{\alpha}$ and $\overline{\mathbf{g}}_{\alpha}$ are on parallel planes (from the Kirchhoff–Love shell theory),

$$\mathbf{g}_3 = \overline{\mathbf{g}}_3. \tag{24}$$

With that, the metric tensor components in 3D space are

$$g_{\alpha\beta} = \overline{g}_{\alpha\beta} - 2\overline{b}_{\alpha\beta}\xi^3 + \overline{b}_{\alpha\gamma}\overline{g}^{\gamma\delta}\overline{b}_{\beta\delta}\left(\xi^3\right)^2, \tag{25}$$

$$g_{\alpha 3} = 0, \tag{26}$$

$$g_{3\alpha} = 0, \tag{27}$$

$$g_{33} = 1. \tag{28}$$

Remark 1 The quadratic term may be omitted. However, if the metric tensor is obtained from the basis vectors, the term will automatically be included.

We now provide similar definitions and derivations for the undeformed configuration. We start with the basis vectors:

$$\overline{\mathbf{G}}_\alpha = \frac{\partial \overline{\mathbf{X}}}{\partial \xi_0^\alpha} \tag{29}$$

$$= \overline{\mathbf{X}}_{,\alpha}, \tag{30}$$

and

$$\mathbf{N} \equiv \overline{\mathbf{G}}_3 \tag{31}$$

$$= \frac{\overline{\mathbf{G}}_1 \times \overline{\mathbf{G}}_2}{\left\| \overline{\mathbf{G}}_1 \times \overline{\mathbf{G}}_2 \right\|}. \tag{32}$$

A position $\mathbf{X} \in \Omega_0$ is expressed as

$$\mathbf{X} = \overline{\mathbf{X}} + \mathbf{N}\xi_0^3, \tag{33}$$

where $-1 \leq \xi_0^\alpha \leq 1$ and $\xi_0^3 \in (h_{\text{th}})_0$. The basis vectors are represented as

$$\mathbf{G}_\alpha = \mathbf{X}_{,\alpha} \tag{34}$$

$$= \overline{\mathbf{G}}_\alpha + \mathbf{N}_{,\alpha}\xi_0^3 \tag{35}$$

$$= \overline{\mathbf{G}}_\alpha - \overline{B}_{\alpha\gamma}\overline{\mathbf{G}}^\gamma \xi_0^3, \tag{36}$$

$$\mathbf{G}_3 = \overline{\mathbf{G}}_3. \tag{37}$$

The metric tensor components in 3D space are

$$G_{\alpha\beta} = \overline{G}_{\alpha\beta} - 2\overline{B}_{\alpha\beta}\xi_0^3 + \overline{B}_{\alpha\gamma}\overline{G}^{\gamma\delta}\overline{B}_{\beta\delta}\left(\xi_0^3\right)^2, \tag{38}$$

$$G_{3\alpha} = 0, \tag{39}$$

$$G_{\alpha 3} = 0, \tag{40}$$

$$G_{33} = 1, \tag{41}$$

and $\overline{B}_{\alpha\beta}$ is the second fundamental form for the midsurface of the undeformed configuration. On the midsurface the parametric coordinates indicate the same material points, and therefore, $\xi^\alpha = \xi_0^\alpha$. In the third direction, however, because of the normalization, the coordinates may not be the same. The relationship is

$$\frac{\mathrm{d}\xi^3}{\mathrm{d}\xi_0^3} = \lambda_3, \tag{42}$$

where λ_3 is the stretch in the third direction.

3.1 Kinematics

We obtain $\mathbf{F}$ from the following relationship:

$$\mathbf{g}_\alpha \mathrm{d}\xi_0^\alpha + \mathbf{g}_3\lambda_3\mathrm{d}\xi_0^3 = \mathbf{F}\cdot\left(\mathbf{G}_\alpha \mathrm{d}\xi_0^\alpha + \mathbf{G}_3\mathrm{d}\xi_0^3\right). \tag{43}$$

This means that

$$\mathbf{g}_\alpha = \mathbf{F}\cdot\mathbf{G}_\alpha, \tag{44}$$

$$\lambda_3\mathbf{g}_3 = \mathbf{F}\cdot\mathbf{G}_3. \tag{45}$$

Then we can write $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{g}_\alpha\mathbf{G}^\alpha + \lambda_3\mathbf{g}_3\mathbf{G}^3, \tag{46}$$

and J as

$$J = \frac{1}{\mathbf{G}_3\cdot(\mathbf{G}_1\times\mathbf{G}_2)}(\mathbf{F}\cdot\mathbf{G}_3)\cdot((\mathbf{F}\cdot\mathbf{G}_1)\times(\mathbf{F}\cdot\mathbf{G}_2)) \tag{47}$$

$$= \lambda_3\frac{(\mathbf{g}_1\times\mathbf{g}_2)\cdot\mathbf{n}}{(\mathbf{G}_1\times\mathbf{G}_2)\cdot\mathbf{N}} \tag{48}$$

$$= \lambda_3\frac{\|\mathbf{g}_1\times\mathbf{g}_2\|}{\|\mathbf{G}_1\times\mathbf{G}_2\|}. \tag{49}$$

From Eq. (7), we can write $\mathbf{C}$ as

$$\mathbf{C} = \left(\mathbf{G}^\alpha\mathbf{g}_\alpha + \lambda_3\mathbf{G}^3\mathbf{g}_3\right)\cdot\left(\mathbf{g}_\beta\mathbf{G}^\beta + \lambda_3\mathbf{g}_3\mathbf{G}^3\right) \tag{50}$$

$$= g_{\alpha\beta}\mathbf{G}^\alpha\mathbf{G}^\beta + \lambda_3^2\mathbf{G}^3\mathbf{G}^3, \tag{51}$$

and the determinant of $\mathbf{C}$ gives the square of J:

$$J^2 = \det \mathbf{C} \tag{52}$$

$$= \frac{A^2}{A_0^2}\lambda_3^2, \tag{53}$$

$$A^2 = \det\left[g_{\alpha\beta}\right], \tag{54}$$

$$A_0^2 = \det\left[G_{\alpha\beta}\right]. \tag{55}$$

From Eq. (6), we can write $\mathbf{E}$ as

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \left(G_{\alpha\beta}\mathbf{G}^\alpha\mathbf{G}^\beta + \mathbf{G}^3\mathbf{G}^3\right)\right) \tag{56}$$

$$= \frac{1}{2}\left(g_{\alpha\beta} - G_{\alpha\beta}\right)\mathbf{G}^\alpha\mathbf{G}^\beta + \frac{1}{2}\left(\lambda_3^2 - 1\right)\mathbf{G}^3\mathbf{G}^3. \tag{57}$$

The covariant components of the in-plane strain tensor are

$$E_{\alpha\beta} = \frac{1}{2}\left(g_{\alpha\beta} - G_{\alpha\beta}\right) \tag{58}$$

$$= \underbrace{\frac{1}{2}\left(\overline{g}_{\alpha\beta} - \overline{G}_{\alpha\beta}\right)}_{\overline{\varepsilon}_{\alpha\beta}} + \left(-\overline{b}_{\alpha\beta}\xi^3 + \overline{B}_{\alpha\beta}\xi_0^3\right)$$

$$+\frac{1}{2}\left(\overline{b}_{\alpha\gamma}\overline{g}^{\gamma\delta}\overline{b}_{\beta\delta}\left(\xi^3\right)^2 - \overline{B}_{\alpha\gamma}\overline{G}^{\gamma\delta}\overline{B}_{\beta\delta}\left(\xi_0^3\right)^2\right). \tag{59}$$

We write $\xi^3\left(\xi_0^3\right)$ as $\xi^3\left(\xi_0^3\right) = \hat{\lambda_3}\left(\xi_0^3\right)\xi_0^3$. From the Taylor expansion of $\hat{\lambda_3}$ around $\xi_0^3 = 0$, we obtain

$$\xi^3 = \overline{\lambda_3}\xi_0^3 + \frac{\mathrm{d}\hat{\lambda_3}}{\mathrm{d}\xi_0^3}\left(\xi_0^3\right)^2 + \mathcal{O}\left(\left(\xi_0^3\right)^3\right). \tag{60}$$

We note that $\overline{\lambda_3}$ is the stretch at $\xi_0^3 = 0$, which is $\hat{\lambda_3}(0)$. With that,

$$E_{\alpha\beta} = \overline{\varepsilon}_{\alpha\beta} + \left(-\overline{b}_{\alpha\beta}\xi^3 + \overline{B}_{\alpha\beta}\xi_0^3\right)$$

$$+ \frac{1}{2}\left(\overline{b}_{\alpha\gamma}\overline{g}^{\gamma\delta}\overline{b}_{\beta\delta}\overline{\lambda_3}^2 - \overline{B}_{\alpha\gamma}\overline{G}^{\gamma\delta}\overline{B}_{\beta\delta}\right)\left(\xi_0^3\right)^2 + \mathcal{O}\left(\left(\xi_0^3\right)^3\right) \tag{61}$$

$$= \overline{\varepsilon}_{\alpha\beta} + \left(-\overline{b}_{\alpha\beta}\xi^3 + \overline{B}_{\alpha\beta}\xi_0^3\right) + \mathcal{O}\left(\left(\xi_0^3\right)^2\right). \tag{62}$$

3.2 Constitutive Equations

The total differential of the second Piola–Kirchhoff stress tensor is

$$\mathrm{d}\mathbf{S} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : \mathrm{d}\mathbf{E} \tag{63}$$

$$= \frac{\partial S^{IJ}}{\partial E_{KL}} \mathbf{G}_I \mathbf{G}_J \mathbf{G}_K \mathbf{G}_L : \mathrm{d}E_{MN} \mathbf{G}^M \mathbf{G}^N \tag{64}$$

$$= \mathbb{C}^{IJKL} \mathbf{G}_I \mathbf{G}_J \mathrm{d}E_{KL}, \tag{65}$$

where $I, J, K, L, M, N = 1, \ldots, n_{\mathrm{sd}}$. From Eq. (6), the following expression can be used:

$$\mathbb{C}^{IJKL} = 2 \frac{\partial S^{IJ}}{\partial C_{KL}}. \tag{66}$$

For shells,

$$\mathrm{d}S^{IJ} = \mathbb{C}^{IJ\gamma\delta} \mathrm{d}E_{\gamma\delta} + \mathbb{C}^{IJ33} \mathrm{d}E_{33}, \tag{67}$$

because $\mathrm{d}E_{3\alpha} = \mathrm{d}E_{\alpha 3} = 0$. From Eq. (67), we can write

$$\mathrm{d}S^{33} = \mathbb{C}^{33\gamma\delta} \mathrm{d}E_{\gamma\delta} + \mathbb{C}^{3333} \mathrm{d}E_{33}, \tag{68}$$

and from Eqs. (6) and (68), we can write

$$\mathrm{d}S^{33} = \frac{1}{2} \mathbb{C}^{33\gamma\delta} \mathrm{d}C_{\gamma\delta} + \frac{1}{2} \mathbb{C}^{3333} \mathrm{d}C_{33}. \tag{69}$$

From the plane stress condition $S^{33} = 0$, $\mathrm{d}S^{33} = 0$, and consequently

$$\mathrm{d}E_{33} = -\frac{\mathbb{C}^{33\gamma\delta}}{\mathbb{C}^{3333}} \mathrm{d}E_{\gamma\delta}, \tag{70}$$

which makes

$$\mathrm{d}S^{\alpha\beta} = \mathbb{C}^{\alpha\beta\gamma\delta} \mathrm{d}E_{\gamma\delta} - \frac{\mathbb{C}^{\alpha\beta 33} \mathbb{C}^{33\gamma\delta}}{\mathbb{C}^{3333}} \mathrm{d}E_{\gamma\delta}, \tag{71}$$

and therefore we introduce

$$\hat{\mathbb{C}}^{\alpha\beta\gamma\delta} = \mathbb{C}^{\alpha\beta\gamma\delta} - \frac{\mathbb{C}^{\alpha\beta 33} \mathbb{C}^{33\gamma\delta}}{\mathbb{C}^{3333}}. \tag{72}$$

In computing C_{33}, we have different methods for incompressible and compressible materials. In the case of incompressible material, from Eq. (51) we can write $C_{33} = \lambda_3^2$. Because $J = 1$,

$$\lambda_3 = \frac{1}{A/A_0}, \tag{73}$$

and therefore

$$C_{33} = \frac{A_0^2}{A^2}. \tag{74}$$

In the case of compressible material, as can be found in [52], C_{33} can be calculated by Newton–Raphson iterations that would make $S^{33} = 0$. Because $C_{\gamma\delta}$ does not change during the iterations, the iteration increment is

$$\Delta C_{33}^i = -\frac{\left(S^{33}\right)^i}{\left(\mathrm{d}S^{33}/\mathrm{d}C_{33}\right)^i}. \tag{75}$$

From Eq. (69) and remembering that $\mathrm{d}C_{\gamma\delta} = 0$ during the iterations,

$$\Delta C_{33}^i = -2\frac{\left(S^{33}\right)^i}{\left(\mathbb{C}^{3333}\right)^i}. \tag{76}$$

The update takes place as

$$C_{33}^{i+1} = C_{33}^i + \Delta C_{33}^i, \tag{77}$$

where superscript i is the iteration counter, and as the initial guess we have the following three options:

$$C_{33}^0 = 1, \tag{78}$$

$$C_{33}^0 = \frac{A_0^2}{A^2}, \tag{79}$$

$$C_{33}^0 = \frac{1}{2} g_{\alpha\beta} G^{\alpha\beta}. \tag{80}$$

The option given by Eq. (80) comes from the constitutive law for zero bulk modulus. To preclude C_{33} being negative, we introduce an alternative update method based on the logarithm of C_{33}:

$$\ln C_{33}^{i+1} = \ln C_{33}^i + \frac{\mathrm{d}\ln C_{33}}{\mathrm{d}C_{33}} \Delta C_{33}^i \tag{81}$$

$$= \ln C_{33}^i + \frac{\Delta C_{33}^i}{C_{33}^i}. \tag{82}$$

3.3 Variational Formulation

The variation of the in-plane components of the Green–Lagrange tensor is

$$\delta E_{\alpha\beta} = \delta\overline{\varepsilon}_{\alpha\beta} - \delta\overline{b}_{\alpha\beta}\xi^3 + \overline{b}_{\alpha\beta}\delta\xi^3. \tag{83}$$

The variation of ξ^3 can be dropped (see Appendix 3), and we obtain

$$\delta E_{\alpha\beta} = \delta\overline{\varepsilon}_{\alpha\beta} + \underbrace{\left(-\delta\overline{b}_{\alpha\beta}\right)}_{\delta\overline{\kappa}_{\alpha\beta}}\xi^3. \tag{84}$$

With that,

$$\delta W_{\text{int}} = -\int_{\Omega_0} \delta\mathbf{E} : \mathbf{S}\mathrm{d}\Omega \tag{85}$$

$$= -\int_{\overline{\Gamma}_0}\int_{(h_{\text{th}})_0} \delta\overline{\varepsilon}_{\alpha\beta}\mathbf{G}^\alpha\mathbf{G}^\beta : S^{\gamma\delta}\mathbf{G}_\gamma\mathbf{G}_\delta\frac{A_0}{\overline{A}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma$$

$$-\int_{\overline{\Gamma}_0}\int_{(h_{\text{th}})_0} \xi^3\delta\overline{\kappa}_{\alpha\beta}\mathbf{G}^\alpha\mathbf{G}^\beta : S^{\gamma\delta}\mathbf{G}_\gamma\mathbf{G}_\delta\frac{A_0}{\overline{A}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma, \tag{86}$$

which means

$$\delta W_{\text{int}} = -\int_{\overline{\Gamma}_0}\delta\overline{\varepsilon}_{\alpha\beta}\int_{(h_{\text{th}})_0} S^{\alpha\beta}\frac{A_0}{\overline{A}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma - \int_{\overline{\Gamma}_0}\delta\overline{\kappa}_{\alpha\beta}\int_{(h_{\text{th}})_0}\xi^3 S^{\alpha\beta}\frac{A_0}{\overline{A}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma. \tag{87}$$

Remark 2 Evaluation of $\mathbf{S}$ requires a material point correspondence in the third direction. We take that into account by integrating Eq. (42) with the 4th order Runge–Kutta method, and λ_3 can be obtained from the constitutive law given in Sect. 5.1. In general, stretch at a convex side is less than the stretch at the concave side, which results in a nonuniform λ_3.

Now we derive what we need:

$$\delta\overline{\varepsilon}_{\alpha\beta} = \frac{1}{2}\left(\delta\overline{\mathbf{g}}_\alpha\cdot\overline{\mathbf{g}}_\beta + \overline{\mathbf{g}}_\alpha\cdot\delta\overline{\mathbf{g}}_\beta\right) \tag{88}$$

$$= \frac{1}{2}\left(\frac{\partial\delta\overline{\mathbf{x}}}{\partial\xi^\alpha}\cdot\overline{\mathbf{g}}_\beta + \overline{\mathbf{g}}_\alpha\cdot\frac{\partial\delta\overline{\mathbf{x}}}{\partial\xi^\beta}\right), \tag{89}$$

$$\delta\overline{\kappa}_{\alpha\beta} = -\frac{\partial\delta\overline{\mathbf{g}}_\alpha}{\partial\xi^\beta}\cdot\mathbf{n} - \frac{\partial\overline{\mathbf{g}}_\alpha}{\partial\xi^\beta}\cdot\delta\mathbf{n} \tag{90}$$

$$= -\delta\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\mathbf{n} + \overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^\gamma\left(\mathbf{n}\cdot\delta\overline{\mathbf{g}}_\gamma\right) \tag{91}$$

$$= -\left(\delta\overline{\boldsymbol{\Gamma}}_{\alpha\beta} - \left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^\gamma\right)\delta\overline{\mathbf{g}}_\gamma\right)\cdot\mathbf{n}, \tag{92}$$

where

$$\delta\overline{\boldsymbol{\Gamma}}_{\alpha\beta} = \delta\overline{\mathbf{x}}_{,\alpha\beta}, \tag{93}$$

and the variation of the normal vector (see section "Variation of the Normal Vector" in Appendix 2) is

$$\delta\mathbf{n} = -\overline{\mathbf{g}}^{\gamma}\left(\mathbf{n}\cdot\delta\overline{\mathbf{g}}_{\gamma}\right). \tag{94}$$

3.4 Linearization for the Newton–Raphson Iterations

The linearization for the Newton–Raphson iterations is done as

$$\begin{aligned}
\delta_a\delta_b W_{\mathrm{int}} = &-\int_{\overline{\Gamma}_0}\delta_a\delta_b\overline{\varepsilon}_{\alpha\beta}\int_{(h_{\mathrm{th}})_0} S^{\alpha\beta}\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma\\
&-\int_{\overline{\Gamma}_0}\delta_a\delta_b\overline{\kappa}_{\alpha\beta}\int_{(h_{\mathrm{th}})_0}\xi^3 S^{\alpha\beta}\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\mathrm{d}\Gamma\\
&-\int_{\overline{\Gamma}_0}\delta_a\overline{\varepsilon}_{\alpha\beta}\int_{(h_{\mathrm{th}})_0}\hat{\mathbb{C}}^{\alpha\beta\gamma\delta}\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\delta_b\overline{\varepsilon}_{\gamma\delta}\mathrm{d}\Gamma\\
&-\int_{\overline{\Gamma}_0}\delta_a\overline{\varepsilon}_{\alpha\beta}\int_{(h_{\mathrm{th}})_0}\hat{\mathbb{C}}^{\alpha\beta\gamma\delta}\xi^3\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\delta_b\overline{\kappa}_{\gamma\delta}\mathrm{d}\Gamma\\
&-\int_{\overline{\Gamma}_0}\delta_a\overline{\kappa}_{\gamma\delta}\int_{(h_{\mathrm{th}})_0}\hat{\mathbb{C}}^{\alpha\beta\gamma\delta}\xi^3\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\delta_b\overline{\varepsilon}_{\gamma\delta}\mathrm{d}\Gamma\\
&-\int_{\overline{\Gamma}_0}\delta_a\overline{\kappa}_{\gamma\delta}\int_{(h_{\mathrm{th}})_0}\hat{\mathbb{C}}^{\alpha\beta\gamma\delta}\left(\xi^3\right)^2\frac{A_0}{\overline{\overline{A}}_0}\mathrm{d}\xi^3\delta_b\overline{\kappa}_{\gamma\delta}\mathrm{d}\Gamma.
\end{aligned} \tag{95}$$

The variation with subscript a is associated with the variational formulation, and the variation with subscript b is associated with the iteration linearization. Again, the variation of ξ^3 is dropped.

In this part too, we derive what we need:

$$\delta_b\delta_a\overline{\varepsilon}_{\alpha\beta} = \frac{1}{2}\left(\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\alpha}}\cdot\frac{\partial\delta_b\overline{\mathbf{x}}}{\partial\xi^{\beta}}+\frac{\partial\delta_b\overline{\mathbf{x}}}{\partial\xi^{\alpha}}\cdot\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\beta}}\right), \tag{96}$$

$$\begin{aligned}
\delta_a\delta_b\overline{\kappa}_{\alpha\beta} &= -\left(\delta_a\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}\right)\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\gamma}}\right)\cdot\delta_b\mathbf{n}+\delta_b\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}\right)\delta_a\overline{\mathbf{g}}_{\gamma}\cdot\mathbf{n}\\
&= -\left(\delta_a\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}\right)\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\gamma}}\right)\cdot\delta_b\mathbf{n}
\end{aligned} \tag{97}$$

$$+\left(\delta_b\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}+\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\delta_b\overline{\mathbf{g}}^{\gamma}\right)\delta_a\overline{\mathbf{g}}_{\gamma}\cdot\mathbf{n} \tag{98}$$

$$=\left(\delta_a\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}\right)\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\gamma}}\right)\cdot\overline{\mathbf{g}}^{\gamma}\left(\mathbf{n}\cdot\delta_b\overline{\mathbf{g}}_{\gamma}\right)$$

$$+\delta_a\overline{\mathbf{g}}_{\gamma}\cdot\mathbf{n}\left(\overline{\mathbf{g}}^{\gamma}\cdot\delta_b\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\delta}\overline{\mathbf{g}}^{\gamma}\cdot\delta_b\overline{\mathbf{g}}_{\delta}\right) \tag{99}$$

$$=\left(\delta_a\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\gamma}\right)\frac{\partial\delta_a\overline{\mathbf{x}}}{\partial\xi^{\gamma}}\right)\cdot\overline{\mathbf{g}}^{\gamma}\left(\mathbf{n}\cdot\delta_b\overline{\mathbf{g}}_{\gamma}\right)$$

$$+\left(\delta_b\overline{\boldsymbol{\Gamma}}_{\alpha\beta}-\left(\overline{\boldsymbol{\Gamma}}_{\alpha\beta}\cdot\overline{\mathbf{g}}^{\delta}\right)\frac{\partial\delta_b\overline{\mathbf{x}}}{\partial\xi^{\delta}}\right)\cdot\overline{\mathbf{g}}^{\gamma}\left(\mathbf{n}\cdot\delta_a\overline{\mathbf{g}}_{\gamma}\right). \tag{100}$$

Here, we used

$$\delta\overline{\mathbf{g}}^{\gamma}=-\overline{\mathbf{g}}^{\delta}\overline{\mathbf{g}}^{\gamma}\cdot\delta\overline{\mathbf{g}}_{\delta}, \tag{101}$$

and the proof for this can be found in Appendix 4.

4 Aorta Flow Analysis

We start with a geometry obtained from medical images and then use cubic T-splines to represent the surface. The density and kinematic viscosity of the blood are $1050\,\mathrm{kg/m^3}$ and $4.2\times10^{-6}\,\mathrm{m^2/s}$.

4.1 Conditions

The computational domain and boundary conditions are shown in Fig. 1. The diameters are given in Table 1. The inflow flow rate, plug flow, is in Fig. 2. The peak value of the average inflow velocity is $0.709\,\mathrm{m/s}$. We estimate the outflows as distributed by Murray's law [111]:

$$Q_o\propto D_o^3, \tag{102}$$

where Q_o is the volumetric outflow rate, and the outlet diameter D_o is defined based on the outlet area A_o:

$$D_o=2\sqrt{\frac{A_o}{\pi}}. \tag{103}$$

We form a plug flow profile at the smaller outlets, and the main outlet is set to traction free.

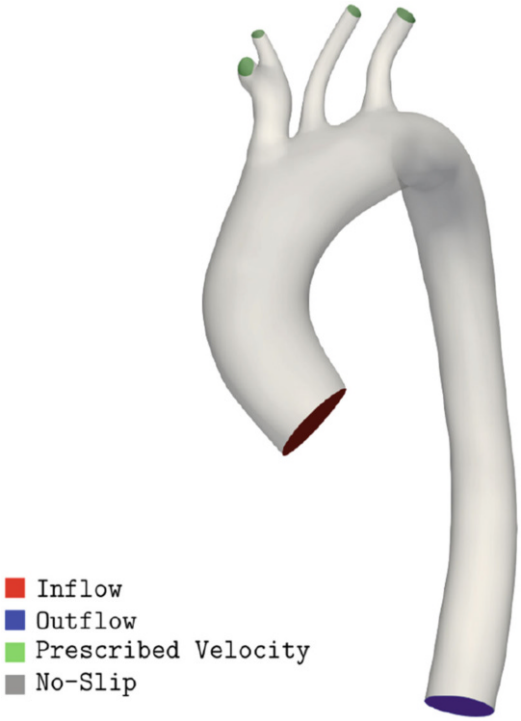


Fig. 1 Aorta flow analysis. Geometry and boundary conditions

Table 1 Aorta flow analysis

	Inlet	Outlet 1	Outlet 2	Outlet 3	Outlet 4	Outlet 5
Diameter	25.6	5.81	3.90	4.41	6.43	19.9

Diameter (mm) of the inlet and outlets. The outlets are listed in the order of closeness to the inlet

4.2 Mesh

We create a quadratic NURBS mesh from the T-spline surface, using the technique introduced in [92, 93]. Figure 3 shows one of the NURBS patches and five of the patches together to illustrate the block-structured nature of the NURBS mesh. The function space has only C^0 continuity between the patches. Figure 4 shows the base mesh. Figure 5 shows the base and refined meshes at the inlet. The meshes are refined by knot insertion, therefore the geometry is unchanged, and the basis functions for the coarser meshes are subsets of the basis functions for the finer meshes. The refinement is in the normal direction, and at each refinement, the element thickness is halved in half of the most refined layers. For the base mesh, the element thickness in the normal direction is approximately 1% of the local diameter. There is no refinement in the tangential directions. During the refinement,

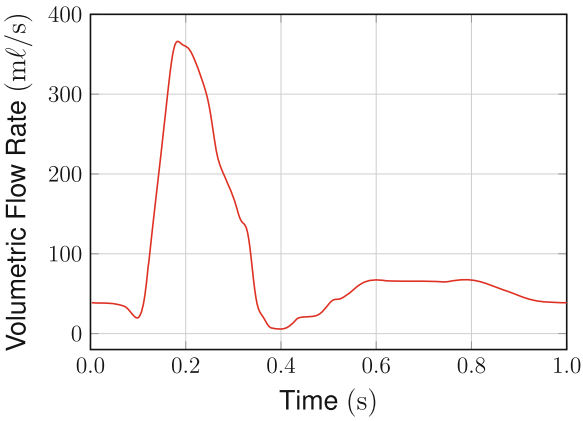


Fig. 2 Aorta flow analysis. Volumetric flow rate at the inlet

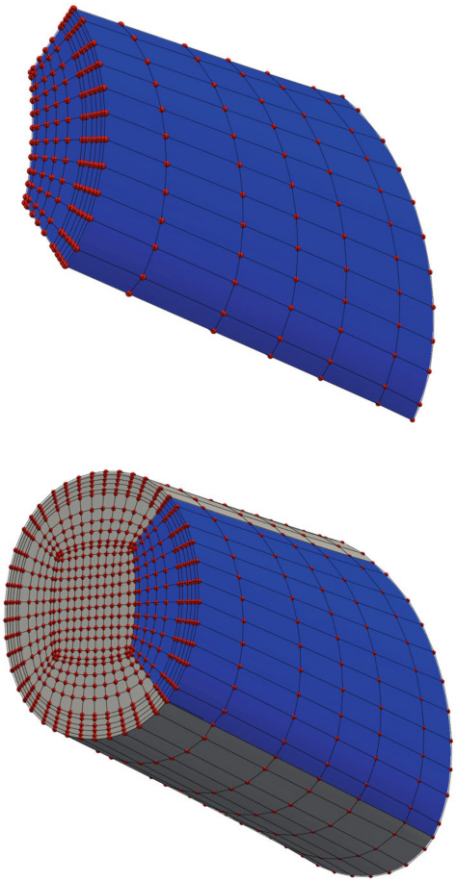


Fig. 3 Aorta flow analysis. NURBS control mesh. One of the patches (*top*) and five of the patches together (*bottom*)

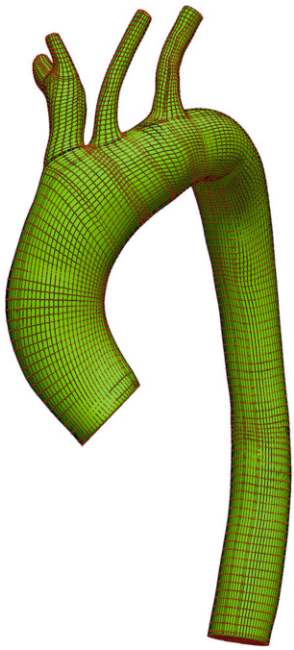


Fig. 4 Aorta flow analysis. Base Mesh. Control mesh and surface (*green*). *Red* points are control points

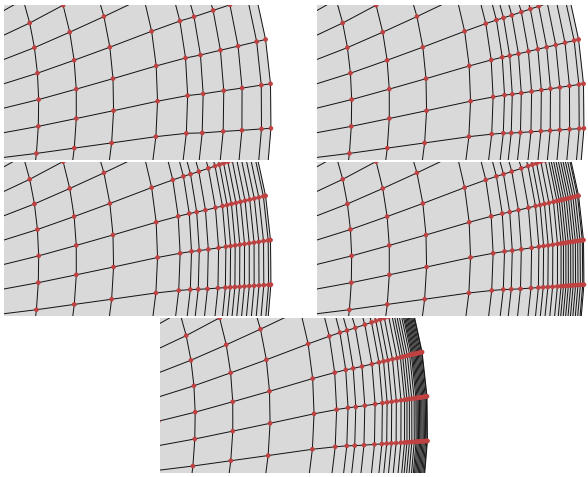


Fig. 5 Aorta flow analysis. Control mesh at the inlet. Base Mesh, Refinement Mesh 1, Refinement Mesh 2, Refinement Mesh 3, and Refinement Mesh 4 (from left to right and top to bottom)

Table 2 Aorta flow analysis

	nc	*ne*
Base Mesh	202,497	151,513
Refinement Mesh 1	266,437	205,733
Refinement Mesh 2	330,377	259,953
Refinement Mesh 3	394,317	314,173
Refinement Mesh 4	458,257	368,393

Number of control points (*nc*) and element (*ne*) for the quadratic NURBS meshes used in the computations

the original plug flow profiles of the base mesh are retained. Table 2 shows the number of elements and control points.

4.3 Mesh Refinement Study

We compute with the five meshes in Table 2. The time-step sizes are $\Delta t = 0.0025$ s for Base Mesh and Refinement Mesh 1 and 2, and $\Delta t = 0.00125$ s for Refinement Mesh 3 and 4. The number of nonlinear iterations per time step is 3, and the number of GMRES iterations per nonlinear iteration is 800 for Base Mesh and Refinement Mesh 1, and 1200, 1400, and 1600 for Refinement Mesh 2, 3, and 4, respectively. The ST-SUPS method (see Appendix 1) is used and the stabilization parameters are those given by Eqs. (2.4)–(2.6), (2.8), and (2.10) in [9].

We first compute nine cycles with Base Mesh, and the initial condition for the refined meshes is obtained by knot insertion. The solution reported here is for the 10th cycle. Figure 6 shows the solution computed with Refinement Mesh 4. At the peak flow rate a complex flow pattern is formed, and the vortex structure breaks down into smaller structures during the deceleration. The magnitude of the WSS ($\mathbf{h}_\mathrm{v}$) at the peak flow rate is shown for each mesh in Fig. 7. Qualitatively, all results are in good agreement, and the convergence can be seen with refinement. To quantify the mesh refinement level, we calculate the y^+ value for the first-element thickness h as

$$y^+ = \frac{u^* h}{\nu}, \tag{104}$$

where the friction velocity u^* is based on the computed value of the WSS as follows:

$$u^* = \sqrt{\frac{\|\mathbf{h}_\mathrm{v}^h\|}{\rho}}. \tag{105}$$

Figure 8 shows the spatial distribution of y^+ at the peak flow rate. It shows that for the meshes used here, y^+ range is from approximate maximum 10 to less than 1. Comparing Figs. 7 and 8, we see that the WSS values computed over different meshes are in agreement where $y^+ \leq 1$.

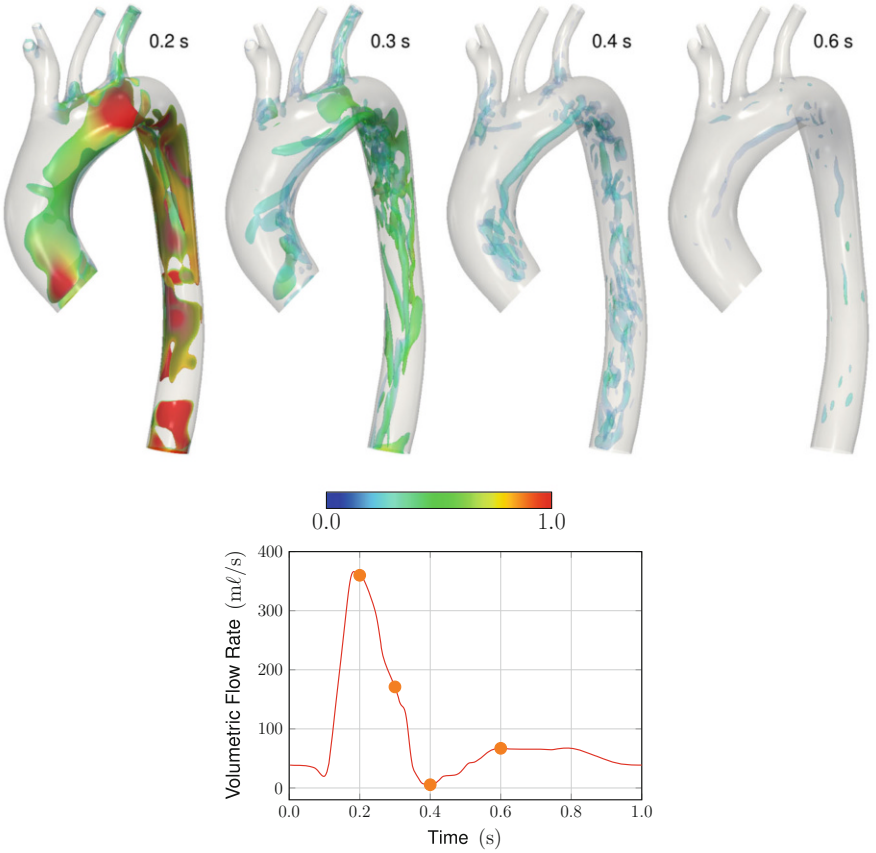


Fig. 6 Aorta flow analysis. Mesh refinement study. Computed with Refinement Mesh 4. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s) (*top*). The time instants are shown with circles (*bottom*)

The time-averaged WSS magnitude (TAWSS) is shown in Fig. 9, and Fig. 10 shows the spatially averaged WSS magnitude in a cycle. Figure 11 shows the oscillatory shear index (OSI), defined as

$$\mathrm{OSI} = \frac{1}{2}\left(1 - \frac{\left\|\int_0^T \mathbf{h}_{\mathrm{v}}^h \mathrm{d}t\right\|}{\int_0^T \left\|\mathbf{h}_{\mathrm{v}}^h\right\| \mathrm{d}t}\right). \tag{106}$$

Overall for OSI, even Base Mesh is in a good agreement with others. However, if we compare details such as branches, we see some difference even where y^+ value is small. To see the flow differences, using the solution from Refinement Mesh 4 as the reference solution, we inspect the velocity difference $\left\|\mathbf{u}_k^h - \mathbf{u}_4^h\right\|$, where the subscripts indicate Base Mesh and Refinement Mesh k.

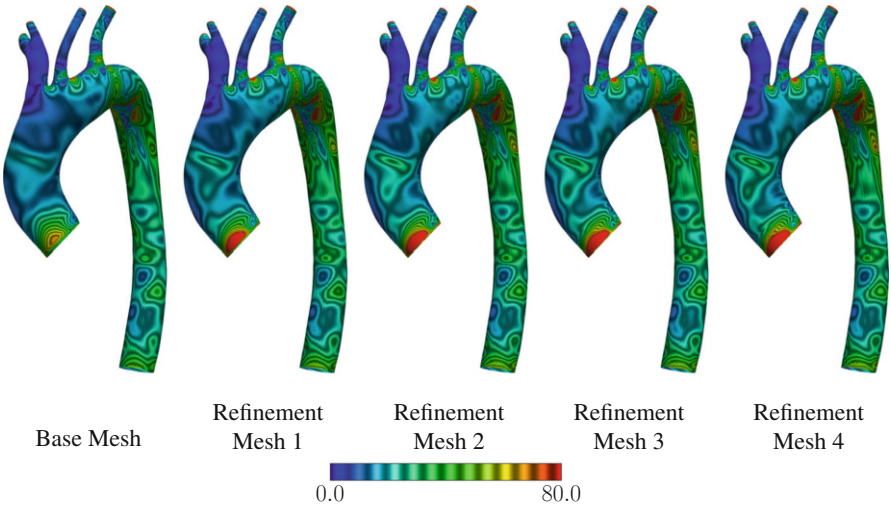


Fig. 7 Aorta flow analysis. Mesh refinement study. WSS (dyn/cm^2) at the peak flow rate

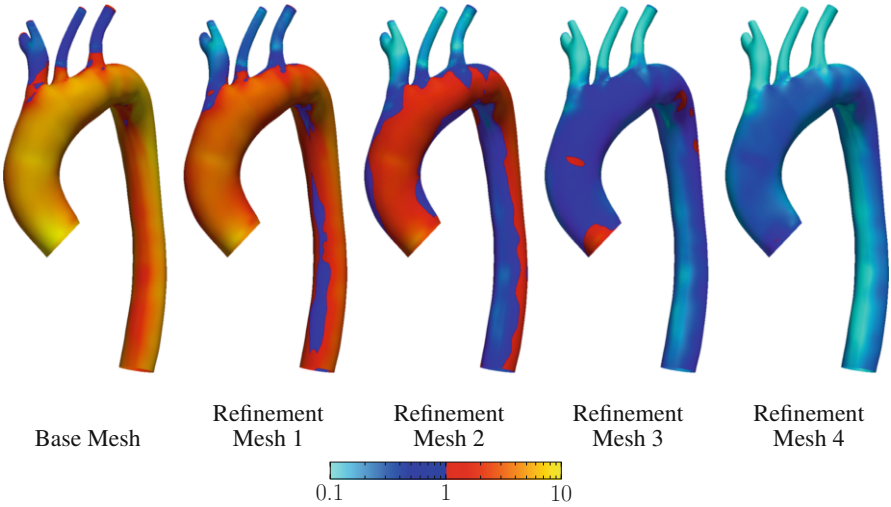


Fig. 8 Aorta flow analysis. Mesh refinement study. y^+ value for the first-element thickness, based on the WSS computed at the peak flow rate

Remark 3 To calculate the velocity difference, all meshes and corresponding solutions are refined by using the knot-insertion technique, and the control variables are obtained based on Refinement Mesh 4. The visualization is done after taking the difference between the control variables, interpolating the vector, and taking its magnitude.

The spatial average of the difference is maximum at around 0.5 s. This indicates that the vortex breakdown, due to the small-scale flow behavior that needs to be

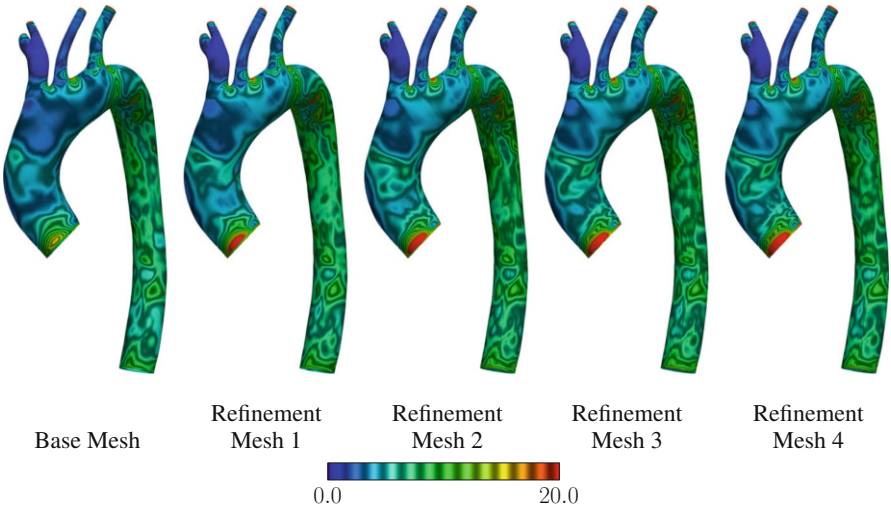


Fig. 9 Aorta flow analysis. Mesh refinement study. TAWSS (dyn/cm^2)

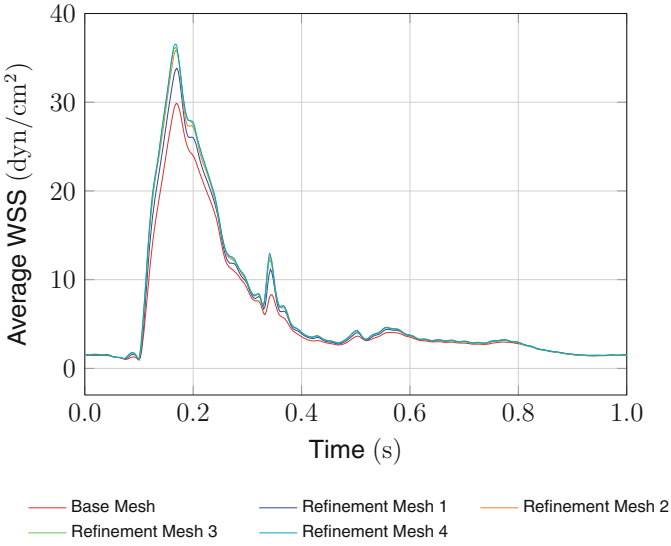


Fig. 10 Aorta flow analysis. Mesh refinement study. Spatially averaged WSS during a cycle

dealt with, would not be easy to resolve. Figure 12 shows the velocity difference at 0.5 s.

In summary, good accuracy in the WSS magnitude can be obtained with locally good representation, and the OSI requires a good flow representation overall, including the vortex breakdown.

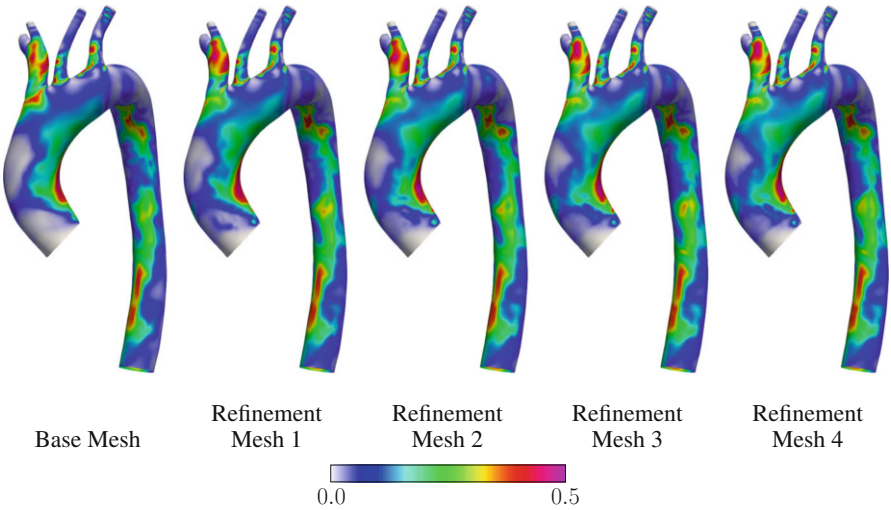


Fig. 11 Aorta flow analysis. Mesh refinement study. OSI

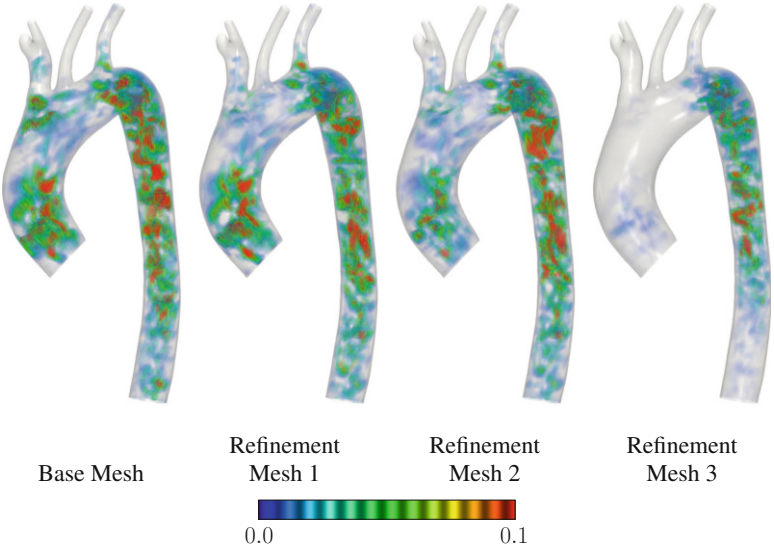


Fig. 12 Aorta flow analysis. Mesh refinement study. Velocity difference $\left\|\mathbf{u}_k^h - \mathbf{u}_4^h\right\|$ (m/s) at 0.5 s, where the subscripts indicate Base Mesh and Refinement Mesh k

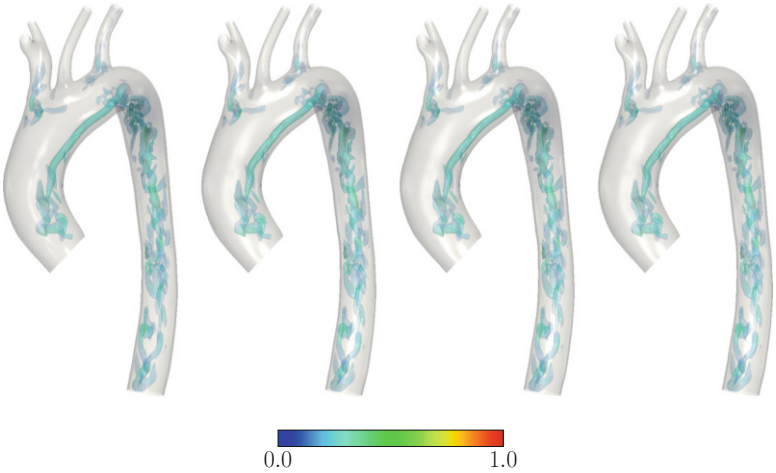


Fig. 13 Aorta flow analysis. Periodicity study. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s). At 0.4 s in cycles 7, 8, 9, and 10

4.4 Periodicity Study

In this study, we check to see if we reach flow periodicity in the cardiac cycles. We use the base mesh. Figure 13 shows, for cycles 7–10, the vortex structure at 0.4 s, and Figs. 14 and 15 show the TAWSS and OSI for those cycles. Despite the complex, unsteady behavior, we do reach flow periodicity in the computations, already in the 7th cycle.

5 Shell Tests

We test the formulation given in Sect. 3 by using pressurized cylindrical and spherical shells. The tests include shell models with compressible and incompressible material and continuum models with compressible material.

5.1 Constitutive Models

We test two constitutive models: neo-Hookean and Fung's. The elastic-energy density functions are

$$\varphi_{\mathrm{NH}}\left(\mathbf{C}\right) = \frac{1}{2}\mu\left(\mathrm{tr}\mathbf{C} - n_{\mathrm{sd}}\right), \tag{107}$$

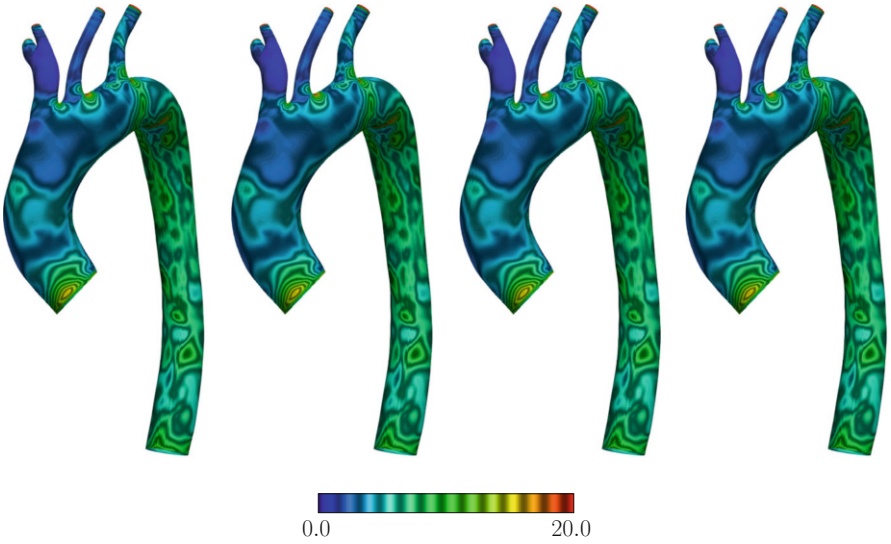


Fig. 14 Aorta flow analysis. Periodicity study. TAWSS in cycles 7, 8, 9, and 10

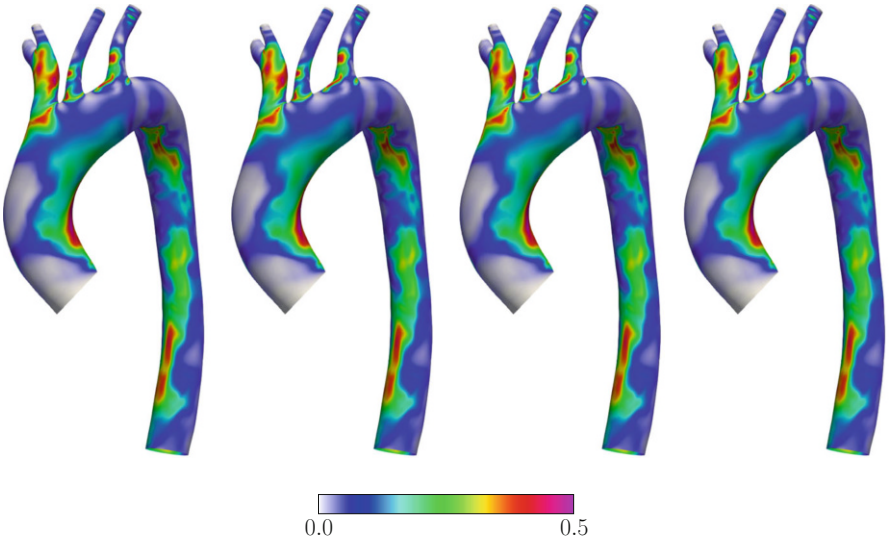


Fig. 15 Aorta flow analysis. Periodicity study. OSI in cycles 7, 8, 9, and 10

$$\varphi_{\mathrm{F}}(\mathbf{C}) = D_1 \left(e^{(D_2 \mathrm{tr}\mathbf{C} - n_{\mathrm{sd}})} - 1 \right), \tag{108}$$

where μ is the shear modulus, and D_1 and D_2 are the coefficients of the Fung's model.

For incompressible material, we use

$$\varphi_{\mathrm{NHI}}(\mathbf{C}) = \varphi_{\mathrm{NH}}(\mathbf{C}) + p(J-1), \tag{109}$$

$$\varphi_{\mathrm{FI}}(\mathbf{C}) = \varphi_{\mathrm{F}}(\mathbf{C}) + p(J-1), \tag{110}$$

where p is the pressure, which can be eliminated by the plane stress condition.

For compressible material, we use

$$\varphi_{\mathrm{NHR}}(\mathbf{C}) = \varphi_{\mathrm{NH}}\left(J^{-\frac{2}{n_{\mathrm{sd}}}}\mathbf{C}\right) + \varphi_{\mathrm{vol}}(J), \tag{111}$$

$$\varphi_{\mathrm{FR}}(\mathbf{C}) = \varphi_{\mathrm{F}}\left(J^{-\frac{2}{n_{\mathrm{sd}}}}\mathbf{C}\right) + \varphi_{\mathrm{vol}}(J), \tag{112}$$

where

$$\varphi_{\mathrm{vol}}(J) = \frac{1}{2}\kappa\left(\frac{1}{2}\left(J^2-1\right) - \ln J\right), \tag{113}$$

and κ is the bulk modulus.

5.2 Test Computations

The pressure, applied at $r = r_p$, is normalized by the shear modulus (at the zero-stress state):

$$p^* = \frac{p}{\mu} \tag{114}$$

for the neo-Hookean model,

$$p^* = \frac{p}{2D_1D_2} \tag{115}$$

for the Fung's model, and we use $D_2 = 8.365$. We determine the bulk modulus from the Poisson's ratio ν as follows:

$$\kappa = \frac{2\mu(1+\nu)}{3(1-2\nu)} \tag{116}$$

for the neo-Hookean model, and

$$\kappa = \frac{4D_1D_2(1+\nu)}{3(1-2\nu)} \tag{117}$$

for the Fung's model.

How to deal with pressure acting on the inner surface is not easy because the midsurface is the geometry we are using in the computation. Here we propose two ways. In the first one, "midsurface model," the pressure is applied on the midsurface of the current configuration. In the second one,"innersurface model," the structure "midsurface" is moved to the inner surface and the pressure is applied there.

Remark 4 In applying the pressure, the midsurface model is physically wrong, especially when the thickness is significant. The innersurface model will have larger absolute value for ξ^3, which would lead to larger discretization errors.

In the test cases, we use the inner and outer radii R_I and R_O, and the thickness $H = R_\mathrm{O} - R_\mathrm{I}$. The condition used here is $\frac{H}{2R_\mathrm{I}} = 0.1$, which is slightly thinner than most arteries. To have a reference solution to compare the results to, we provide in Appendix 5 the second Piola–Kirchhoff stress tensor expressed in terms of the principal stretches. The results are compared by inspecting pressure as a function of stretch. The stretch is $\overline{\lambda_1} \equiv \frac{r_p}{R}$ for the midsurface model, where $r_p = \overline{r}$, and $\overline{\lambda_1} \equiv \frac{r_p}{R_\mathrm{I}}$ for the innersurface model, where $r_p = r_\mathrm{I}$. In the computations, we increase the pressure gradually in obtaining the solution and calculate the stretch. In obtaining the reference solutions, we use numerical integrations, which are explained in the following subsections.

5.2.1 Pressurized Cylinder

We use orthogonal basis vectors: the first basis vector is in the radial direction, the second one is in the cylinder axis direction, and the third one is normal to the surface. The force equilibrium gives the following relationship:

$$p2r_p = 2\int_{r_\mathrm{I}}^{r_\mathrm{O}} \sigma_{11}\mathrm{d}x_3 \tag{118}$$

$$= 2\int_{r_\mathrm{I}}^{r_\mathrm{O}} J^{-1}\lambda_1^2 S_{11}\mathrm{d}x_3 \tag{119}$$

$$= 2\int_{R_\mathrm{I}}^{R_\mathrm{O}} J^{-1}\lambda_1^2 S_{11}\lambda_3\mathrm{d}X_3 \tag{120}$$

$$= 2\int_{R_\mathrm{I}}^{R_\mathrm{O}} \frac{\lambda_1}{\lambda_2} S_{11}\mathrm{d}X_3. \tag{121}$$

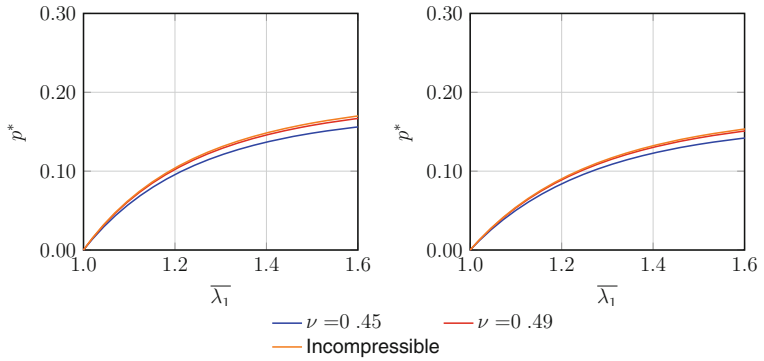


Fig. 16 Pressurized cylinder. Neo-Hookean model. Reference solutions. Midsurface model (*left*) and innersurface model (*right*)

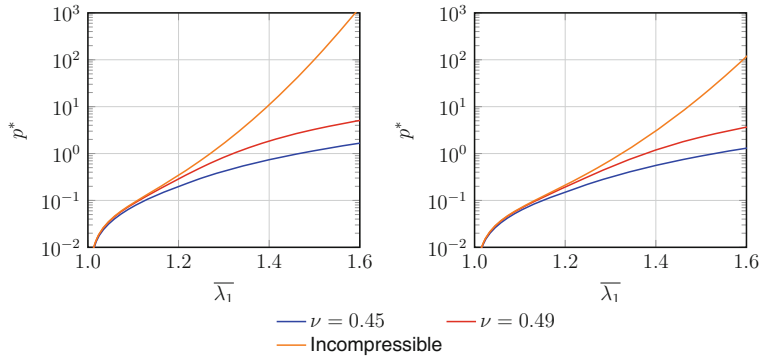


Fig. 17 Pressurized cylinder. Fung's model. Reference solutions. Midsurface model (*left*) and innersurface model (*right*). Note that the pressure scale is logarithmic

Because the cylinder height does not change, $\lambda_2 = 1$, and we obtain

$$p = \frac{1}{r_p} \int_{R_\mathrm{I}}^{R_\mathrm{O}} \lambda_1 S_{11} \mathrm{d}X_3. \tag{122}$$

See section "Cylinder" in Appendix 5 for S_{11}. Figures 16 and 17 show the reference solutions for the neo-Hookean and Fung's models. We compute, in 2D, with uniform, periodic cubic B-splines with eight elements. For comparison purposes, we also compute with the continuum model, using 128 uniform, periodic cubic B-spline elements in the circumferential direction, and one element in the radial direction.

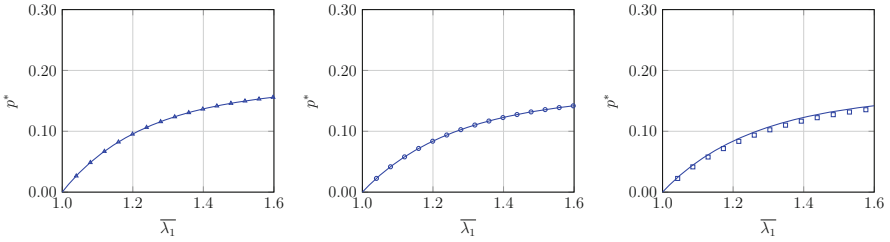


Fig. 18 Pressurized cylinder. Neo-Hookean model ($\nu = 0.45$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

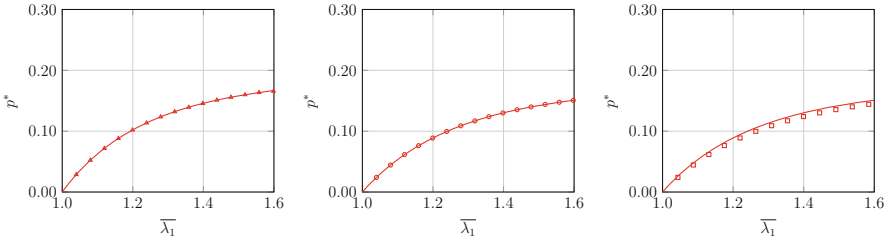


Fig. 19 Pressurized cylinder. Neo-Hookean model ($\nu = 0.49$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

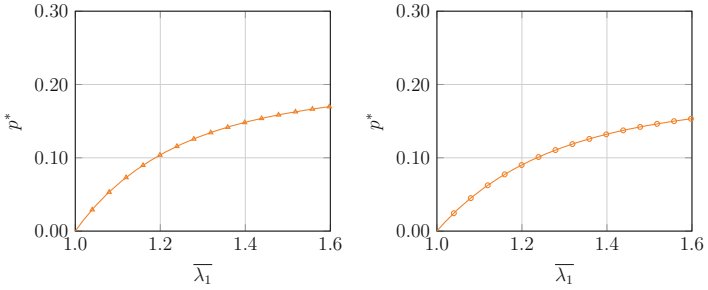


Fig. 20 Pressurized cylinder. Neo-Hookean model (incompressible). Midsurface model (*left*), and innersurface model (*right*). Solid curve is the reference solution

Figures 18, 19, and 20 show the solutions for the neo-Hookean model, and Figs. 21, 22, and 23 for the Fung's model. Figures 18, 21 and 19, 22 show, for $\nu = 0.45$ and 0.49, the solutions from the midsurface, innersurface, and continuum models. Figures 20 and 23 show, for incompressible material, the solutions from the midsurface and innersurface models.

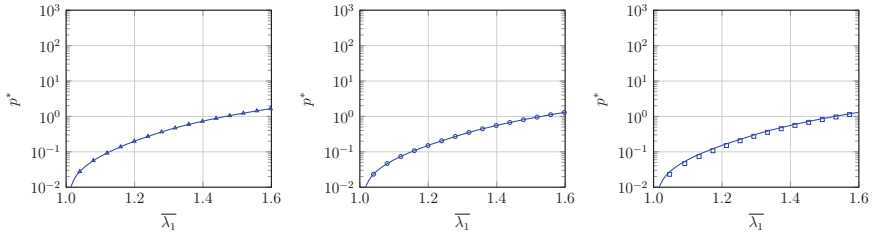


Fig. 21 Pressurized cylinder. Fung's model ($\nu = 0.45$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

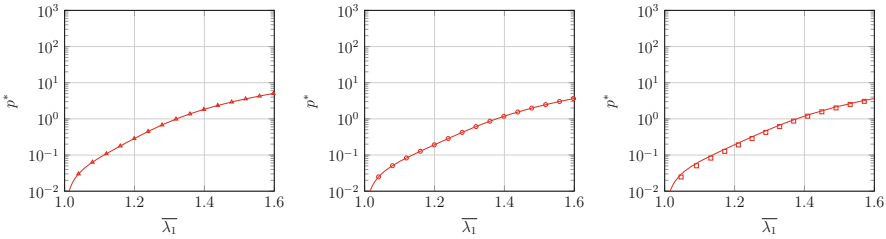


Fig. 22 Pressurized cylinder. Fung's model ($\nu = 0.49$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

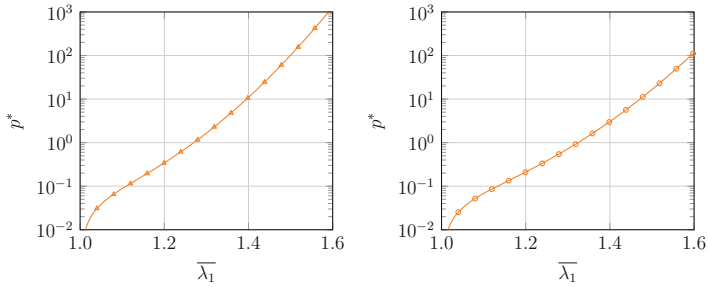


Fig. 23 Pressurized cylinder. Fung's model (incompressible). Midsurface model (*left*), and innersurface model (*right*). Solid curve is the reference solution

5.2.2 Pressurized Sphere

We use orthogonal basis vectors: the first two vectors are on the surface, and the third vector is normal to the surface. The force equilibrium gives the following relationship:

$$p\pi r_p^2 = \int_{r_\mathrm{I}}^{r_\mathrm{O}} 2\pi x_3 \sigma_{11} \mathrm{d}x_3 \tag{123}$$

$$= \int_{r_\mathrm{I}}^{r_\mathrm{O}} 2\pi J^{-1} \lambda_1^2 x_3 S_{11} \mathrm{d}x_3 \tag{124}$$

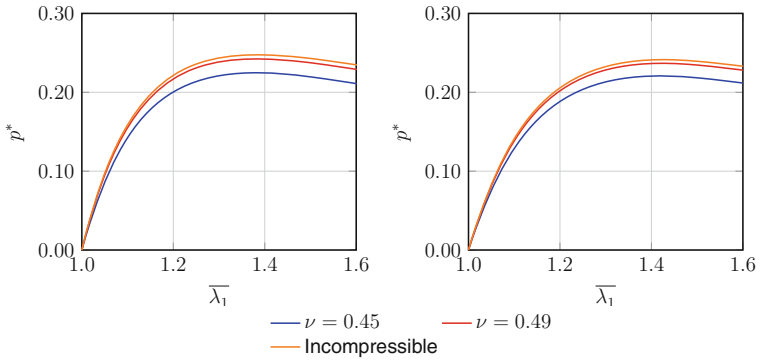


Fig. 24 Pressurized sphere. Neo-Hookean model. Reference solutions. Midsurface model (*left*) and innersurface model (*right*)

$$= \int_{R_I}^{R_O} 2\pi J^{-1}\lambda_1^2 \underbrace{\lambda_1 X_3}_{x_3} S_{11}\lambda_3 \mathrm{d}X_3 \tag{125}$$

$$= \int_{R_I}^{R_O} 2\pi \frac{\lambda_1^2}{\lambda_2} X_3 S_{11} \mathrm{d}X_3. \tag{126}$$

Because of the symmetry between the two basis vector directions on the surface, $\lambda_1 = \lambda_2$, and we obtain

$$p = \frac{2}{r_p^2} \int_{R_I}^{R_O} X_3 \lambda_1 S_{11} \mathrm{d}X_3. \tag{127}$$

See section “Sphere" in Appendix 5 for S_{11}. Figures 24 and 25 show the reference solutions for the neo-Hookean and Fung’s models.

We compute, in 3D, with a cubic T-spline mesh, which consists of 296 control points and 534 Bézier elements (see Fig. 26). For comparison purposes, we also compute with the continuum model, which is extruded in the thickness direction with one element.

Remark 5 The number of elements used in the integration is the number of Bézier elements, which is 534 in this case. The mesh was generated by a commercial software, Rhinoceros with the T-splines plug-in. It actually has, in the finite element sense, 294 elements.

Figures 27, 28, and 29 show the solutions for the neo-Hookean model, and Figures 30, 31, and 32 for the Fung’s model. Figures 27, 30 and 28, 31 show, for

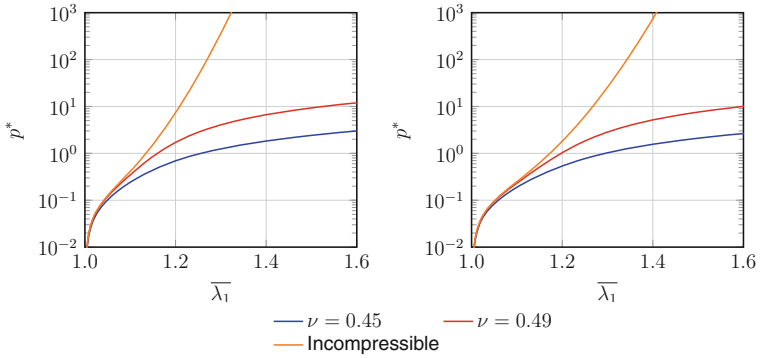
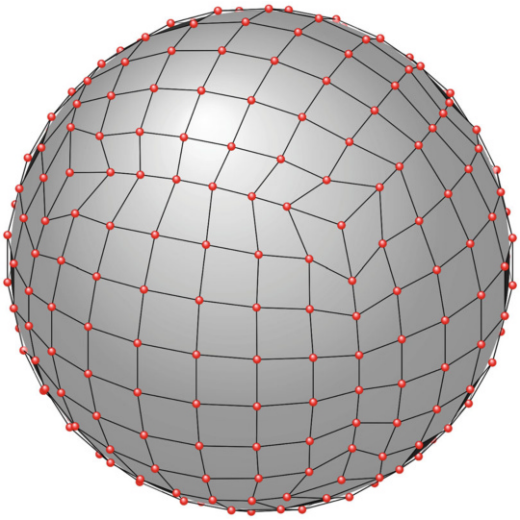


Fig. 25 Pressurized sphere. Fung's model. Reference solutions. Midsurface model (*left*) and innersurface model (*right*). Note that the pressure scale is logarithmic

Fig. 26 Pressurized sphere. T-spline mesh used in the computations. *Red* circles represent the control points



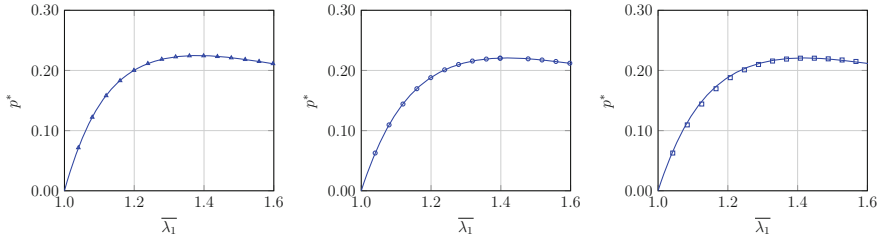


Fig. 27 Pressurized sphere. Neo-Hookean model ($\nu = 0.45$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

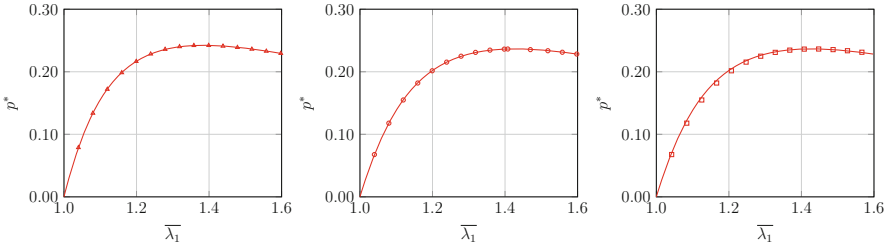


Fig. 28 Pressurized sphere. Neo-Hookean model ($\nu = 0.49$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

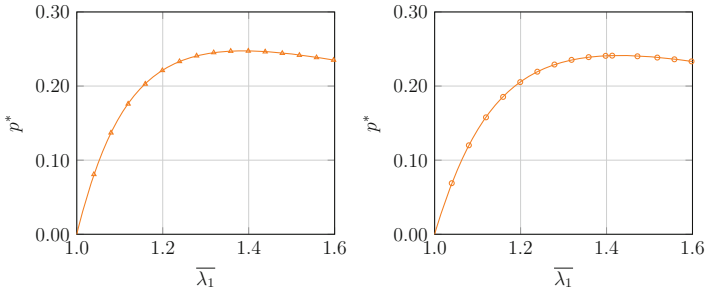


Fig. 29 Pressurized sphere. Neo-Hookean model (incompressible). Midsurface model (*left*), and innersurface model (*right*). Solid curve is the reference solution

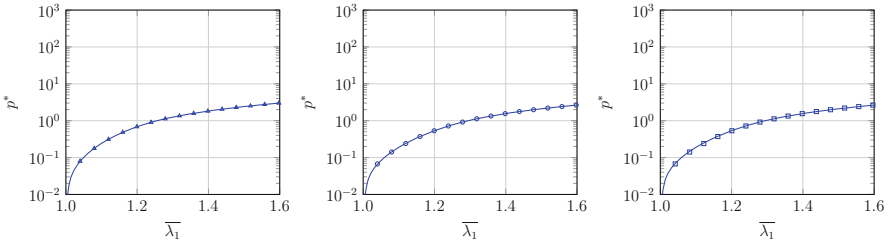


Fig. 30 Pressurized sphere. Fung's model ($\nu = 0.45$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

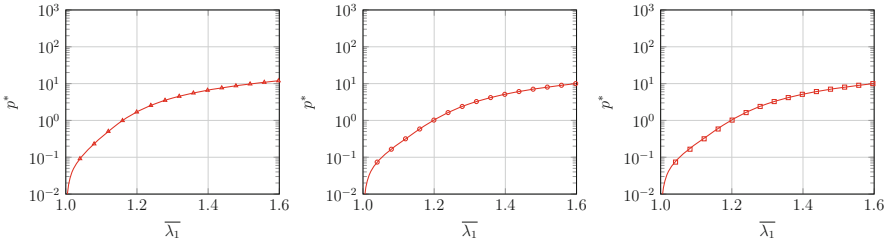


Fig. 31 Pressurized sphere. Fung's model ($\nu = 0.49$). Midsurface model (*left*), innersurface model (*center*), and continuum model (*right*). Solid curve is the reference solution

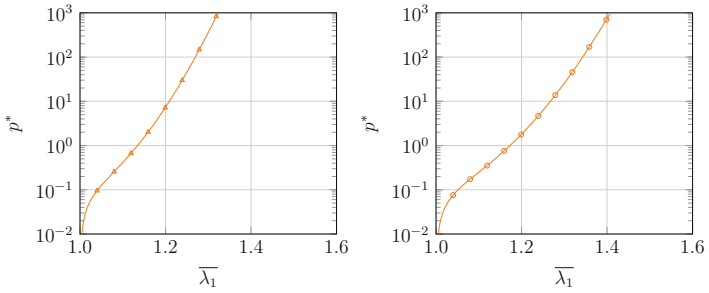


Fig. 32 Pressurized sphere. Fung's model (incompressible). Midsurface model (*left*), and innersurface model (*right*). Solid curve is the reference solution

$\nu = 0.45$ and 0.49, the solutions from the midsurface, innersurface, and continuum models. Figures 29 and 32 show, for incompressible material, the solutions from the midsurface and innersurface models.

6 Heart Valve

The heart valve we compute here is the aortic valve; there are three valve leaflets. We first perform a structural mechanics computation with a prescribed, time-periodic, spatially uniform pressure difference between the upper and lower surfaces of the leaflets. Based on the leaflet deformation coming from that computation, we next perform the flow computation with the ST-SI-TC-IGA.

6.1 Structural Mechanics

We use the Fung's model, with $D_1 = 2.0611 \times 10^4$ Pa and $D_2 = 8.365$. The thickness and density are 0.0386 cm and 1000 kg/m^3. We do the computation for only one of the leaflets. The contact between the leaflet and the periodicity planes on its two sides represent the contact with the other leaflets. The contact is handled by not allowing the control points cross the planes. The pressure difference we impose is given in Fig. 33. The profile is similar to the one used in [102]. We use a cubic T-spline mesh. The number of control points and number of elements are 1349 and 1236. We add a mass-proportional damping, with damping coefficient $2.041 \times 10^3\,\mathrm{s}^{-1}$. The generalized-$\alpha$ method with the parameters based on the spectral radius, which we set to zero, is used with a time-step size of 0.5 ms. The number of nonlinear iterations per time step is 6, and the number of GMRES iterations per nonlinear iteration is 100. The shell model we use in this computation is an earlier version of the one described in Sect. 3. In taking λ_3 into account in the curvature term, we use its midsurface value. Figure 34 shows the results for the first half of the cycle. For the rest of the cycle, the leaflet shape does not change much.

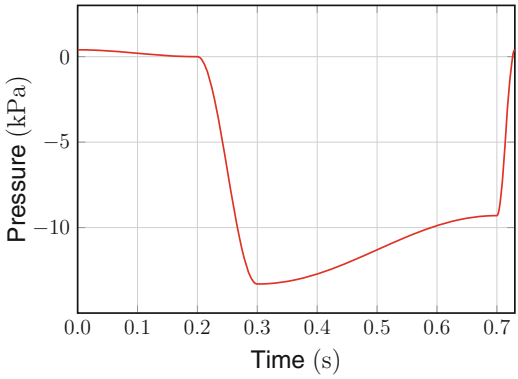


Fig. 33 Heart valve. Pressure difference. The profile is similar to the one used in [102]

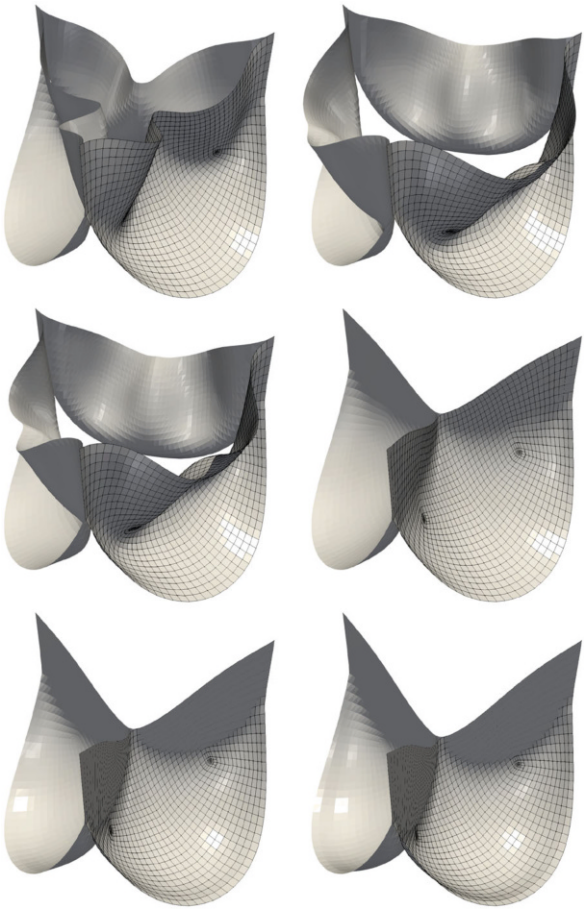


Fig. 34 Heart valve structural mechanics computation. The time instants are 0.036, 0.097, 0.158, 0.219, 0.281, 0.342 s (from *left* to *right* and *top* to *bottom*). The lines are T-spline element boundaries

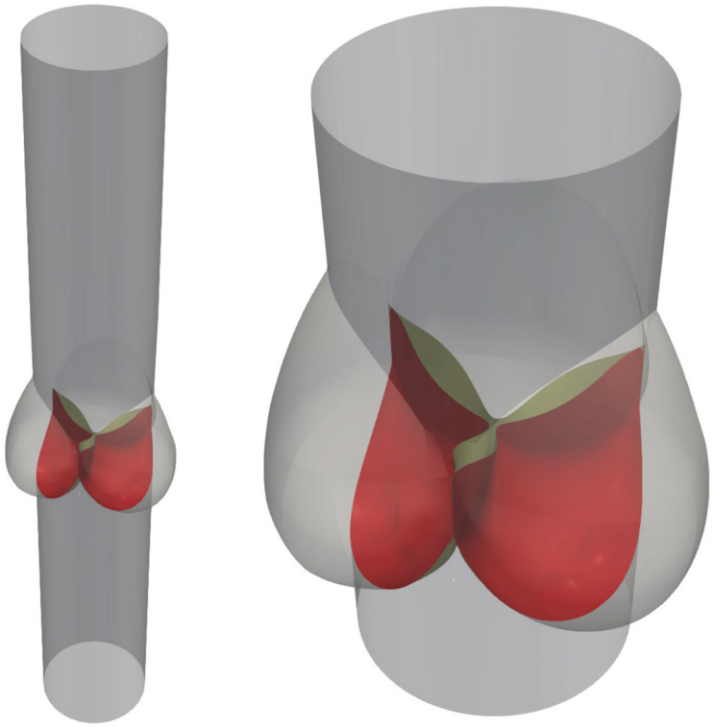


Fig. 35 Heart valve fluid mechanics computation. Computational domain (*left*) and zoomed view (*right*)

6.2 Fluid Mechanics

The fluid mechanics domain is shown in Fig. 35. The mesh is made of 231 quadratic NURBS patches, and it has five SIs. Three of the SIs connect the mesh sectors containing the leaflets in the valve region of the aorta, and the other two, which are the top and bottom circular planes, connect the meshes in the inlet and outlet regions to the valve region. The number of control points and number of elements are 584,964 and 438,000. The motion of the fluid boundaries on the valve surfaces are prescribed based on the structural mechanics computation. To do that, the displacement on the cubic T-spline mesh is projected to the boundary of the fluid mechanics mesh. Figure 36 shows the T-spline and NURBS surfaces. Figures 37 and 38 show the moving mesh. The mesh motion and master–slave relationship [5, 11] are all prepared manually based on the position of the boundary. The ST-SUPS method (see Appendix 1) is used, and the stabilization parameters are those given by Eqs. (2.4)–(2.6), (2.8) and (2.10) in [9]. The time-step size is 0.00324 s. The number of nonlinear iterations per time step is 3, and the number of GMRES iterations per nonlinear iteration is 300. Figures 39 and 40 show the computational results.

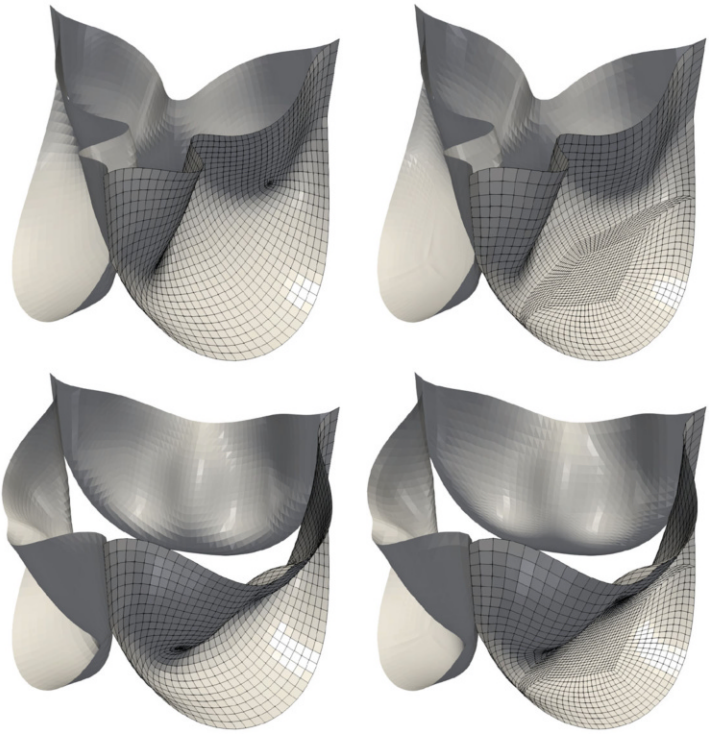


Fig. 36 Heart valve structure solution (*left*) and the fluid mechanics boundary obtained by projection from the structure (*right*). The lines are T-spline and NURBS element boundaries

7 Concluding Remarks

We have presented our computational methods for aorta flow analysis and heart valve flow and structure analysis. We have also presented results from the computations carried out with these methods. The computational challenges addressed in the aorta flow analysis included unsteady flow through a curved geometry with multiple outlets, relatively high Reynolds numbers, and multiscale flow behavior. The key targets were smoother representation of the geometry and increased accuracy in the flow solution. In the heart valve flow analysis, the challenges addressed included unsteady flow through a complex geometry, solid surfaces with large motion, and contact between the valve leaflets. The main target was computation with the leaflet deformation coming from a structure analysis. In the flow computations, the core method was the ST-VMS method, and the other key methods were the ST-SI and ST-TC methods and the ST-IGA. For the structure analysis, we have presented a Kirchhoff–Love shell model, where we take the stretch in the third direction into account in calculating the curvature term.

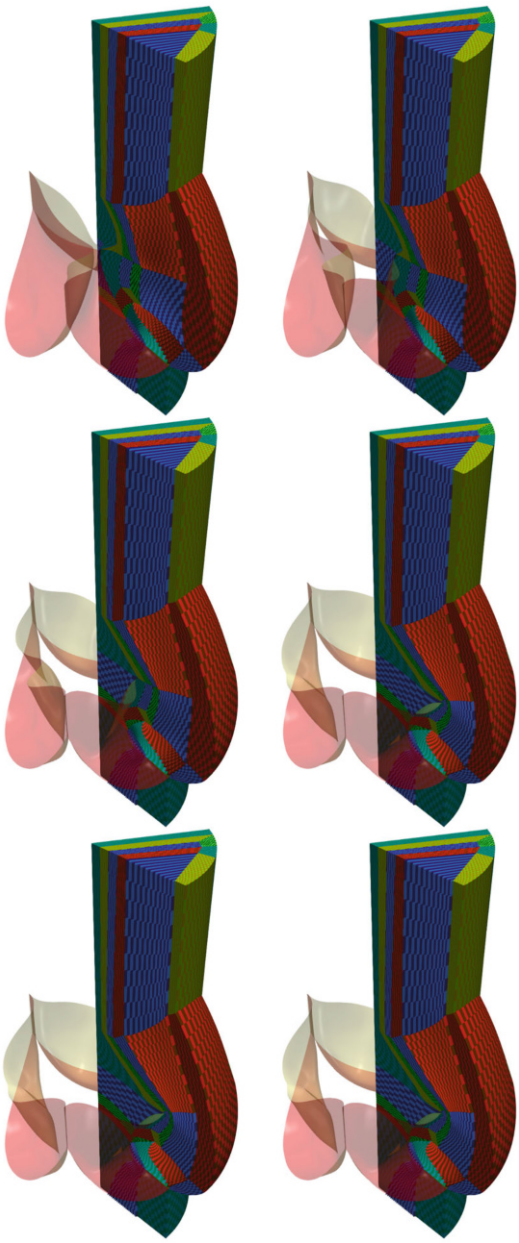


Fig. 37 Heart valve fluid mechanics computation. A set of selected NURBS elements, at $t =$ 0.000, 0.020, 0.039, 0.059, 0.078, 0.098 s (from *left* to *right* and *top* to *bottom*)

The ST framework, in a general context, provides higher-order accuracy. The VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flows. The moving-mesh feature of the ST framework enables high-resolution computation near moving solid surfaces. The ST-SI connects the sectors of meshes containing different moving solid surfaces,

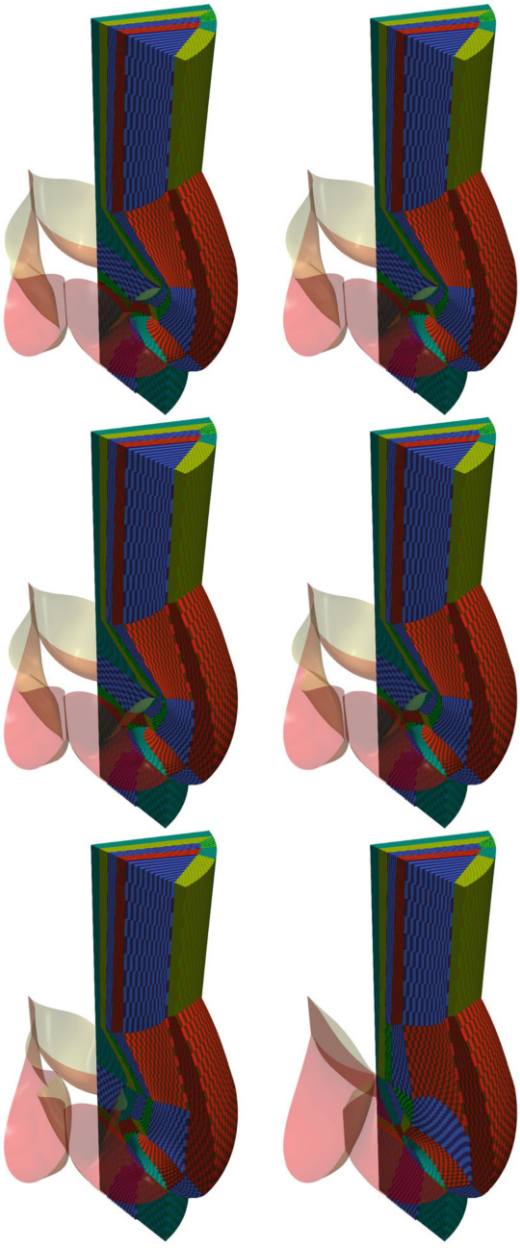


Fig. 38 Heart valve fluid mechanics computation. A set of selected NURBS elements, at $t =$ 0.117, 0.137, 0.156, 0.176, 0.195, 0.215 s (from *left* to *right* and *top* to *bottom*)

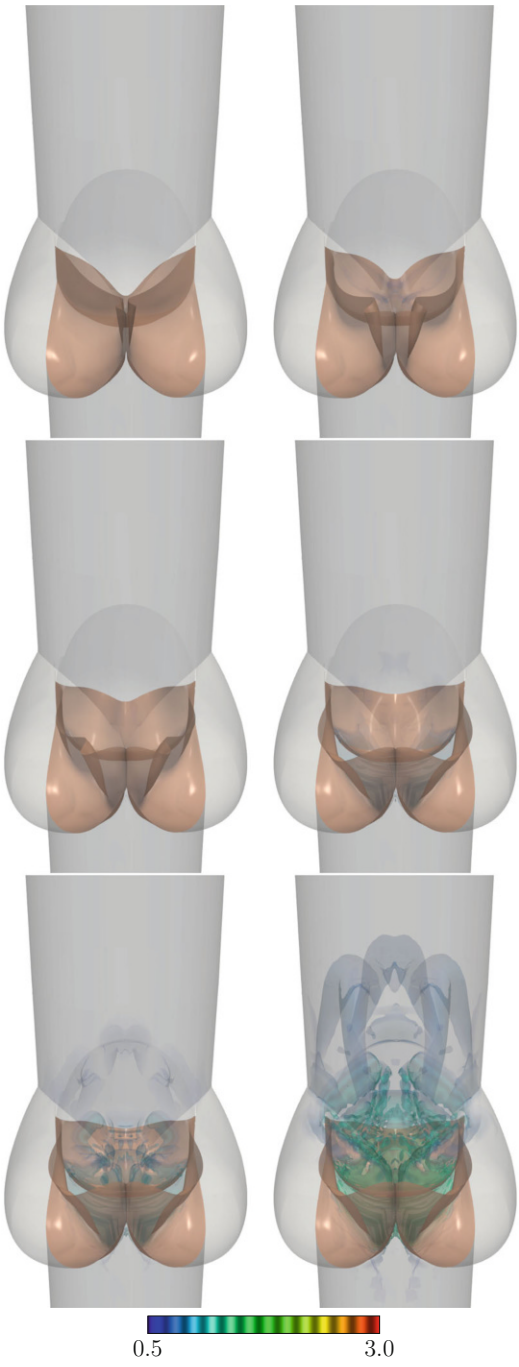


Fig. 39 Heart valve fluid mechanics computation. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s). The time instants are the same as those in Fig. 37

Fig. 40 Heart valve fluid mechanics computation. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s). The time instants are the same as those in Fig. 38

such as the valve leaflets, enabling a more effective mesh moving. The ST-TC enables moving-mesh computation even with the TC created by the contact between the leaflets. It deals with the contact while maintaining high-resolution representation near the leaflets. Integration of the ST-SI and ST-TC enables high-resolution representation even when parts of the SI coincide with the leaflet surfaces. It also enables dealing with leaflet–leaflet contact location change and contact sliding. The ST-IGA provides smoother representation of aorta and valve surfaces and increased accuracy in the flow solution. With the integration of the ST-IGA with the ST-SI and ST-TC, the element density in the narrow spaces near the contact areas is kept at a reasonable level.

In the aorta flow analysis, we used a geometry obtained from medical images and conducted studies to systematically determine the mesh refinement influence and to assess flow periodicity in the cardiac cycles. To evaluate the performance of the shell model presented, we conducted structural mechanics test computations with cylindrical and spherical geometries and compared the results to near-analytical reference solutions. In the heart valve analysis, we first performed a structural mechanics computation with a prescribed, time-periodic, spatially uniform pressure difference between the upper and lower surfaces of the valve leaflets. Based on the leaflet deformation coming from that computation, we next performed the flow computation. The computations presented demonstrate the scope and effectiveness of the methods.

Acknowledgements This work was supported in part by JST-CREST; Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Grant-in-Aid for Scientific Research (A) 18H04100 from Japan Society for the Promotion of Science; and Rice–Waseda research agreement. This work was also supported in part by Grant-in-Aid for JSPS Research Fellow 17J11096 (fourth author) and 18J14680 (fifth author). The mathematical model and computational method parts of the work were also supported (second author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University.

Appendix 1: ST-VMS and ST-SI Methods

We include from [9, 14] the ST-VMS and ST-SI methods.

The ST-VMS method is given as

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

$$+\sum_{e=1}^{(n_{\rm el})_n}\int_{Q_n^e}\nu_{\rm LSIC}\nabla\cdot\mathbf{w}^h\rho r_{\rm C}(\mathbf{u}^h)\mathrm{d}Q$$

$$-\sum_{e=1}^{(n_{\rm el})_n}\int_{Q_n^e}\tau_{\rm SUPS}\mathbf{w}^h\cdot\left(\mathbf{r}_{\rm M}(\mathbf{u}^h,p^h)\cdot\nabla\mathbf{u}^h\right)\mathrm{d}Q$$

$$-\sum_{e=1}^{(n_{\rm el})_n}\int_{Q_n^e}\frac{\tau_{\rm SUPS}^2}{\rho}\mathbf{r}_{\rm M}(\mathbf{u}^h,p^h)\cdot\left(\nabla\mathbf{w}^h\right)\cdot\mathbf{r}_{\rm M}(\mathbf{u}^h,p^h)\mathrm{d}Q=0, \tag{128}$$

where

$$\mathbf{r}_{\rm M}(\mathbf{u}^h,p^h)=\rho\left(\frac{\partial\mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h-\mathbf{f}^h\right)-\nabla\cdot\boldsymbol{\sigma}(\mathbf{u}^h,p^h), \tag{129}$$

$$r_{\rm C}(\mathbf{u}^h)=\nabla\cdot\mathbf{u}^h \tag{130}$$

are the residuals of the momentum equation and incompressibility constraint. The test functions associated with the velocity and pressure are $\mathbf{w}$ and q. A superscript "h" indicates that the function is coming from a finite-dimensional space. The symbol Q_n represents the ST slice between time levels n and $n+1$, $(P_n)_{\rm h}$ is the part of the lateral boundary of that slice associated with the traction boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and $n_{\rm el}$ is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level. See [8, 9, 16, 17, 79] for the definitions used here for the stabilization parameters $\tau_{\rm SUPS}$ and $\nu_{\rm LSIC}$. For more ways of calculating the stabilization parameters in finite element computation of flow problems, see [112–133].

Remark 6 The ST-SUPS method can be obtained from the ST-VMS method by dropping the eighth and ninth integrations.

In the ST-SI method, labels "Side A" and "Side B" represent the two sides of the SI. We add boundary terms to Eq. (128). The boundary terms are first added separately for the two sides, using test functions $\mathbf{w}_{\rm A}^h$ and $q_{\rm A}^h$ and $\mathbf{w}_{\rm B}^h$ and $q_{\rm B}^h$. Putting them together, the complete set of terms added becomes

$$-\int_{(P_n)_{\rm SI}}\left(q_{\rm B}^h\mathbf{n}_{\rm B}-q_{\rm A}^h\mathbf{n}_{\rm A}\right)\cdot\frac{1}{2}\left(\mathbf{u}_{\rm B}^h-\mathbf{u}_{\rm A}^h\right)\mathrm{d}P$$

$$-\int_{(P_n)_{\rm SI}}\rho\mathbf{w}_{\rm B}^h\cdot\frac{1}{2}\left(\left(\mathcal{F}_{\rm B}^h-\left|\mathcal{F}_{\rm B}^h\right|\right)\mathbf{u}_{\rm B}^h-\left(\mathcal{F}_{\rm B}^h-\left|\mathcal{F}_{\rm B}^h\right|\right)\mathbf{u}_{\rm A}^h\right)\mathrm{d}P$$

$$-\int_{(P_n)_{\rm SI}}\rho\mathbf{w}_{\rm A}^h\cdot\frac{1}{2}\left(\left(\mathcal{F}_{\rm A}^h-\left|\mathcal{F}_{\rm A}^h\right|\right)\mathbf{u}_{\rm A}^h-\left(\mathcal{F}_{\rm A}^h-\left|\mathcal{F}_{\rm A}^h\right|\right)\mathbf{u}_{\rm B}^h\right)\mathrm{d}P$$

$$+\int_{(P_n)_{\mathrm{SI}}}\left(\mathbf{n}_{\mathrm{B}}\cdot\mathbf{w}_{\mathrm{B}}^{h}+\mathbf{n}_{\mathrm{A}}\cdot\mathbf{w}_{\mathrm{A}}^{h}\right)\frac{1}{2}\left(p_{\mathrm{B}}^{h}+p_{\mathrm{A}}^{h}\right)\mathrm{d}P$$

$$-\int_{(P_n)_{\mathrm{SI}}}\left(\mathbf{w}_{\mathrm{B}}^{h}-\mathbf{w}_{\mathrm{A}}^{h}\right)\cdot\left(\hat{\mathbf{n}}_{\mathrm{B}}\cdot\mu\left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^{h})+\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^{h})\right)\right)\mathrm{d}P$$

$$-\gamma_{\mathrm{ACI}}\int_{(P_n)_{\mathrm{SI}}}\hat{\mathbf{n}}_{\mathrm{B}}\cdot\mu\left(\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{B}}^{h}\right)+\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{A}}^{h}\right)\right)\cdot\left(\mathbf{u}_{\mathrm{B}}^{h}-\mathbf{u}_{\mathrm{A}}^{h}\right)\mathrm{d}P$$

$$+\int_{(P_n)_{\mathrm{SI}}}\frac{\mu C}{h}\left(\mathbf{w}_{\mathrm{B}}^{h}-\mathbf{w}_{\mathrm{A}}^{h}\right)\cdot\left(\mathbf{u}_{\mathrm{B}}^{h}-\mathbf{u}_{\mathrm{A}}^{h}\right)\mathrm{d}P, \tag{131}$$

where

$$\mathcal{F}_{\mathrm{B}}^{h}=\mathbf{n}_{\mathrm{B}}\cdot\left(\mathbf{u}_{\mathrm{B}}^{h}-\mathbf{v}_{\mathrm{B}}^{h}\right), \tag{132}$$

$$\mathcal{F}_{\mathrm{A}}^{h}=\mathbf{n}_{\mathrm{A}}\cdot\left(\mathbf{u}_{\mathrm{A}}^{h}-\mathbf{v}_{\mathrm{A}}^{h}\right), \tag{133}$$

$$h=\frac{h_{\mathrm{B}}+h_{\mathrm{A}}}{2}, \tag{134}$$

$$h_{\mathrm{B}}=2\left(\sum_{\alpha=1}^{n_{\mathrm{ent}}}\sum_{a=1}^{n_{\mathrm{ens}}}\left|\mathbf{n}_{\mathrm{B}}\cdot\boldsymbol{\nabla}N_a^{\alpha}\right|\right)^{-1}\quad\text{(for Side B)}, \tag{135}$$

$$h_{\mathrm{A}}=2\left(\sum_{\alpha=1}^{n_{\mathrm{ent}}}\sum_{a=1}^{n_{\mathrm{ens}}}\left|\mathbf{n}_{\mathrm{A}}\cdot\boldsymbol{\nabla}N_a^{\alpha}\right|\right)^{-1}\quad\text{(for Side A)}, \tag{136}$$

$$\hat{\mathbf{n}}_{\mathrm{B}}=\frac{\mathbf{n}_{\mathrm{B}}-\mathbf{n}_{\mathrm{A}}}{\|\mathbf{n}_{\mathrm{B}}-\mathbf{n}_{\mathrm{A}}\|}. \tag{137}$$

Here, $(P_n)_{\mathrm{SI}}$ is the SI in the ST domain, $\mathbf{v}$ is the mesh velocity, n_{ens} and n_{ent} are the number of spatial and temporal element nodes, N_a^{α} is the basis function associated with spatial and temporal nodes a and α, $\gamma_{\mathrm{ACI}}=1$, and C is a nondimensional constant. For our element length definition, we typically set $C=1$.

A number of remarks were provided in [9] to explain the added terms and to comment on related interpretations. We refer the reader interested in those details to [9].

Remark 7 A coefficient γ_{ACI} was added in [14] to the sixth integration so that we have the option of using $\gamma_{\mathrm{ACI}}=-1$. This option was added, in [96], also in the context of compressible flows. Using $\gamma_{\mathrm{ACI}}=1$ in a discontinuous Galerkin method was introduced in the symmetric interior penalty Galerkin method [134], and using $\gamma_{\mathrm{ACI}}=-1$ was introduced in the nonsymmetric interior penalty Galerkin method [135]. Stabilized methods based on both $\gamma_{\mathrm{ACI}}=1$ and -1 were reported in [31] in the context of the advection–diffusion equation. In the computations reported in this article, we set $\gamma_{\mathrm{ACI}}=1$.

Appendix 2: Derivative and Variation of the Normal Vector in the Shell Model

Derivative of the Normal Vector

Derivative of the normal vector with respect to ξ^α can be obtained as follows:

$$\overline{\mathbf{n}}_{,\alpha} = \frac{\partial}{\partial \xi^\alpha}\left(\frac{\overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2}{\left(\overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2\right)\cdot \mathbf{n}}\right) \tag{138}$$

$$= (\mathbf{I} - \mathbf{nn}) \cdot \frac{\overline{\mathbf{g}}_{1,\alpha} \times \overline{\mathbf{g}}_2 + \overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_{2,\alpha}}{\left(\overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2\right)\cdot \mathbf{n}} \tag{139}$$

$$= (\mathbf{I} - \mathbf{nn}) \cdot \frac{\overline{\mathbf{g}}_{1,\alpha} \times \left(\mathbf{n} \times \overline{\mathbf{g}}^1\right) + \overline{\mathbf{g}}_{2,\alpha} \times \left(\mathbf{n} \times \overline{\mathbf{g}}^2\right)}{\left(\overline{\mathbf{g}}^1 \times \overline{\mathbf{g}}^2\right)\cdot \mathbf{n}\left(\overline{\mathbf{g}}_1 \times \overline{\mathbf{g}}_2\right)\cdot \mathbf{n}} \tag{140}$$

$$= (\mathbf{I} - \mathbf{nn}) \cdot \left(\overline{\mathbf{g}}_{\beta,\alpha} \times \left(\mathbf{n} \times \overline{\mathbf{g}}^\beta\right)\right) \tag{141}$$

$$= (\mathbf{I} - \mathbf{nn}) \cdot \left(\left(\overline{\mathbf{g}}_{\beta,\alpha} \cdot \overline{\mathbf{g}}^\beta\right)\mathbf{n} - \left(\overline{\mathbf{g}}_{\beta,\alpha} \cdot \mathbf{n}\right)\overline{\mathbf{g}}^\beta\right) \tag{142}$$

$$= -\left(\overline{\mathbf{g}}_{\beta,\alpha} \cdot \mathbf{n}\right)\overline{\mathbf{g}}^\beta + \mathbf{n}\underbrace{\left(\mathbf{n} \cdot \overline{\mathbf{g}}^\beta\right)}_{=0}\left(\overline{\mathbf{g}}_{\beta,\alpha} \cdot \mathbf{n}\right) \tag{143}$$

$$= -\overline{\mathbf{g}}^\beta \overline{\mathbf{g}}_{\beta,\alpha} \cdot \mathbf{n} \tag{144}$$

$$= -\overline{\mathbf{g}}^\beta \overline{b}_{\alpha\beta}. \tag{145}$$

In the derivation, we used the following relationships, which generally hold:

$$\mathbf{g}_1 = \frac{\mathbf{g}^2 \times \mathbf{g}^3}{\left(\mathbf{g}^1 \times \mathbf{g}^2\right)\cdot \mathbf{g}^3}, \tag{146}$$

$$\mathbf{g}_2 = \frac{\mathbf{g}^3 \times \mathbf{g}^1}{\left(\mathbf{g}^1 \times \mathbf{g}^2\right)\cdot \mathbf{g}^3}, \tag{147}$$

$$\left(\mathbf{g}^1 \times \mathbf{g}^2\right)\cdot \mathbf{g}^3 = \left((\mathbf{g}_1 \times \mathbf{g}_2)\cdot \mathbf{g}_3\right)^{-1}. \tag{148}$$

Variation of the Normal Vector

From the steps given by Eqs. (138)–(144), the variation of the normal vector can be written as

$$\delta\mathbf{n} = -\overline{\mathbf{g}}^\beta \delta\overline{\mathbf{g}}_\beta \cdot \mathbf{n}. \tag{149}$$

Appendix 3: Variation of ξ^3 is a Second-Order Term

Taking the variation of both sides of Eq. (60), we obtain

$$\delta\xi^3 = \delta\overline{\lambda_3}\xi_0^3 + \delta\frac{\mathrm{d}\hat{\lambda_3}}{\mathrm{d}\xi_0^3}\left(\xi_0^3\right)^2 + \mathcal{O}\left(\left(\xi_0^3\right)^3\right). \tag{150}$$

For a representative value of the variation, we take the average over the thickness:

$$\overline{\delta\xi^3} \equiv \frac{1}{(h_{\mathrm{th}})_0}\int_{-(h_{\mathrm{th}})_0/2}^{(h_{\mathrm{th}})_0/2}\delta\xi^3\mathrm{d}\xi_0^3 \tag{151}$$

$$= \frac{1}{(h_{\mathrm{th}})_0}\left(\frac{2}{3}\delta\frac{\mathrm{d}\hat{\lambda_3}}{\mathrm{d}\xi_0^3}\left(\frac{(h_{\mathrm{th}})_0}{2}\right)^3 + \mathcal{O}\left((h_{\mathrm{th}})_0^5\right)\right) \tag{152}$$

$$= \frac{1}{12}\delta\frac{\mathrm{d}\hat{\lambda_3}}{\mathrm{d}\xi_0^3}(h_{\mathrm{th}})_0^2 + \mathcal{O}\left((h_{\mathrm{th}})_0^4\right). \tag{153}$$

Thus, the variation of ξ^3 is a second-order term.

Appendix 4: Variation of the Contravariant Basis Vector

Here we show that $\delta\mathbf{g}^\gamma$ can be expressed as

$$\delta\mathbf{g}^\gamma = -\mathbf{g}^\delta\mathbf{g}^\gamma \cdot \delta\mathbf{g}_\delta. \tag{154}$$

We start with the transformation from the contravariant basis vectors to the covariant basis vectors:

$$\mathbf{g}_\alpha = g_{\alpha\delta}\mathbf{g}^\delta. \tag{155}$$

We take the variation of both sides:

$$\delta\mathbf{g}_\alpha = \delta g_{\alpha\delta}\mathbf{g}^\delta + g_{\alpha\delta}\delta\mathbf{g}^\delta, \tag{156}$$

and from that obtain

$$g_{\alpha\delta}\delta\mathbf{g}^\delta = \delta\mathbf{g}_\alpha - \delta g_{\alpha\delta}\mathbf{g}^\delta. \tag{157}$$

From that and Eq. (13), we obtain

$$g_{\alpha\delta}\delta\mathbf{g}^{\delta} = \delta\mathbf{g}_{\alpha} - (\delta\mathbf{g}_{\alpha}\cdot\mathbf{g}_{\delta} + \mathbf{g}_{\alpha}\cdot\delta\mathbf{g}_{\delta})\,\mathbf{g}^{\delta} \tag{158}$$

$$= \delta\mathbf{g}_{\alpha} - \delta\mathbf{g}_{\alpha}\cdot\underbrace{\mathbf{g}_{\delta}\mathbf{g}^{\delta}}_{=\mathbf{I}} - \mathbf{g}_{\alpha}\cdot\delta\mathbf{g}_{\delta}\mathbf{g}^{\delta} \tag{159}$$

$$= \delta\mathbf{g}_{\alpha} - \delta\mathbf{g}_{\alpha} - \mathbf{g}_{\alpha}\cdot\delta\mathbf{g}_{\delta}\mathbf{g}^{\delta} \tag{160}$$

$$= -\mathbf{g}_{\alpha}\cdot\delta\mathbf{g}_{\delta}\mathbf{g}^{\delta}. \tag{161}$$

Multiplying both sides with $g^{\gamma\alpha}$, we obtain

$$\underbrace{g^{\gamma\alpha}g_{\alpha\delta}}_{=\delta^{\gamma}_{\delta}}\delta\mathbf{g}^{\delta} = -\underbrace{g^{\gamma\alpha}\mathbf{g}_{\alpha}}_{=\mathbf{g}^{\gamma}}\cdot\delta\mathbf{g}_{\delta}\mathbf{g}^{\delta}. \tag{162}$$

Thus,

$$\delta\mathbf{g}^{\gamma} = -\mathbf{g}^{\gamma}\cdot\delta\mathbf{g}_{\delta}\mathbf{g}^{\delta} \tag{163}$$

$$= -\mathbf{g}^{\delta}\mathbf{g}^{\gamma}\cdot\delta\mathbf{g}_{\delta}. \tag{164}$$

Appendix 5: Constitutive Law: Second Piola–Kirchhoff Tensor

$$(S_{\mathrm{NHI}})_{11} = \mu\left(1 - \frac{\lambda_3^2}{\lambda_1^2}\right), \tag{165}$$

$$(S_{\mathrm{NHR}})_{11} = \mu J^{-\frac{2}{3}}\left(1 - \frac{\lambda_1+\lambda_2+\lambda_3}{3\lambda_1^2}\right) + \frac{1}{2\lambda_1^2}\kappa\left(J^2-1\right), \tag{166}$$

$$(S_{\mathrm{FI}})_{11} = 2D_1D_2e^{(D_2((\lambda_1+\lambda_2+\lambda_3)-3))}\left(1 - \frac{\lambda_3^2}{\lambda_1^2}\right), \tag{167}$$

$$(S_{\mathrm{FR}})_{11} = 2D_1D_2e^{\left(D_2\left(J^{\frac{-2}{3}}(\lambda_1+\lambda_2+\lambda_3)-3\right)\right)}J^{-\frac{2}{3}}\left(1 - \frac{\lambda_1+\lambda_2+\lambda_3}{3\lambda_1^2}\right) + \frac{1}{2\lambda_1^2}\kappa\left(J^2-1\right). \tag{168}$$

Cylinder

$$(S_{\mathrm{NHI}})_{11} = \mu\left(1-\lambda_1^{-4}\right), \tag{169}$$

$$(S_{\mathrm{NHR}})_{11} = \mu\left(\lambda_1\lambda_3\right)^{-\frac{2}{3}}\left(1-\frac{\lambda_1+\lambda_3+1}{3\lambda_1^2}\right)+\frac{1}{2}\kappa\left(\lambda_3^2-\lambda_1^{-2}\right), \tag{170}$$

$$(S_{\mathrm{FI}})_{11} = 2D_1D_2e^{\left(D_2\left(\left(\lambda_1+\lambda_1^{-1}+1\right)-3\right)\right)}\left(1-\lambda_1^{-4}\right), \tag{171}$$

$$(S_{\mathrm{FR}})_{11} = 2D_1D_2e^{\left(D_2\left(J^{-\frac{2}{3}}(\lambda_1+\lambda_3+1)-3\right)\right)}\left(\lambda_1\lambda_3\right)^{-\frac{2}{3}}\left(1-\frac{\lambda_1+\lambda_3+1}{3\lambda_1^2}\right) + \frac{1}{2}\kappa\left(\lambda_3^2-\lambda_1^{-2}\right). \tag{172}$$

Sphere

$$(S_{\mathrm{NHI}})_{11} = \mu\left(1-\frac{1}{\lambda_1^6}\right), \tag{173}$$

$$(S_{\mathrm{NHR}})_{11} = \mu\left(\lambda_1^2\lambda_3\right)^{-\frac{2}{3}}\left(1-\frac{2\lambda_1+\lambda_3}{3\lambda_1^2}\right)+\frac{1}{2}\kappa\left(\lambda_3^2-\lambda_1^{-2}\right), \tag{174}$$

$$(S_{\mathrm{FI}})_{11} = 2D_1D_2e^{\left(D_2\left(\left(2\lambda_1+\lambda_1^{-2}\right)-3\right)\right)}\left(1-\frac{1}{\lambda_1^6}\right), \tag{175}$$

$$(S_{\mathrm{FR}})_{11} = 2D_1D_2e^{\left(D_2\left(\left(\lambda_1^2\lambda_3\right)^{-\frac{2}{3}}(2\lambda_1+\lambda_3)-3\right)\right)}\left(\lambda_1^2\lambda_3\right)^{-\frac{2}{3}}\left(1-\frac{2\lambda_1+\lambda_3}{3\lambda_1^2}\right) + \frac{1}{2}\kappa\left(\lambda_3^2-\lambda_1^{-2}\right). \tag{176}$$

References

1. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, and T. Ueda, "FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta", *Computational Mechanics*, **54** (2014) 1035–1045, https://doi.org/10.1007/s00466-014-1017-1.
2. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, T. Ueda, and T.E. Tezduyar, "A geometrical-characteristics study in patient-specific FSI analysis of blood flow in the thoracic aorta", in Y. Bazilevs and K. Takizawa, editors, *Advances in Computational Fluid–Structure*

Interaction and Flow Simulation: New Methods and Challenging Computations, Modeling and Simulation in Science, Engineering and Technology, 379–386, Springer, 2016, ISBN 978-3-319-40825-5.

3. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, "Space–time fluid mechanics computation of heart valve models", *Computational Mechanics*, **54** (2014) 973–986, https://doi.org/10.1007/s00466-014-1046-9.
4. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the Space–Time Slip Interface Topology Change (ST-SI-TC) method and Isogeometric Analysis (IGA)", in P. Wriggers and T. Lenarz, editors, *Biomedical Technology: Modeling, Experiments and Simulation*, Lecture Notes in Applied and Computational Mechanics, 77–99, Springer, 2018, ISBN 978-3-319-59547-4.
5. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods", *Computers & Fluids*, **158** (2017) 176–188, https://doi.org/10.1016/j.compfluid.2016.11.012.
6. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", *Computational Mechanics*, **48** (2011) 247–267, https://doi.org/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** (supp02) (2012) 1230001, https://doi.org/10.1142/S0218202512300013.
8. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, "Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, https://doi.org/10.1142/S0218202515400072.
9. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, "Space–time VMS method for flow computations with slip interfaces (ST-SI)", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, https://doi.org/10.1142/S0218202515400126.
10. K. Takizawa, T.E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi, "Computational thermo-fluid analysis of a disk brake", *Computational Mechanics*, **57** (2016) 965–977, https://doi.org/10.1007/s00466-016-1272-4.
11. 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, https://doi.org/10.1007/s00466-013-0935-7.
12. 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, https://doi.org/10.1115/1.4005073.
13. K. Takizawa, T.E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori, "Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA)", *Computers & Fluids*, **142** (2017) 15–20, https://doi.org/10.1016/j.compfluid.2016.02.021.
14. K. Takizawa, T.E. Tezduyar, S. Asada, and T. Kuraishi, "Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC)", *Computers & Fluids*, **141** (2016) 124–134, https://doi.org/10.1016/j.compfluid.2016.05.006.
15. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44, https://doi.org/10.1016/S0065-2156(08)70153-4.
16. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, https://doi.org/10.1002/fld.505.

17. 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, https://doi.org/10.1002/fld.1430.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.
24. 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.
25. 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, https://doi.org/10.1007/s11831-012-9071-3.
26. 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** (supp02) (2012) 1230002, https://doi.org/10.1142/S0218202512300025.
27. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
28. 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, https://doi.org/10.1142/S0218202513400010.
29. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, "New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2217–2226, https://doi.org/10.1142/S0218202515020029.
30. 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, https://doi.org/10.1016/S0045-7825(00)00204-8.
31. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", *Computers and Fluids*, **36** (2007) 12–26.
32. 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.
33. 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.
34. 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.

35. M.-C. Hsu and Y. Bazilevs, "Fluid–structure interaction modeling of wind turbines: simulating the full machine", *Computational Mechanics*, **50** (2012) 821–833.
36. M.E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, A.L. Marsden, and M. of Congenital Hearts Alliance (MOCHA), "A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations", *Computational Mechanics*, **48** (2011) 277–291, https://doi.org/10.1007/s00466-011-0599-0.
37. 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, https://doi.org/10.1002/fld.2400.
38. 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.
39. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "High-performance computing of wind turbine aerodynamics using isogeometric analysis", *Computers and Fluids*, **49** (2011) 93–100.
40. Y. Bazilevs, M.-C. Hsu, and M.A. Scott, "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines", *Computer Methods in Applied Mechanics and Engineering*, **249–252** (2012) 28–41.
41. 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.
42. 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.
43. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre, "Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods", *Archives of Computational Methods in Engineering*, **21** (2014) 359–398, https://doi.org/10.1007/s11831-014-9119-7.
44. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines", *International Journal for Numerical Methods in Engineering*, **102** (2015) 766–783, https://doi.org/10.1002/nme.4738.
45. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Aerodynamic simulation of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2013) 021011, https://doi.org/10.1115/1.4024415.
46. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel, and J.O. Dabiri, "FSI modeling of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2014) 081006, https://doi.org/10.1115/1.4027466.
47. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs, "Computational free-surface fluid–structure interaction with application to floating offshore wind turbines", *Computers and Fluids*, **141** (2016) 155–174, https://doi.org/10.1016/j.compfluid.2016.03.008.
48. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S.M.I. Gohari, and S. Sarkar, "ALE–VMS formulation for stratified turbulent incompressible flows with applications", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2349–2375, https://doi.org/10.1142/S0218202515400114.
49. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades", *Journal of Applied Mechanics*, **83** (6) (2016) 061010.
50. 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.
51. 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.

52. 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.
53. 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.
54. 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.
55. 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.
56. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Fluid–structure interaction simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **52** (2013) 971–981, https://doi.org/10.1007/s00466-013-0858-3.
57. C.C. Long, M. Esmaily-Moghadam, A.L. Marsden, and Y. Bazilevs, "Computation of residence time in the simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **54** (2014) 911–919, https://doi.org/10.1007/s00466-013-0931-y.
58. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk", *Computational Mechanics*, **54** (2014) 921–932, https://doi.org/10.1007/s00466-013-0967-z.
59. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation", *Computational Mechanics*, **54** (2014) 1055–1071, https://doi.org/10.1007/s00466-014-1059-4.
60. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks, "Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models", *Computational Mechanics*, **55** (2015) 1211–1225, https://doi.org/10.1007/s00466-015-1166-x.
61. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 1005–1053.
62. 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.
63. 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.
64. C. Wang, M.C.H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs, "Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis", *Computers and Fluids*, **142** (2017) 3–14, https://doi.org/10.1016/j.compfluid.2015.12.004.
65. M.C.H. Wu, D. Kamensky, C. Wang, A.J. Herrema, F. Xu, M.S. Pigazzini, A. Verma, A.L. Marsden, Y. Bazilevs, and M.-C. Hsu, "Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear", *Computer Methods in Applied Mechanics and Engineering*, (2017), Published online. https://doi.org/10.1016/j.cma.2016.09.032.
66. J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs, "Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines", *Computers and Fluids*, **158** (2017) 157–166, https://doi.org/10.1016/j.compfluid.2016.06.016.
67. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "Experimental and numerical FSI study of compliant hydrofoils", *Computational Mechanics*, **55** (2015) 1079–1090, https://doi.org/10.1007/s00466-014-1090-5.

68. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration", *Computers and Fluids*, **141** (2016) 201–211, https://doi.org/10.1016/j.compfluid.2015.07.013.
69. K. Takizawa and T.E. Tezduyar, "Computational methods for parachute fluid–structure interactions", *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, https://doi.org/10.1007/s11831-012-9070-4.
70. 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, https://doi.org/10.1007/s00466-012-0761-3.
71. 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, https://doi.org/10.1007/s00466-013-0880-5.
72. K. Takizawa, T.E. Tezduyar, C. Boswell, Y. Tsutsui, and K. Montel, "Special methods for aerodynamic-moment calculations from parachute FSI modeling", *Computational Mechanics*, **55** (2015) 1059–1069, https://doi.org/10.1007/s00466-014-1074-5.
73. 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, https://doi.org/10.1142/S0218202513400058.
74. K. Takizawa, T.E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel, "FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1203–1220, https://doi.org/10.1007/s00466-014-1052-y.
75. K. Takizawa, T.E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel, "Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1461–1476, https://doi.org/10.1007/s00466-014-1069-2.
76. K. Takizawa, T.E. Tezduyar, and R. Kolesar, "FSI modeling of the Orion spacecraft drogue parachutes", *Computational Mechanics*, **55** (2015) 1167–1179, https://doi.org/10.1007/s00466-014-1108-z.
77. 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, https://doi.org/10.1007/s00466-011-0589-2.
78. 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, https://doi.org/10.1007/s00466-011-0614-5.
79. 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, https://doi.org/10.1007/s00466-013-0888-x.
80. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, M.-C. Hsu, O. Øiseth, K.M. Mathisen, N. Kostov, and S. McIntyre, "Engineering analysis and design with ALE-VMS and space–time methods", *Archives of Computational Methods in Engineering*, **21** (2014) 481–508, https://doi.org/10.1007/s11831-014-9113-0.
81. K. Takizawa, "Computational engineering analysis with the new-generation space–time methods", *Computational Mechanics*, **54** (2014) 193–211, https://doi.org/10.1007/s00466-014-0999-z.
82. 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, https://doi.org/10.1007/s00466-012-0759-x.
83. 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, https://doi.org/10.1016/j.compfluid.2012.11.008.

84. 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, https://doi.org/10.1007/s00466-012-0758-y.
85. K. Takizawa, T.E. Tezduyar, and N. Kostov, "Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV", *Computational Mechanics*, **54** (2014) 213–233, https://doi.org/10.1007/s00466-014-0980-x.
86. K. Takizawa, T.E. Tezduyar, and A. Buscher, "Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping", *Computational Mechanics*, **55** (2015) 1131–1141, https://doi.org/10.1007/s00466-014-1095-0.
87. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt, "ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling", *Mathematical Models and Methods in Applied Sciences*, **24** (2014) 2437–2486, https://doi.org/10.1142/S0218202514500250.
88. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent", *Computational Mechanics*, **50** (2012) 675–686, https://doi.org/10.1007/s00466-012-0760-4.
89. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms", *Computational Mechanics*, **51** (2013) 1061–1073, https://doi.org/10.1007/s00466-012-0790-y.
90. K. Takizawa, D. Montes, S. McIntyre, and T.E. Tezduyar, "Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers", *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 223–248, https://doi.org/10.1142/s0218202513400022.
91. K. Takizawa, T.E. Tezduyar, and H. Hattori, "Computational analysis of flow-driven string dynamics in turbomachinery", *Computers & Fluids*, **142** (2017) 109–117, https://doi.org/10.1016/j.compfluid.2016.02.019.
92. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method", *Computers & Fluids*, **158** (2017) 189–200, https://doi.org/10.1016/j.compfluid.2017.04.017.
93. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "A general-purpose NURBS mesh generation method for complex geometries", to appear in a special volume to be published by Springer, 2018.
94. T. Kuraishi, K. Takizawa, and T.E. Tezduyar, "Space–time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation", to appear in a special volume to be published by Springer, 2018.
95. K. Takizawa, T.E. Tezduyar, and T. Terahara, "Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)", *Computers & Fluids*, **141** (2016) 191–200, https://doi.org/10.1016/j.compfluid.2016.05.027.
96. K. Takizawa, T.E. Tezduyar, and T. Kanai, "Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity", *Mathematical Models and Methods in Applied Sciences*, **27** (2017) 771–806, https://doi.org/10.1142/S0218202517500166.
97. 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, https://doi.org/10.1016/0045-7825(94)00082-4.
98. K. Takizawa and T.E. Tezduyar, "Space–time computation techniques with continuous representation in time (ST-C)", *Computational Mechanics*, **53** (2014) 91–99, https://doi.org/10.1007/s00466-013-0895-y.
99. J. Kiendl, K.U. Bletzinger, J. Linhard, and R. Wüchner, "Isogeometric shell analysis with Kirchhoff–Love elements", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3902–3914.

100. J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger, "The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 2403–2416.
101. 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.
102. J. Kiendl, M.-C. Hsu, M.C.H. Wu, and A. Reali, "Isogeometric Kirchhoff–Love shell formulations for general hyperelastic material", *Computer Methods in Applied Mechanics and Engineering*, **291** (2015) 280–303.
103. T.E. Tezduyar, S. Sathe, M. Schwaab, and B.S. Conklin, "Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique", *International Journal for Numerical Methods in Fluids*, **57** (2008) 601–629, https://doi.org/10.1002/fld.1633.
104. 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, https://doi.org/10.1016/j.cma.2008.05.024.
105. 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, https://doi.org/10.1002/cnm.1241.
106. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Multiscale sequentially-coupled arterial FSI technique", *Computational Mechanics*, **46** (2010) 17–29, https://doi.org/10.1007/s00466-009-0423-2.
107. 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, https://doi.org/10.1007/s00466-009-0425-0.
108. 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, https://doi.org/10.1002/fld.2360.
109. 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, https://doi.org/10.1002/cnm.1433.
110. 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, https://doi.org/10.1115/1.4005071.
111. C.D. Murray, "The physiological principle of minimum work: I. the vascular system and the cost of blood volume", *Proceedings of the National Academy of Sciences of the United States of America*, **12** (1926) 207–214.
112. T.E. Tezduyar and D.K. Ganjoo, "Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion problems", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 49–71, https://doi.org/10.1016/0045-7825(86)90023-X.
113. G.J. Le Beau, S.E. Ray, S.K. Aliabadi, and T.E. Tezduyar, "SUPG finite element computation of compressible flows with the entropy and conservation variables formulations", *Computer Methods in Applied Mechanics and Engineering*, **104** (1993) 397–422, https://doi.org/10.1016/0045-7825(93)90033-T.
114. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, https://doi.org/10.1016/j.compfluid.2005.02.011.
115. 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, https://doi.org/10.1016/j.cma.2005.05.032.

116. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", *Computers & Fluids*, **36** (2007) 147–159, https://doi.org/10.1016/j.compfluid.2005.07.009.
117. 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, https://doi.org/10.1007/s00466-005-0025-6.
118. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, https://doi.org/10.1007/s00466-006-0056-7.
119. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364, https://doi.org/10.1007/s00466-006-0045-x.
120. 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, https://doi.org/10.1016/j.compfluid.2005.07.004.
121. 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, https://doi.org/10.1002/fld.1743.
122. 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, https://doi.org/10.1002/fld.1447.
123. 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, https://doi.org/10.1002/fld.1484.
124. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms", *Journal of Applied Mechanics*, **76** (2009) 021211, https://doi.org/10.1115/1.3062967.
125. 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, https://doi.org/10.1115/1.3057496.
126. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167, https://doi.org/10.1007/s00466-009-0441-0.
127. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "Improving stability of stabilized and multiscale formulations in flow simulations at small time steps", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840, https://doi.org/10.1016/j.cma.2009.06.019.
128. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270, https://doi.org/10.1002/fld.2451.
129. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique", *Journal of Applied Mechanics*, **79** (2012) 010910, https://doi.org/10.1115/1.4005060.
130. A. Corsini, F. Rispoli, A.G. Sheard, and T.E. Tezduyar, "Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations", *Computational Mechanics*, **50** (2012) 695–705, https://doi.org/10.1007/s00466-012-0789-4.
131. P.A. Kler, L.D. Dalcin, R.R. Paz, and T.E. Tezduyar, "SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems", *Computational Mechanics*, **51** (2013) 171–185, https://doi.org/10.1007/s00466-012-0712-z.
132. A. Corsini, F. Rispoli, A.G. Sheard, K. Takizawa, T.E. Tezduyar, and P. Venturini, "A variational multiscale method for particle-cloud tracking in turbomachinery flows", *Computational Mechanics*, **54** (2014) 1191–1202, https://doi.org/10.1007/s00466-014-1050-0.

133. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T.E. Tezduyar, "Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing", *Computational Mechanics*, **55** (2015) 1201–1209, https://doi.org/10.1007/s00466-015-1160-3.
134. M.F. Wheeler, "An elliptic collocation-finite element method with interior penalties", *SIAM Journal on Numerical Analysis*, **15** (1978) 152–161.
135. P. Houston, C. Schwab, and E. Suli, "Discontinuous hp-finite element methods for advection-diffusion reaction problems", *SIAM Journal on Numerical Analysis*, **39** (2002) 2133–2163.

Residual-Based Large Eddy Simulation with Isogeometric Divergence-Conforming Discretizations

John A. Evans, Christopher Coley, Ryan M. Aronson, Corey L. Wetterer-Nelson, and Yuri Bazilevs

Abstract Isogeometric divergence-conforming discretizations have recently arisen as an attractive candidate for approximation of the incompressible Navier-Stokes problem. By construction, isogeometric divergence-conforming discretizations yield discrete velocity fields which are pointwise divergence-free, and as a consequence, they admit discrete balance laws for mass, momentum, angular momentum, energy, vorticity, enstrophy, and helicity. It has been demonstrated in previous work that isogeometric divergence-conforming discretizations are simultaneously more accurate and more stable than classical mixed methods when applied to the direct numerical simulation of incompressible fluid flow. In this chapter, we present two new residual-based large eddy simulation methodologies specifically designed for isogeometric divergence-conforming discretizations. The first methodology arises from a structure-preserving variational multiscale analysis of the incompressible Navier-Stokes equations. The second methodology combines ideas from variational multiscale analysis and large eddy simulation methodologies employing an eddy viscosity, yielding a residual-based eddy viscosity method. We develop quasi-static and dynamic models for both methodologies. Numerical results illustrate the new methodologies yield improved results as compared with standard eddy viscosity based approaches when applied to a transitional flow problem.

J. A. Evans (✉) · C. L. Wetterer-Nelson
University of Colorado Boulder, Boulder, CO, USA
e-mail: john.a.evans@colorado.edu; corey.nelson@colorado.edu

C. Coley
Laboratory for Physical Sciences, College Park, MD, USA
e-mail: coleyc@lps.umd.edu

R. M. Aronson
Stanford University, Stanford, CA, USA
e-mail: rmaronso@stanford.edu

Y. Bazilevs
Brown University, Providence, RI, USA
e-mail: yuri_bazilevs@brown.edu

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_3

1 Introduction

Isogeometric divergence-conforming discretizations have recently arisen as an attractive candidate for the spatial discretization of incompressible flow problem [10, 20–24, 38, 47]. These discretizations comprise a discrete Stokes complex [19, 25] and may be interpreted as smooth generalizations of Raviart-Thomas-Nédélec elements [41, 46]. When applied to incompressible flow problems, isogeometric divergence-conforming discretizations produce pointwise divergence-free velocity fields and hence exactly satisfy mass conservation. As a result, they preserve the balance law structure of the incompressible Navier-Stokes equations, and in particular, the properly conserve mass, linear momentum, angular momentum, energy, vorticity, enstrophy (in the two-dimensional setting), and helicity (in the three-dimensional setting) in the inviscid limit [23]. Moreover, isogeometric divergence-conforming discretizations are pressure-robust in that the velocity error is independent of the pressure field due to the pointwise satisfaction of the divergence-free constraint [48]. Isogeometric divergence-conforming discretizations have recently been applied to Cahn-Hilliard flow [54], turbulent flow [51], and fluid–structure interaction [39] where improved results were obtained in comparison with state-of-the-art mixed finite element discretization procedures. Moreover, the underlying structure of isogeometric divergence-conforming discretizations has been recently exploited to develop fast and robust geometric multigrid solvers [15] which converge faster than state-of-the-art multigrid solvers using Vanka [53], Braess-Sarazin [9], and Uzawa [27] smoothers.

In this chapter, we present two new residual-based large eddy simulation methodologies specifically designed for isogeometric divergence-conforming discretizations. Both of the methodologies are developed using variational multiscale analysis, a theoretical framework for incorporating missing fine-scale effects into numerical problems governing coarse-scale behavior [3, 31, 32, 34, 35]. The first methodology arises from a structure-preserving variational multiscale analysis of the incompressible Navier-Stokes equations and generalizes a recent methodology introduced by the authors [51], while the second methodology combines ideas from variational multiscale analysis and large eddy simulation methodologies employing an eddy viscosity, yielding a residual-based eddy viscosity method [44]. We develop both quasi-static and dynamic models for each of these methodologies. The new large eddy simulation methodologies preserve many of the key features of isogeometric divergence-conforming discretizations, most notably preservation of the pointwise divergence-free constraint for both the coarse-scales and the fine-scales. The new large eddy simulation methodologies also are residual-based, so the subgrid models associated with the methodologies automatically turn off when the flow is either laminar or fully resolved by the discretization. Finally, the new large eddy simulation methodologies are pressure robust as the discrete velocity field is completely decoupled from the pressure field approximation. Numerical results illustrate the new methodologies yield improved results as compared with standard eddy viscosity based approaches when applied to a transitional flow problem, namely the decay of the three-dimensional Taylor-Green vortex.

An outline of the remainder of the chapter is as follows. In Sect. 2, we present the problem of interest, namely the incompressible Navier-Stokes problem. In Sect. 3, we discuss the construction of isogeometric divergence-conforming discretizations for incompressible flow problems. In Sect. 4, we demonstrate how to construct a structure-preserving variational multiscale method for an isogeometric divergence-conforming discretization, and in Sect. 5, we present simplified quasi-static and dynamic subgrid models for the structure-preserving variational multiscale method. In Sect. 6, we discuss a methodology for discretizing the subgrid models presented in Sect. 5. In Sect. 7, we present the final form of the residual-based variational multiscale method which can be actually employed in a computer implementation, and we present the corresponding residual-based eddy viscosity method in Sect. 8. In Sect. 9, we apply our two new residual-based large eddy simulation methodologies to the simulation of Taylor-Green vortex flow, and finally, we present conclusions in Sect. 10.

2 The Incompressible Navier-Stokes Problem

In this section, we present the problem of interest in this chapter, namely three-dimensional, unsteady, incompressible fluid flow subject to no-penetration and free-slip boundary conditions. With this in mind, let $\Omega \subset \mathbb{R}^3$ denote an open, Lipschitz bounded domain, let $\partial\Omega$ denote the boundary of Ω, and let $T \in \mathbb{R}_+$. Given $\nu \in \mathbb{R}_+$, $\mathbf{f} : \Omega \times (0, T) \to \mathbb{R}^3$, and $\mathbf{u}_0 : \Omega \to \mathbb{R}^3$, the strong form of the Navier-Stokes problem reads as follows: Find $\mathbf{u} : \overline{\Omega} \times [0, T] \to \mathbb{R}^3$ and $p : \Omega \times (0, T) \to \mathbb{R}$ such that:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot (2\nu \nabla^s \mathbf{u}) + \nabla p &= \mathbf{f} \quad \text{for } (\mathbf{x}, t) \in \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{for } (\mathbf{x}, t) \in \Omega \times (0, T) \\ \mathbf{u} \cdot \mathbf{n} &= 0 \quad \text{for } (\mathbf{x}, t) \in \partial\Omega \times (0, T) \\ \frac{\partial}{\partial n} (\mathbf{u} \cdot \mathbf{t}_i) &= 0 \quad \text{for } (\mathbf{x}, t) \in \partial\Omega \times (0, T) \text{ and } i = 1, 2 \\ \mathbf{u}|_{t=0} &= \mathbf{u}_0 \quad \text{for } \mathbf{x} \in \Omega. \end{aligned} \tag{1}$$

Above, $\mathbf{u}$ denotes the velocity field, p denotes the pressure field, ν denotes the kinematic viscosity, $\mathbf{f}$ denotes the force per unit mass, $\mathbf{u}_0$ denotes the initial velocity field, $\mathbf{n}$ denotes the outward facing normal to $\partial\Omega$, $\mathbf{t}_i$ denotes the ith tangent vector to $\partial\Omega$, and $\nabla^s \mathbf{u}$ denotes the symmetrized gradient of the velocity field defined by:

$$\nabla^s \mathbf{u} = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right).$$

The velocity field is uniquely specified by the Navier-Stokes problem while the pressure field is unique up to a constant.

To state a weak formulation for the problem, we must first define suitable velocity and pressure test and trial spaces. In this direction, let us first define velocity and pressure test spaces:

$$\mathscr{V} := \mathbf{H}_n^1(\Omega) := \left\{ \mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega \right\}$$

$$\mathscr{Q} := L_0^2(\Omega) := \left\{ q \in L^2(\Omega) : \int_\Omega q d\Omega = 0 \right\},$$

and corresponding space-time velocity and pressure trial spaces:

$$\mathscr{V}_T := \left\{ \mathbf{v} \in C^0\left([0,T]; \mathscr{V}\right) : \partial_t \mathbf{v} \in L^2\left((0,T); \mathscr{V}^*\right) \right\}$$

$$\mathscr{Q}_T := L^2\left((0,T); \mathscr{Q}\right)$$

where, for a real Banach space X, $L^2\left((0,T); X\right)$ is the space of all strongly measurable functions $\phi : (0,T) \to X$ with:

$$\int_0^T \|\phi(t)\|_X^2 dt < \infty$$

and $C^0\left([0,T]; X\right)$ is the space of all continuous functions $\phi : (0,T) \to X$. Let us further assume that $\mathbf{f} \in L^2\left((0,T); \mathbf{L}^2(\Omega)\right)$. With the above definitions and assumptions in hand, the weak form of the Navier-Stokes problem is stated as follows: Find $\mathbf{u} \in \mathscr{V}_T$ and $p \in \mathscr{Q}_T$ such that $\mathbf{u}(0) = \mathbf{u}_0$ and, for almost every $t \in (0,T)$,

$$a(\mathbf{v}; \mathbf{u}) - b(\mathbf{v}, p) + b(\mathbf{u}, q) = \ell(\mathbf{v}) \tag{2}$$

for all $\mathbf{v} \in \mathscr{V}$ and $q \in \mathscr{Q}$ where:

$$a(\mathbf{v}; \mathbf{u}) := \int_\Omega \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial t}\, d\Omega - \int_\Omega \left(\nabla^s \mathbf{v}\right) : (\mathbf{u} \otimes \mathbf{u})\; d\Omega + \int_\Omega 2\nu \left(\nabla^s \mathbf{v}\right) : \left(\nabla^s \mathbf{u}\right)\; d\Omega$$

$$b(\mathbf{v}, p) := \int_\Omega (\nabla \cdot \mathbf{v})\, p\, d\Omega$$

$$\ell(\mathbf{v}) := \int_\Omega \mathbf{v} \cdot \mathbf{f}\, d\Omega.$$

With a weak form defined, we can turn to the matter of approximating the solution of the Navier-Stokes problem.

3 Approximation of Velocity and Pressure Fields

In this section, we present a particular selection of velocity and pressure approximation spaces which is not only inf-sup stable but also yields pointwise divergence-free discrete velocity fields when applied to the discretization of incompressible fluid flow. Before doing so, however, we first introduce the so-called Stokes complex,

which succinctly captures the fundamental theorem of calculus and expresses the differential relationships between potential, velocity, and pressure fields.

3.1 The Stokes Complex

The Stokes complex [19, 25] is a cochain complex of the form:

$$0 \longrightarrow \Phi \xrightarrow{\nabla} \boldsymbol{\Psi} \xrightarrow{\nabla\times} \mathscr{V} \xrightarrow{\nabla\cdot} \mathscr{Q} \longrightarrow 0 \tag{3}$$

in the three-dimensional setting where:

$$\begin{aligned}
\Phi &:= H_0^1(\Omega) \\
\boldsymbol{\Psi} &:= \left\{ \boldsymbol{\psi} \in \mathbf{L}^2(\Omega) : \nabla \times \boldsymbol{\psi} \in \mathbf{H}^1(\Omega) \text{ and } \boldsymbol{\psi} \times \mathbf{n} = \mathbf{0} \text{ on } \partial\Omega \right\} \\
\mathscr{V} &:= \mathbf{H}_n^1(\Omega) \\
\mathscr{Q} &:= L_0^2(\Omega)
\end{aligned}$$

are infinite-dimensional spaces of scalar potential fields, vector potential fields, velocity fields, and pressure fields. The Stokes complex is a smoothed version of the classical L^2 de Rham complex, and when the domain $\Omega \subset \mathbb{R}^3$ is simply connected with simply connected boundary, the Stokes complex is exact. This means that every pressure field may be represented as the divergence of a velocity field, every divergence-free velocity field may be represented as the curl of a vector potential field, and every curl-free vector potential field may be represented as the gradient of a scalar potential field.

It has been shown in previous works that the Stokes complex endows the incompressible Navier-Stokes problem with important underlying topological structure. In particular, the infinite-dimensional inf-sup condition may be derived from the complex [19]. As such, there is impetus for developing finite-dimensional approximations of the Stokes complex. Such discrete complexes are referred to as discrete Stokes complexes, and when these complexes are endowed with special commuting projection operators, they form the following commuting diagram with the Stokes complex:

$$\begin{array}{ccccccccccc}
0 & \longrightarrow & \Phi & \xrightarrow{\nabla} & \boldsymbol{\Psi} & \xrightarrow{\nabla\times} & \mathscr{V} & \xrightarrow{\nabla\cdot} & \mathscr{Q} & \longrightarrow & 0 \\
 & & \downarrow{\scriptstyle \Pi_\phi} & & \downarrow{\scriptstyle \Pi_\psi} & & \downarrow{\scriptstyle \Pi_v} & & \downarrow{\scriptstyle \Pi_q} & & \\
0 & \longrightarrow & \Phi_h & \xrightarrow{\nabla} & \boldsymbol{\Psi}_h & \xrightarrow{\nabla\times} & \mathscr{V}_h & \xrightarrow{\nabla\cdot} & \mathscr{Q}_h & \longrightarrow & 0
\end{array} \tag{4}$$

where Φ_h, $\boldsymbol{\Psi}_h$, $\mathscr{V}_h$, and $\mathscr{Q}_h$ are discrete scalar potential, vector potential, velocity, and pressure spaces and $\Pi_\phi : \Phi \to \Phi_h$, $\Pi_\psi : \boldsymbol{\Psi} \to \boldsymbol{\Psi}_h$, $\Pi_v : \mathscr{V} \to \mathscr{V}_h$, and $\Pi_q : \mathscr{Q} \to \mathscr{Q}_h$ are the aforementioned commuting projection operators. Remarkably,

when $\mathscr{V}_h$ and $\mathscr{Q}_h$ are selected as velocity and pressure approximation spaces in a mixed Galerkin formulation of the incompressible Navier-Stokes problem, the resulting approximation scheme is *inf-sup stable and free of spurious oscillations* and the *returned discrete velocity solution will be pointwise divergence-free* [21, 22]. Both of these properties are a direct consequence of the commuting diagram above, and for the sake of completeness, we prove the second property below.

Lemma 1 *Assume that the discrete velocity and pressure spaces $\mathscr{V}_h$ and $\mathscr{Q}_h$ are associated with a discrete complex which commutes with the Stokes complex. Suppose $\mathbf{v}_h \in \mathscr{V}_h$ satisfies $b(\mathbf{v}_h, q_h) = 0$ for every $q_h \in \mathscr{Q}_h$. Then $\nabla \cdot \mathbf{v}_h = 0$ pointwise.*

Proof Let $q_h = \nabla \cdot \mathbf{v}_h$. Then $\|\nabla \cdot \mathbf{v}_h\|^2_{L^2(\Omega)} = b(\mathbf{v}_h, q_h) = 0$ and the desired result follows.

While we have demonstrated the benefit of using velocity and pressure spaces coming from a discrete Stokes complex, we have not yet described how to arrive at such spaces. In order to do so, we turn to the use of so-called isogeometric compatible B-spline discretizations, which are the focus of the next two subsections.

3.2 Univariate and Multivariate B-splines

The basic building blocks of isogeometric compatible B-spline discretizations, like any isogeometric analysis technology, are B-splines [18]. B-splines are piecewise polynomial functions, but unlike C^0-continuous finite elements, B-splines may exhibit high levels of continuity. Univariate B-splines are constructed by first specifying a polynomial degree p, a number of basis functions n, and an open knot vector $\Xi = \{\xi_0, \xi_1, \ldots, \xi_{n+p+1}\}$, a non-decreasing vector of knots ξ_i such that the first and last knot are repeated $p + 1$ times. We assume without loss of generality that the first and last knot are 0 and 1, respectively, such that the domain of the knot vector is (0, 1). With a knot vector in hand, univariate B-spline basis functions are defined recursively through the Cox-deBoor formula:

$$\hat{N}_{i,p}(\xi) := \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \hat{N}_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \hat{N}_{i+1,p-1}(\xi) \quad \text{for } p > 0$$

$$\hat{N}_{i,0}(\xi) := \begin{cases} 1 \ \xi_i \leq \xi < \xi_{i+1} \\ 0 \text{ elsewhere.} \end{cases}$$

Figure 1 shows example sets of unvariate B-spline basis functions. We can alternatively define B-splines not from the knot vector itself, but instead a vector of unique knot values $\boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \ldots, \zeta_{n_k}\}$ and a regularity vector $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{n_k}\}$ such that the B-splines have α_j continuous derivatives across ζ_j. By construction, $\alpha_1 = \alpha_{n_k} = -1$. We will later employ the convention:

$$\boldsymbol{\alpha} - 1 = \{-1, \alpha_2 - 1, \ldots, \alpha_{n_k - 1} - 1, -1\}.$$

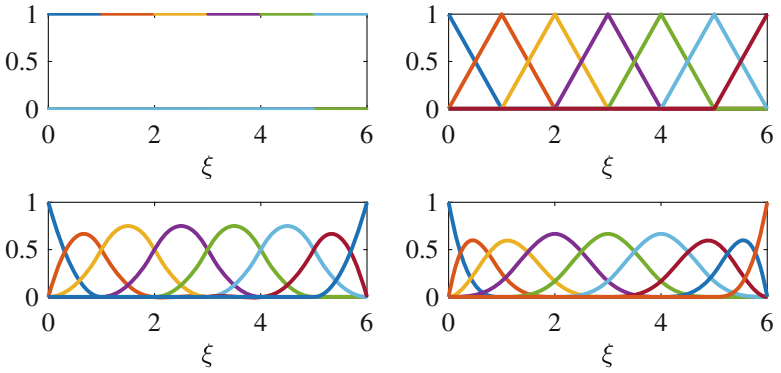


Fig. 1 Maximal continuity univariate B-spline basis functions of varying polynomial degree associated with a vector of unique knot values $\boldsymbol{\zeta} = \{0, 1, 2, 3, 4, 5, 6\}$: $p = 0$ (upper left), $p = 1$ (upper right), $p = 2$ (lower left), and $p = 3$ (lower right)

Given a set of knot vectors and polynomial degrees, multivariate B-spline basis functions are obtained through a tensor-product of univariate B-spline basis functions:

$$\hat{N}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) := \prod_{k=1}^{d} \hat{N}_{i_k,p_k}(\xi_k)$$

where $\mathbf{i} = (i_1, i_2, \ldots, i_d)$ and $\mathbf{p} = (p_1, p_2, \ldots, p_d)$. We denote the corresponding space of multidimensional B-splines over the parametric domain $\hat{\Omega} = (0, 1)^d$ as:

$$S^{p_1,p_2,\ldots,p_d}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_d}(\mathscr{M}_h) := \left\{ f : \hat{\Omega} \to \mathbb{R} \;\middle|\; f(\boldsymbol{\xi}) = \sum_{\mathbf{i}} a_{\mathbf{i}} \hat{N}_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) \right\},$$

where $\boldsymbol{\alpha}_j$ is the regularity vector associated with the jth direction where $j = 1, \ldots, d$ and $\mathscr{M}_h$ is the parametric mesh defined by the vectors of unique knot values in each parametric direction. Note that the space is fully characterized by the polynomial degrees, regularity vectors, and parametric mesh as indicated by the notation. For ease of notation, however, we drop the dependence on the parameteric mesh and instead use $S^{p_1,p_2,\ldots,p_d}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_d} = S^{p_1,p_2,\ldots,p_d}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_d}(\mathscr{M}_h)$ in what follows.

3.3 Isogeometric Compatible B-splines

We are now in a position to define isogeometric compatible B-splines. Their definition is made possible through the observation that the derivatives of univariate B-splines of degree p are univariate B-splines of degree $p - 1$. Since multivariate B-splines are tensor-products of univariate B-splines, the aforementioned property naturally generalizes to higher dimension, allowing us to build a discrete Stokes complex of B-spline spaces [11, 19]. We first define such a discrete Stokes complex

in the parametric domain $\hat{\Omega} = (0,1)^3$ before constructing a discrete Stokes complex in the physical domain of interest using a set of structure-preserving push-forward/pull-back operators.

Consider the following B-spline spaces over the unit cube:

$$\hat{\Phi}_h := \left\{ \hat{\phi}_h \in S^{p_1,p_2,p_3}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3} : \hat{\phi}_h = 0 \text{ on } \partial\hat{\Omega} \right\}$$

$$\hat{\boldsymbol{\Psi}}_h := \left\{ \hat{\boldsymbol{\psi}}_h \in S^{p_1-1,p_2,p_3}_{\boldsymbol{\alpha}_1-1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3} \times S^{p_1,p_2-1,p_3}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2-1,\boldsymbol{\alpha}_3} \times S^{p_1,p_2,p_3-1}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3-1} : \hat{\boldsymbol{\psi}}_h \times \mathbf{n} = \mathbf{0} \text{ on } \partial\hat{\Omega} \right\}$$

$$\hat{\mathscr{V}}_h := \left\{ \hat{\mathbf{v}}_h \in S^{p_1,p_2-1,p_3-1}_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2-1,\boldsymbol{\alpha}_3-1} \times S^{p_1-1,p_2,p_3-1}_{\boldsymbol{\alpha}_1-1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3-1} \times S^{p_1-1,p_2-1,p_3}_{\boldsymbol{\alpha}_1-1,\boldsymbol{\alpha}_2-1,\boldsymbol{\alpha}_3} : \hat{\mathbf{v}}_h \cdot \mathbf{n} = 0 \text{ on } \partial\hat{\Omega} \right\}$$

$$\hat{\mathscr{Q}}_h := \left\{ \hat{q}_h \in S^{p_1-1,p_2-1,p_3-1}_{\boldsymbol{\alpha}_1-1,\boldsymbol{\alpha}_2-1,\boldsymbol{\alpha}_3-1} : \int_{\hat{\Omega}} \hat{q}_h d\hat{\Omega} = 0 \right\},$$

where $\hat{\Phi}_h$ is the B-spline space of scalar potentials, $\hat{\boldsymbol{\Psi}}_h$ is the B-spline space of vector potentials, $\hat{\mathscr{V}}_h$ is the B-spline space of flow velocities, and $\hat{\mathscr{Q}}_h$ is the B-spline space of pressures. These discrete spaces are endowed with the basis functions $\{\hat{N}^{\phi}_i\}_{i=1}^{n_\phi}$, $\{\hat{\mathbf{N}}^{\psi}_i\}_{i=1}^{n_\psi}$, $\{\hat{\mathbf{N}}^{v}_i\}_{i=1}^{n_v}$, and $\{\hat{N}^{q}_i\}_{i=1}^{n_q}$, respectively, where n_ϕ is the number of scalar potential basis functions, n_ψ is the number of vector potential basis functions, n_v is the number of velocity basis functions, and n_q is the number of pressure basis functions, all of which can be inferred from the chosen polynomial degrees and knot vectors. It is easily shown that the above spaces form the following discrete Stokes complex:

$$0 \longrightarrow \hat{\Phi}_h \xrightarrow{\nabla} \hat{\boldsymbol{\Psi}}_h \xrightarrow{\nabla\times} \hat{\mathscr{V}}_h \xrightarrow{\nabla\cdot} \hat{\mathscr{Q}}_h \longrightarrow 0, \tag{5}$$

and provided the functions in the B-spline pressure space are at least C^0-continuous, there exists a set of commuting projection operators that make the above discrete complex commute with the Stokes complex [19].

To define compatible B-splines in the physical domain Ω, we need to first define a piece-wise smooth bijective mapping $\mathbf{F} : \hat{\Omega} \to \Omega$. This mapping can be defined using Non-Uniform Rational B-splines (NURBS), for instance, as is commonly done in the isogeometric analysis community [18], although this is certainly not the only way to define the mapping. With this mapping in hand, compatible B-spline spaces are defined in the physical domain via the relations:

$$\Phi_h := \left\{ \phi_h \in \Phi : \phi_h \circ \mathbf{F} \in \hat{\Phi}_h \right\}$$

$$\boldsymbol{\Psi}_h := \left\{ \boldsymbol{\psi}_h \in \boldsymbol{\Psi} : \mathbf{J}^{-T} \boldsymbol{\psi}_h \circ \mathbf{F} \in \hat{\boldsymbol{\Psi}}_h \right\}$$

$$\mathscr{V}_h := \left\{ \mathbf{v}_h \in \mathscr{V} : \det(\mathbf{J})\, \mathbf{J}^{-1} \mathbf{v}_h \circ \mathbf{F} \in \hat{\mathscr{V}}_h \right\}$$

$$\mathscr{Q}_h := \left\{ q_h \in \mathscr{Q} : \det(\mathbf{J})\, q_h \circ \mathbf{F} \in \hat{\mathscr{Q}}_h \right\},$$

where $\mathbf{J} = \partial_{\boldsymbol{\xi}} \mathbf{F}$ is the Jacobian of the parametric mapping. Corresponding basis functions in the physical domain are defined via push-forwards of the basis functions in the parametric domain, and we denote the discrete velocity basis functions as

$\{\mathbf{N}_i^v\}_{i=1}^{n_v}$ and the basis functions for other quantities in analogous fashion. It is easily shown that the compatible B-spline spaces in the physical domain also comprise a discrete complex which commutes with the Stokes complex. The compatible B-splines in the physical domain are referred to as isogeometric compatible B-splines, as they are built from B-splines, the basic building blocks of geometric modeling, and they are defined on the exact geometry of the problem of interest. A discretization which employs $\mathscr{V}_h$ as the discrete velocity space and $\mathscr{Q}_h$ as the discrete pressure space is referred to as an isogeometric divergence-conforming discretization since it automatically yields a divergence-free velocity field.

While we strictly consider three-dimensional compatible B-splines in this work, it is easiest to visualize two-dimensional compatible B-splines, where we have a discrete Stokes complex of the form:

$$0 \longrightarrow \Psi_h \xrightarrow{\nabla^\perp} \mathscr{V}_h \xrightarrow{\nabla\cdot} \mathscr{Q}_h \longrightarrow 0. \tag{6}$$

where Ψ_h is the B-spline space of streamfunctions, $\mathscr{V}_h$ is the B-spline space of flow velocities, and $\mathscr{Q}_h$ is the B-spline space of pressures. The proper construction of these B-spline spaces is discussed in [19]. The degrees of freedom associated with two-dimensional compatible B-splines are associated with the geometrical entries of the underlying control mesh. This is graphically illustrated in Fig. 2, which shows

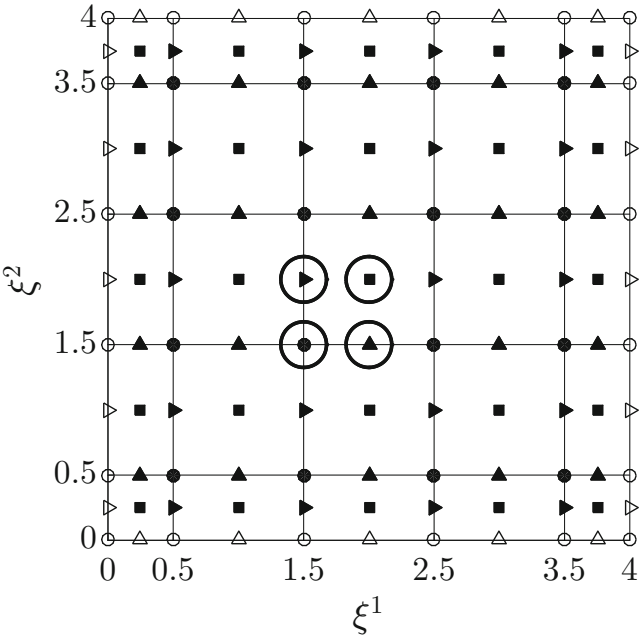


Fig. 2 Control mesh and degrees of freedom for maximal continuity compatible B-splines of degree $p_1 = p_2 = 2$ associated with vectors of unique knot values $\boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_2 = \{0, 1, 2, 3, 4\}$. Filled circles denote streamfunction degrees of freedom, filled triangles denote velocity degrees of freedom, and filled squares denote pressure degrees of freedom. Hollow markers indicate degrees of freedom associated with boundary conditions

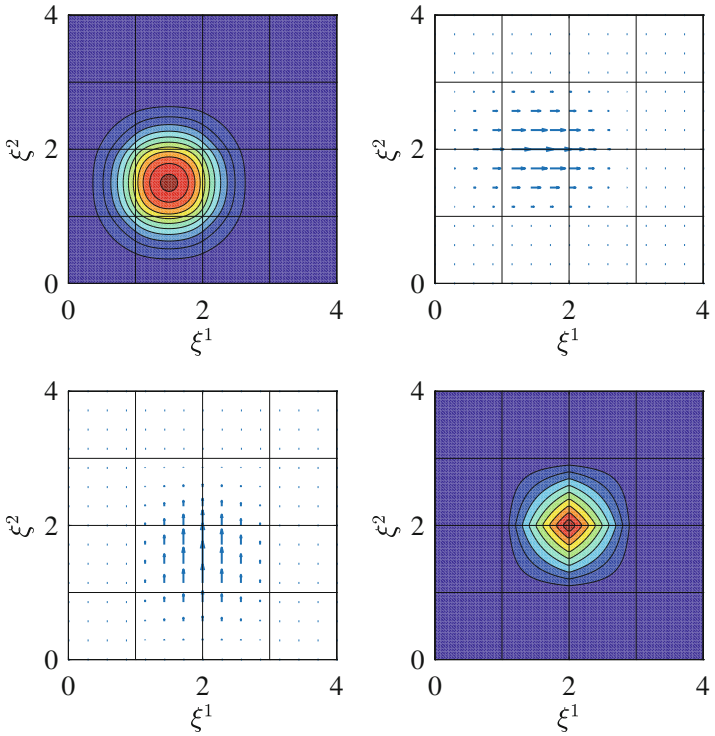


Fig. 3 Streamfunction (upper left), velocity (upper right and lower left), and pressure (lower right) basis functions associated with the circled degrees of freedom in Fig. 2

that streamfunction degrees of freedom are associated with control points, velocity degrees of freedom are associated with (and aligned normal to) control edges, and pressure degrees of freedom are associated with control cells. Each degree of freedom corresponds to a particular basis function, and to visualize these basis functions, we have selected four degrees of freedom in Fig. 2 and visualized the respective basis functions in Fig. 3. Note that the velocity basis functions are in fact vector-valued basis functions.

Analogous to the two-dimensional setting, the degrees of freedom associated with three-dimensional compatible B-splines are associated with the geometrical entries of the underlying control mesh. In particular, scalar potential degrees of freedom are associated with control points, vector potential degrees of freedom are associated (and aligned with) control edges, velocity degrees of freedom are associated with (and aligned normal to) control faces, and pressure degrees of freedom are associated with control cells.

4 The Structure-Preserving Variational Multiscale Method

In this section, we define a structure-preserving variational multiscale (VMS) method which allows us to capture the effect of unresolved fine-scale motion on the resolved velocity and pressure fields. The VMS method was originally introduced in [31] in order to identify the origins of stabilized finite element methods, and it was quickly realized that the VMS method provides a theoretical framework for incorporating missing fine-scale effects into numerical problems governing coarse-scale behavior [3, 31, 32, 34, 35]. Construction of the VMS method is simple: decompose the solution to a partial differential equation into a sum of coarse-scale and fine-scale components, determine the fine-scale component analytically in terms of the coarse-scale component, and solve for the coarse-scale component numerically. The above scale decomposition is uniquely specified by identifying a projector from the space of all scales onto the coarse-scale subspace. As a consequence, the coarse-scale component is a priori guaranteed to best-fit the solution in a variational sense.

To proceed forward, let us first define $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{Q}^h \subset \mathcal{Q}$ to be finite-dimensional compatible B-spline spaces of flow velocities and pressures as described in the previous section, and let us define corresponding continuous, linear projection operators $\mathscr{P}_v : \mathcal{V} \to \mathcal{V}_h$ and $\mathscr{P}_q : \mathcal{Q} \to \mathcal{Q}_h$. The projection operator $\mathscr{P}_v$ naturally splits the velocity space $\mathcal{V}$ into coarse-scale and fine-scale components as exhibited by the decomposition:

$$\mathcal{V} = \mathcal{V}^h \oplus \mathcal{V}'$$

where $\mathcal{V}' = \ker(\mathscr{P}_v)$ is the infinite-dimensional, fine-scale velocity space. Consequently, each function $\mathbf{v} \in \mathcal{V}$ is uniquely represented as the sum of a coarse-scale function $\mathbf{v}_h = \mathscr{P}_v \mathbf{v} \in \mathcal{V}^h$ and a fine-scale function $\mathbf{v}' = \mathbf{v} - \mathbf{v}_h \in \mathcal{V}'$. Similarly, the projection operator $\mathscr{P}_q$ naturally splits the pressure space $\mathcal{Q}$ into coarse-scale and fine-scale components:

$$\mathcal{Q} = \mathcal{Q}_h \oplus \mathcal{Q}'$$

where $\mathcal{Q}' = \ker(\mathscr{P}_q)$. Therefore, the velocity-pressure solution $(\mathbf{u}, p) \in \mathcal{V} \times \mathcal{Q}$ to the Navier-Stokes problem can be decomposed into coarse-scale components $(\mathbf{u}_h, p_h) \in \mathcal{V}_h \times \mathcal{Q}_h$ and fine-scale components $\left(\mathbf{u}', p'\right) \in \mathcal{V}' \times \mathcal{Q}'$.

Heretofore, we have discussed how to split the solution to the problem into coarse-scale and fine-scale components, but we have not discussed how to obtain said components via a numerical method. To do so, we simply use the decompositions $\mathcal{V} = \mathcal{V}^h \oplus \mathcal{V}'$ and $\mathcal{Q} = \mathcal{Q}^h \oplus \mathcal{Q}'$ and bilinearity to perform a scale splitting of the problem. First, however, we must define space-time trial spaces for coarse-scale and fine-scale velocity and pressure fields as follows:

$$\mathcal{V}_T^h := \left\{ \mathbf{v} \in C^0\left([0,T];\mathcal{V}_h\right) : \partial_t \mathbf{v} \in L^2\left((0,T);(\mathcal{V}_h)^*\right) \right\}$$

$$\mathcal{Q}_T^h := L^2\left((0,T);\mathcal{Q}_h\right)$$

$$\mathscr{V}'_T := \left\{ \mathbf{v} \in C^0\left([0,T];\mathscr{V}'\right) : \partial_t \mathbf{v} \in L^2\left((0,T);\left(\mathscr{V}'\right)^*\right) \right\}$$
$$\mathscr{Q}'_T := L^2\left((0,T);\mathscr{Q}'\right).$$

The VMS method then takes the form: Find $\mathbf{u}_h \in \mathscr{V}^h_T$, $\mathbf{u}' \in \mathscr{V}'_T$, $p_h \in \mathscr{Q}^h_T$, and $p' \in \mathscr{Q}'_T$ such that suitable initial conditions are satisfied,

$$a(\mathbf{v}_h; \mathbf{u}_h + \mathbf{u}') - b(\mathbf{v}_h, p_h + p') + b(\mathbf{u}_h + \mathbf{u}', q_h) = \ell(\mathbf{v}_h) \tag{7}$$

for all $\mathbf{v}_h \in \mathscr{V}_h$, $q_h \in \mathscr{Q}_h$, and almost every $t \in (0,T)$ and

$$a(\mathbf{v}'; \mathbf{u}_h + \mathbf{u}') - b(\mathbf{v}', p_h + p') + b(\mathbf{u}_h + \mathbf{u}', q') = \ell(\mathbf{v}') \tag{8}$$

for all $\mathbf{v}' \in \mathscr{V}'$, $q' \in \mathscr{Q}'$, and almost every $t \in (0,T)$. The coarse-scale problem is given by (7), whereas the fine-scale problem is given by (8). This is the split variational problem that relates the effect of the fine scales on the coarse scales and vice versa. It should be noted that at this point, the VMS method is *exact*, and the solution of the problem satisfies $\mathbf{u}_h = \mathscr{P}_v \mathbf{u}$ and $p_h = \mathscr{P}_q p$. However, the fine-scale problem is an infinite-dimensional problem. Since this is, in general, intractable to solve, some form of modeling must be employed. We will later discuss several strategies in this direction.

At this juncture, the VMS method as presented is not structure-preserving in that it is not guaranteed to yield a coarse-scale velocity field which is pointwise divergence-free. To arrive at such a structure-preserving VMS method, we must make certain assumptions regarding the form of the projection operators $\mathscr{P}_v$ and $\mathscr{P}_q$. Namely, we must assume that they along with the spaces $\mathscr{V}$, $\mathscr{Q}$, $\mathscr{V}_h$, and $\mathscr{Q}_h$ form the commuting diagram:

$$\begin{array}{ccc} \mathscr{V} & \xrightarrow{\nabla\cdot} & \mathscr{Q} \\ \downarrow{\scriptstyle \mathscr{P}_v} & & \downarrow{\scriptstyle \mathscr{P}_q} \\ \mathscr{V}_h & \xrightarrow{\nabla\cdot} & \mathscr{Q}_h. \end{array} \tag{9}$$

It then follows that $\nabla \cdot \mathbf{u}_h = \nabla \cdot \mathscr{P}_v \mathbf{u} = \mathscr{P}_q \nabla \cdot \mathbf{u} = 0$, and since $\mathbf{u} = \mathbf{u}_h + \mathbf{u}'$, it also follows that $\nabla \cdot \mathbf{u}' = 0$. If $\mathscr{P}_q$ is further assumed to be an L^2-projection operator, then we find that:

$$b(\mathbf{v}_h, p') = \int_\Omega (\nabla \cdot \mathbf{v}_h)\, p' d\Omega = 0$$

for all $\mathbf{v}_h \in \mathscr{V}_h$ since $\nabla \cdot \mathbf{v}_h \in \mathscr{Q}_h$ for all $\mathbf{v}_h \in \mathscr{V}_h$. Consequently, the fine-scale pressure field has no influence on the coarse-scale velocity and pressure fields for a structure-preserving VMS method.

We can analytically solve for the fine-scale velocity and pressure fields in terms of the *residuals* of the resolved scales [31, 32]. More specifically, we can write $\mathbf{u}' \in \mathscr{V}'$ and $p' \in \mathscr{Q}'$ in terms of the momentum and continuity residuals as follows:

$$\mathbf{u}' = \mathscr{G}'_{\text{vel}} \left(\text{Res}_m \left(\mathbf{u}_h, p_h \right), \text{Res}_c \left(\mathbf{u}_h, p_h \right) \right) \tag{10}$$

$$p' = \mathscr{G}'_{\text{press}} \left(\text{Res}_m \left(\mathbf{u}_h, p_h, \text{Res}_c \left(\mathbf{u}_h, p_h \right) \right) \right) \tag{11}$$

where $\mathscr{G}'_{\text{vel}}$ and $\mathscr{G}'_{\text{press}}$ represent fine-scale Green's operators [17, 31, 32] and:

$$\text{Res}_m \left(\mathbf{u}_h, p_h \right) := \mathbf{f} - \partial_t \mathbf{u}_h - \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot \left(2\nu \nabla^s \mathbf{u}_h \right) - \nabla p_h \tag{12}$$

and:

$$\text{Res}_c \left(\mathbf{u}_h, p_h \right) := -\nabla \cdot \mathbf{u}_h . \tag{13}$$

With a structure-preserving VMS method, $\text{Res}_c \left(\mathbf{u}_h, p_h \right) \equiv 0$, so we have the simpler expressions:

$$\mathbf{u}' = \mathscr{G}'_{\text{vel}} \left(\text{Res}_m \left(\mathbf{u}_h, p_h \right) \right) \tag{14}$$

$$p' = \mathscr{G}'_{\text{press}} \left(\text{Res}_m \left(\mathbf{u}_h, p_h \right) \right) \tag{15}$$

where we have employed the abuse of notation $\mathscr{G}'_{\text{vel}} \left(\mathbf{r} \right) = \mathscr{G}'_{\text{vel}} \left(\mathbf{r}, 0 \right)$ and $\mathscr{G}'_{\text{press}} \left(\mathbf{r} \right) = \mathscr{G}'_{\text{vel}} \left(\mathbf{r}, 0 \right)$. In order to evaluate (14) and (15), we need to employ suitable approximations for the fine-scale Green's operators $\mathscr{G}'_{\text{vel}}$ and $\mathscr{G}'_{\text{press}}$. This is addressed in the next section. By inserting the above expressions for $\mathbf{u}'$ and p' back into the coarse-scale problem, a final finite-dimensional system for the coarse-scale solution $\left(\mathbf{u}_h, p_h \right) \in \mathscr{V}_h \times \mathscr{Q}_h$ is attained.

The primary issue associated with the VMS method is that $\mathscr{V}'$ and $\mathscr{Q}'$ are infinite-dimensional spaces and thus solving the fine-scale problem is an intractable task. Fortunately, for most problems of interest, it is sufficient to approximate the effect of the fine-scales on the coarse-scale solution in order to produce stable and accurate numerical solutions [17]. This motivates the need to pursue effective approaches for modeling the fine-scale problem.

5 Quasi-Static and Dynamic Fine-Scale Models

In this section, we present two methodologies for approximating the fine-scale problem appearing in the structure-preserving VMS method. To begin, we make a rather strong assumption regarding the form of the fine-scale velocity field. Namely, we assume that the fine-scale velocity field is zero along element boundaries. This assumption is somewhat justified by the fact that the fine-scale Green's function has localized support [35], and it can be interpreted physically as assuming that the fine-scale velocity field cannot transport mass between adjacent elements. Practically, the above assumption implies that we can localize the fine-scale problem to a single element. The resulting element-wise fine-scale problem takes the form:

$$
\begin{aligned}
&\partial_t \mathbf{u}' + \mathbf{u}_h \cdot \nabla \mathbf{u}' + \mathbf{u}' \cdot \nabla \mathbf{u}_h + \mathbf{u}' \cdot \nabla \mathbf{u}' \\
&\qquad -\nabla \cdot \left(2\nu \nabla^s \mathbf{u}'\right) + \nabla p' = \mathrm{Res}_m\left(\mathbf{u}_h, p_h\right) \quad \text{for } (\mathbf{x}, t) \in \Omega_e \times (0, T) \\
&\qquad \nabla \cdot \mathbf{u}' = 0 \quad \text{for } (\mathbf{x}, t) \in \Omega_e \times (0, T) \\
&\qquad \mathbf{u}' = \mathbf{0} \quad \text{for } (\mathbf{x}, t) \in \partial\Omega_e \times (0, T) \\
&\qquad \mathbf{u}'\big|_{t=0} = \mathbf{u}'_0 \quad \text{for } \mathbf{x} \in \Omega_e
\end{aligned}
\tag{16}
$$

where Ω_e is a single element in the domain. For an isogeometric divergence-conforming discretization, an element is a Bézier element [7]. The fine-scale pressure field is only uniquely specified up to a constant by the above problem, so we enforce $\int_{\Omega_e} p' d\Omega_e = 0$.

Unfortunately, the fine-scale problem given by (16) is nearly as difficult to solve as the original Navier-Stokes problem. Fortunately, we are not interested in the fine-scale solution field itself but rather its influence on the coarse-scale solution field. This indicates that we can solve an approximate *model* problem for the fine-scale solution field. We consider two candidate approximations to (16). In the first approximation, denoted the *quasi-static* approximation, the fine-scale solution field is assumed to be in local equilibrium with the coarse-scale momentum residual. This yields the model:

$$
\text{Quasi-static:} \quad \partial_t \mathbf{u}' + \mathbf{u}_h \cdot \nabla \mathbf{u}' + \mathbf{u}' \cdot \nabla \mathbf{u}_h + \mathbf{u}' \cdot \nabla \mathbf{u}' - \nabla \cdot \left(2\nu \nabla^s \mathbf{u}'\right) \approx \tau_{qs}^{-1} \mathbf{u}' \tag{17}
$$

where τ_{qs} is a suitable time-scale for the fine-scale velocity field unsteadiness and fine-scale advection and diffusion processes. In the second approximation, denoted the *dynamic* approximation, the fine-scale solution field is not assumed to be in local equilibrium with the coarse-scale momentum residual, but it is assumed that fine-scale advection and diffusion processes can be approximated by the model:

$$
\text{Dynamic:} \quad \mathbf{u}_h \cdot \nabla \mathbf{u}' + \mathbf{u}' \cdot \nabla \mathbf{u}_h + \mathbf{u}' \cdot \nabla \mathbf{u}' - \nabla \cdot \left(2\nu \nabla^s \mathbf{u}'\right) \approx \tau_d^{-1} \mathbf{u}' \tag{18}
$$

where τ_d is a suitable time-scale for the fine-scale advection and diffusion processes. The quasi-static model is closely related to the original residual-based VMS method proposed in [3], while the dynamic model is closely related to the dynamic subscales method of Codina proposed in [13]. It remains to specify τ_{qs} and τ_d. To do so, we first employ dimensional analysis to identify appropriate advective and diffusive time scales for the fine-scales from the mesh size h, the viscosity ν, and the coarse-scale velocity field $\mathbf{u}_h$, yielding:

$$
\begin{aligned}
\text{Advective:} \quad \tau_{\text{adv}} &= \frac{h}{C_I \left|\mathbf{u}_h\right|} \\
\text{Diffusive:} \quad \tau_{\text{diff}} &= \frac{h^2}{C_{II} \nu}
\end{aligned}
$$

where C_I and C_{II} are model constants. Next, we identify an additional time-scale related to the unsteadiness of the fine-scale velocity and pressure fields. One option which has its origins in stabilized methods (see, e.g., [33, 49]) is to relate the time-scale to the time step size used in the numerical discretization:

$$\text{Unsteady:} \quad \tau_{\text{uns}} = \frac{\Delta t}{C_{III}}, \tag{19}$$

where again C_{III} is a model constant. In the small time-step limit, however, this yields an infinitely small time-scale which is certainly not proportional to the fine-scale unsteadiness. Worse yet, this can yield an ineffective or even unstable method. One solution to this problem is to employ element-vector-based stabilization parameters, as suggested originally in [50] and later in [30]. Alternatively, a time-scale can be constructed directly from the time-derivative of the coarse-scale velocity field:

$$\text{Unsteady:} \quad \tau_{\text{uns}} = \frac{|\mathbf{u}_h|}{C_{III}\,|\partial_t \mathbf{u}_h|}. \tag{20}$$

We will consider unsteady time-scales defined by both (19) and (20) later in this chapter. Having defined appropriate time scales related to the advection, diffusion, and unsteadiness of the fine-scale flow field, we can now propose models for τ_{qs} and τ_d:

$$\text{Quasi-static:} \quad \tau_{qs} = \left(\tau_{\text{uns}}^{-2} + \tau_{\text{adv}}^{-2} + \tau_{\text{diff}}^{-2}\right)^{-1/2} \tag{21}$$

$$\text{Dynamic:} \quad \tau_{d} = \left(\tau_{\text{adv}}^{-2} + \tau_{\text{diff}}^{-2}\right)^{-1/2}. \tag{22}$$

Utilizing (21) and (22) in conjunction with (17) and (18), respectively, then defines our approximation to the fine-scale problem. Note that the resulting approximate fine-scale problem is still infinite-dimensional, so we will need to discretize it in some form or fashion. We will turn to the so-called method of subgrid vortices in the next section.

It remains to specify the model constants C_I, C_{II}, and C_{III}. Classically, stability analysis has been employed to determine appropriate values for such model constants [36]. Stability analysis in the context of a structure-preserving VMS method is quite challenging and will not be presented here. Nonetheless, 2 is a suitable value for both C_I and C_{III}, and a suitable value for C_{II} can be determined using inverse inequalities [29]. It can be shown that a fine-scale model using these values yields an energy-stable variational formulation [14].

6 Approximating the Fine-Scales with Subgrid Vortices

In the previous section, we developed two models, the quasi-static and dynamic models, for the fine-scale velocity and pressure fields. However, these two models are infinite-dimensional. In this section, we introduce a methodology for discretizing these models. Without loss of generality, let us consider the quasi-static model which takes the form of a Darcy problem with diffusivity τ_{qs}^{-1} driven by the momentum residual of the coarse-scale solution:

$$\begin{aligned} \tau_{qs}^{-1}\mathbf{u}' + \nabla p' &= \mathrm{Res}_m\,(\mathbf{u}_h,\, p_h) \qquad \text{for } (\mathbf{x}, t) \in \Omega_e \times (0, T) \\ \nabla \cdot \mathbf{u}' &= 0 \qquad\qquad\qquad\quad \text{for } (\mathbf{x}, t) \in \Omega_e \times (0, T). \end{aligned} \tag{23}$$

The above problem is subject to the conditions:

$$\begin{aligned} \mathbf{u}' \cdot \mathbf{n} &= 0 \quad \text{for } (\mathbf{x}, t) \in \partial\Omega_e \times (0, T) \\ \int_{\Omega_e} p' d\Omega_e &= 0 \quad \text{for } (\mathbf{x}, t) \in \Omega_e \times (0, T) \end{aligned} \tag{24}$$

where $\mathbf{n}$ is the outward-facing normal vector on the element boundary. Note that while no-slip and no-penetration boundary conditions were applied in the original fine-scale model proposed in the previous section, only no-penetration boundary conditions are applied for the quasi-static model. This is due to the fact that the Darcy problem does not have enough regularity for the enforcement of no-slip boundary conditions.

To proceed forward, we write the quasi-static model in weak form. Defining the function spaces:

$$\begin{aligned} \mathscr{V}_e' &:= \left\{\mathbf{v}' \in \mathbf{H}(\mathrm{div},\, \Omega_e) : \mathbf{v}' \cdot \mathbf{n} = 0 \text{ on } \partial\Omega_e\right\} \\ \mathscr{Q}_e' &:= \left\{q' \in L^2(\Omega_e) : \int_{\Omega_e} q' d\Omega_e = 0\right\}, \end{aligned}$$

the weak form of the quasi-static model becomes: Find $\mathbf{u}' : (0, T) \to \mathscr{V}_e'$ and $p' : (0, T) \to \mathscr{Q}_e'$ such that appropriate initial conditions are satisfied and:

$$a_e(\mathbf{v}', \mathbf{u}') - b_e(\mathbf{v}', p') + b_e(\mathbf{u}', q') = \ell_e(\mathbf{v}') \tag{25}$$

for all $\mathbf{v}' \in \mathscr{V}_e'$, $p' \in \mathscr{Q}_e'$, and almost every $t \in (0, T)$ where:

$$a_e(\mathbf{v}'; \mathbf{u}') := \int_{\Omega_e} \tau_{qs}^{-1}\mathbf{v}' \cdot \mathbf{u}'\, d\Omega_e$$

$$b_e(\mathbf{v}', p') := \int_{\Omega_e} \left(\nabla \cdot \mathbf{v}'\right) p'\, d\Omega_e$$

$$\ell_e(\mathbf{v}') := \int_{\Omega_e} \mathbf{v}' \cdot \mathrm{Res}_m\,(\mathbf{u}_h,\, p_h)\; d\Omega_e.$$

Now, recall that the fine-scale pressure field does not affect the coarse-scales for the structure-preserving VMS method. Consequently, we only need to solve for the fine-scale velocity field in the above problem. With this in mind, we can restrict to the divergence-free subspace of $\mathscr{V}_e'$:

$$\mathring{\mathscr{V}}_e' := \left\{ \mathbf{v}' \in \mathscr{V}_e' : \nabla \cdot \mathbf{v}' \equiv 0 \right\},$$

yielding the reduced problem: Find $\mathbf{u}' : (0, T) \to \mathring{\mathscr{V}}_e'$ such that appropriate initial conditions are satisfied and:

$$a_e(\mathbf{v}', \mathbf{u}') = \ell_e(\mathbf{v}') \tag{26}$$

for all $\mathbf{v}' \in \mathring{\mathscr{V}}_e'$ and almost every $t \in (0, T)$. While we have reduced the complexity of the fine-scale problem, it is still infinite-dimensional. Thus, the premise of the method of subgrid vortices is to replace the divergence-free subspace $\mathring{\mathscr{V}}_e'$ with a finite-dimensional approximation space $\mathring{\mathscr{V}}_{h,e}' \subset \mathring{\mathscr{V}}_e'$.

To arrive at a suitable finite-dimensional approximation space $\mathring{\mathscr{V}}_{h,e}' \subset \mathring{\mathscr{V}}_e'$, we recall that the de Rham complex is exact when the underlying domain is simply connected with simply connected boundary, which is in fact the case when the domain consists of a single element. As such, any function $\mathbf{v}' \in \mathring{\mathscr{V}}_e'$ can be written as the curl of a function in $\mathbf{H}(\mathbf{curl}, \Omega_e)$. Thus, we elect to construct $\mathring{\mathscr{V}}_{h,e}'$ such that any function $\mathbf{v}_h' \in \mathring{\mathscr{V}}_{h,e}'$ can be written as the curl of a curl-conforming Nédélec finite element [41]. To be more precise, let:

$$\hat{\mathscr{N}}_{p_f} := \mathrm{Q}_{p_f-1,p_f,p_f} \times \mathrm{Q}_{p_f,p_f-1,p_f} \times \mathrm{Q}_{p_f,p_f,p_f-1} \tag{27}$$

be the space of Nédélec finite elements of degree p_f where $\mathrm{Q}_{r,s,t}$ is the space of tensor-product polynomials of degree (r, s, t) over the unit cube. Furthermore, let $\mathbf{F}_e : \hat{\Omega} \to \Omega_e$ be the mapping between the unit cube and the element in physical space, and let $\mathbf{J}_e = \partial_{\boldsymbol{\xi}} \mathbf{F}_e$ be the Jacobian of the mapping. Then, we select:

$$\mathring{\mathscr{V}}_{h,e}' := \left\{ \mathbf{v}_h' \in \mathring{\mathscr{V}}_e' : \mathbf{v}_h' = \mathbf{curl} \boldsymbol{\psi}_h' \text{ with } \boldsymbol{\psi}_h' \in \mathscr{N}_{p_f}(\Omega_e) \right\} \tag{28}$$

where:

$$\mathscr{N}_{p_f}(\Omega_e) := \left\{ \boldsymbol{\psi}_h' \in \mathbf{H}(\mathbf{curl}, \Omega_e) : \mathbf{J}_e^{-T} \boldsymbol{\psi}_h \circ \mathbf{F}_e \in \hat{\mathscr{N}}_{p_f} \right\} \tag{29}$$

With this selection, a basis for $\mathring{\mathscr{V}}_{h,e}'$ can be constructed by applying the curl operator directly to a basis for $\mathscr{N}_{p_f}(\Omega_e)$. When basis functions for $\mathscr{N}_{p_f}(\Omega_e)$ are constructed using Bernstein polynomials (which coincides with the isogeometric compatible B-spline basis when there are no interior knots), their curls appear as single vortical filaments or *subgrid vortices*, hence giving rise to the name of the method. The subgrid vortices corresponding to polynomial degree $p_f = 2$ and the unit cube are illustrated in Fig. 4. Vorticity isosurfaces and streamlines are displayed to show the structure of the subgrid vortices.

It should be noted that the application of the curl operator to a basis of $\mathscr{N}_{p_f}(\Omega_e)$ does not immediately yield a basis. This is because the resulting set of functions will contain $\left(p_f - 1\right)^d$ linearly dependent subgrid vortices. There are many methods

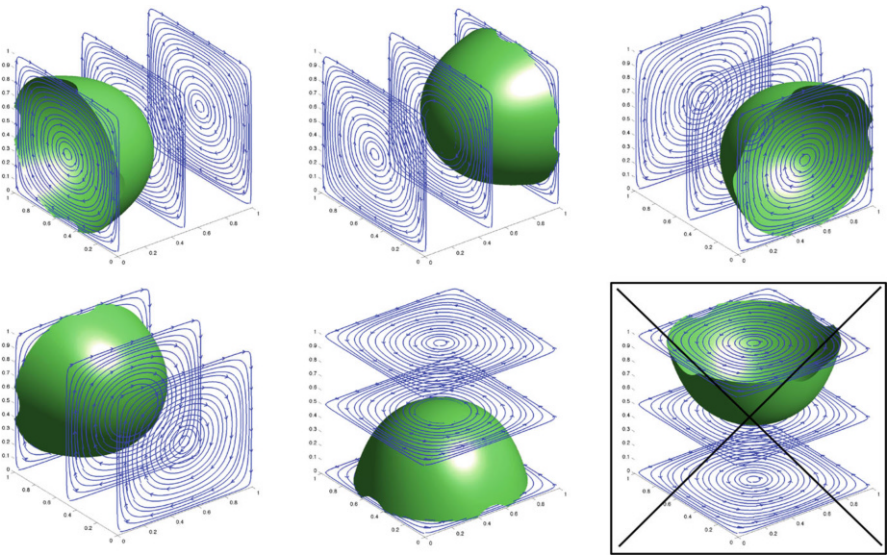


Fig. 4 Subgrid vortices of degree $p_f = 2$ on the unit cube. There are a total of six subgrid vortices, but one of the subgrid vortices can be represented as a linear combination of the other five subgrid vortices

for removing linearly dependent subgrid vortices and extracting a basis, but the method we have found to be most effective is to extract a basis from an element mass matrix. Namely, we can first form an element mass matrix from all of the subgrid vortices, and we can then perform an eigendecomposition of the element mass matrix. The eigenvectors corresponding to non-zero eigenvalues then provide the basis used to solve for the fine-scale velocity. This set of eigenvectors represents a set of subgrid vortices residing in each element, organized by their energy content, and the eigenvectors are also orthonormal with respect to the L^2-norm. This reduces the computational cost of the dynamic subgrid model.

We take a quick aside to show that the fine-scale solution fields arising from the quasi-static model and its discretization are independent of the discrete pressure field. Recall that:

$$\ell_e(\mathbf{v}') = \int_{\Omega_e} \mathbf{v}' \cdot \mathrm{Res}_m\left(\mathbf{u}_h, p_h\right)\, d\Omega_e.$$

Since:

$$\mathrm{Res}_m\left(\mathbf{u}_h, p_h\right) = \mathbf{f} - \partial_t \mathbf{u}_h - \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot \left(2\nu \nabla^s \mathbf{u}_h\right) - \nabla p_h,$$

it holds by integration by parts that:

$$\ell_e(\mathbf{v}') = \int_{\Omega_e} \mathbf{v}' \cdot \left(\mathbf{f} - \partial_t \mathbf{u}_h - \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot \left(2\nu \nabla^s \mathbf{u}_h\right)\right)\, d\Omega_e + \int_{\Omega_e} \nabla \cdot \mathbf{v}' p_h \, d\Omega_e.$$

Thus, for $\mathbf{v}' \in \mathring{\mathscr{V}}'_e$, it follows that:

$$\ell_e(\mathbf{v}') = \int_{\Omega_e} \mathbf{v}' \cdot \left(\mathbf{f} - \partial_t \mathbf{u}_h - \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot \left(2\nu \nabla^s \mathbf{u}_h\right)\right) \, d\Omega_e$$

and indeed the fine-scale solution field arising from the quasi-static model and its discretization are independent of the discrete pressure field. This same property can be shown for the dynamic model. In fact, a stronger property can be shown for both the quasi-static and dynamic models, namely that a structure-preserving VMS method employing either model is pressure-robust in that the velocity error is independent of the pressure field [48].

7 Final Form of Residual-Based Variational Multiscale Models

In this section, we present the final form of the residual-based VMS models developed in the previous sections. In the coarse-scale problem, we neglect the influence of the fine-scale unsteadiness and diffusion on the coarse-scale problem, as is commonly done in practical implementations of the VMS method [36]. This yields the following coarse- and fine-scale problems, wherein $(\cdot, \cdot)$ denotes the L^2-inner product:

Coarse-Scale Problem for both the Quasi-Static and Dynamic Models

Find $\mathbf{u}_h \in \mathscr{V}_T^h$ and $p_h \in \mathscr{Q}_T^h$ such that suitable initial conditions are satisfied and

$$a(\mathbf{v}_h; \mathbf{u}_h) - (\nabla^s \mathbf{v}_h, \mathbf{u}_h \otimes \mathbf{u}' + \mathbf{u}' \otimes \mathbf{u}_h + \mathbf{u}' \otimes \mathbf{u}') - b(\mathbf{v}_h, p_h) + b(\mathbf{u}_h, q_h) = \ell(\mathbf{v}_h) \tag{30}$$

for all $\mathbf{v}_h \in \mathscr{V}_h$, $q_h \in \mathscr{Q}_h$, and almost every $t \in (0, T)$.

Fine-Scale Problem for the Quasi-Static Model

Find $\mathbf{u}' : (0, T) \to \mathring{\mathscr{V}}'_{h,e}$ such that appropriate initial conditions are satisfied and:

$$a_e(\mathbf{v}', \mathbf{u}') = \ell_e(\mathbf{v}') \tag{31}$$

for each element Ω_e.

Fine-Scale Problem for the Dynamic Model

Find $\mathbf{u}' : (0, T) \to \mathring{\mathscr{V}}'_{h,e}$ such that appropriate initial conditions are satisfied and:

$$(\mathbf{v}', \partial_t \mathbf{u}') + a_e(\mathbf{v}', \mathbf{u}') = \ell_e(\mathbf{v}') \tag{32}$$

for each element Ω_e.

8 Residual-Based Eddy Viscosity Models

In this section, we present an alternative class of residual-based large eddy simulation models for isogeometric divergence-conforming discretizations. These models are predicated on the assumption that we can model the effect of unresolved fluid motion on the coarse-scales through viscous diffusion using an appropriately defined eddy viscosity. This yields the approximation:

$$a(\mathbf{v}_h; \mathbf{u}_h + \mathbf{u}') \approx a(\mathbf{v}_h; \mathbf{u}_h) + (2\nu_t \nabla^s \mathbf{v}_h, \nabla^s \mathbf{u}_h) \tag{33}$$

where ν_t is the eddy viscosity. Classically, the eddy viscosity has been expressed algebraically in terms of invariants of the resolved strain rate tensor [45]. Residual-based eddy viscosity (RBEV) models, which were originally proposed by Oberai et al. [44], are built on a different idea. Note that the eddy viscosity scales like:

$$\nu_t \sim \mathscr{U}\ell \tag{34}$$

where $\mathscr{U}$ and ℓ are velocity and length scales associated with the fine-scales. Thus, it is natural to select ℓ as the mesh size h, and it is natural within a VMS framework to select $\mathscr{U}$ as the magnitude of the fine-scale velocity field. This yields the prescription:

$$\nu_t = \bar{C}h\left|\boldsymbol{u}'\right| \tag{35}$$

where $\bar{C}$ is a model constant set to $\bar{C} = 0.074$ so that the subgrid approximation represents all turbulent motion at length scales smaller than the grid resolution for homogeneous isotropic turbulence [44]. We then solve the fine-scale velocity field $\boldsymbol{u}'$ using the same quasi-static and dynamic models as before. This yields the following coarse- and fine-scale problems:

Coarse-Scale Problem for both the Quasi-Static and Dynamic Models

Find $\mathbf{u}_h \in \mathscr{V}_T^h$ and $p_h \in \mathscr{Q}_T^h$ such that suitable initial conditions are satisfied and

$$a(\mathbf{v}_h; \mathbf{u}_h) + (2\nu_t \nabla^s \mathbf{v}_h, \nabla^s \mathbf{u}_h) - b(\mathbf{v}_h, p_h) + b(\mathbf{u}_h, q_h) = \ell(\mathbf{v}_h) \tag{36}$$

for all $\mathbf{v}_h \in \mathscr{V}_h$, $q_h \in \mathscr{Q}_h$, and almost every $t \in (0, T)$.

Fine-Scale Problem for the Quasi-Static Model

Find $\mathbf{u}' : (0, T) \to \mathring{\mathscr{V}}'_{h,e}$ such that appropriate initial conditions are satisfied and:

$$a_e(\mathbf{v}', \mathbf{u}') = \ell_e(\mathbf{v}') \tag{37}$$

for each element Ω_e.

Fine-Scale Problem for the Dynamic Model

Find $\mathbf{u}' : (0, T) \rightarrow \mathring{\mathcal{V}}'_{h,e}$ such that appropriate initial conditions are satisfied and:

$$(\mathbf{v}', \partial_t \mathbf{u}') + a_e(\mathbf{v}', \mathbf{u}') = \ell_e(\mathbf{v}') \tag{38}$$

for each element Ω_e.

Note that the above RBEV models are similar in appearance to the residual-based VMS models presented previously. The main difference is in the utilization of the fine-scale velocity field. In the residual-based VMS models, an expression is found for the fine-scale solution which is then substituted directly into the coarse-scale problem. In the RBEV models, the fine-scale velocity is used to construct an eddy viscosity model for the coarse-scale equations. Note that RBEV models have an automatic dynamic capability in that they return a null eddy viscosity when the flow is either laminar or fully resolved, as do residual-based VMS models.

Residual-based eddy viscosity models have been applied previously to the decay of incompressible and compressible homogeneous turbulence where they were shown to be more accurate than the dynamic Smagorinsky model [44]. Residual-based eddy viscosity models are arguably simpler than residual-based VMS models, and they are more stable in that they are purely dissipative. However, as opposed to residual-based VMS models, RBEV models are predicated on the assumption that there is a one-way transfer of energy from coarse-scales to fine-scales, which is not the case for flows with significant helicity [6].

9 Application to Taylor-Green Vortex Flow

In this section, we apply the residual-based large eddy simulation methodologies developed in the previous chapters to three-dimensional Taylor-Green vortex flow. Taylor-Green vortex flow is one of the simplest systems with which one can study the generation of small scales through vortex stretching and the energy dissipation from the resulting turbulence. The initial condition of Taylor-Green vortex flow is laminar and consists of purely two-dimensional streamlines, but for all time $t > 0$, the flow is three-dimensional. As the solution is evolved in time, vortex stretching causes the generation of small-scale motion and eventual transition into turbulence. Mathematically, the initial condition for this flow is:

$$\mathbf{u}_0(x, y, z) = \begin{bmatrix} \sin(x)\cos(y)\cos(z) \\ -\sin(x)\sin(y)\cos(z) \\ 0 \end{bmatrix}. \tag{39}$$

The initial condition can be visualized in Fig. 5. The flow is periodic in all three spatial directions in the domain $\Omega = (0, 2\pi)^3$ and, due to inherent symmetries in the flow, the flow can be modeled within a computational domain of $\Omega^h = (0, \pi)^3$ with

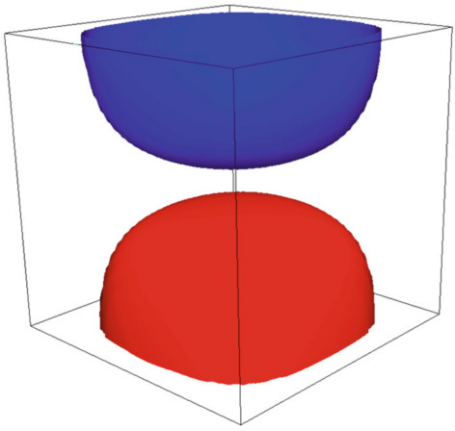


Fig. 5 Isosurfaces of enstrophy colored by z-component of vorticity for Taylor-Green vortex flow at $t = 0\,\mathrm{s}$

no-penetration and free-slip boundaries. At low Reynolds numbers ($Re < 400$), the flow is anisotropic for all time, but as the Reynolds number increases, the flow experiences increased isotropy [8].

To assess the performance of the new residual-based large eddy simulation methodologies, we compare them with four standard eddy viscosity based approaches: the Smagorinsky model, the WALE model, the Vreman model, and the Sigma model. In the Smagorinsky model, the eddy viscosity is calculated as:

$$\nu_t = (C_s \Delta)^2 |\boldsymbol{S}| \tag{40}$$

where Δ is the filter width, $\boldsymbol{S}$ is the resolved rate of strain tensor, and C_s is the Smagorinsky constant [40]. For homogeneous isotropic turbulence, a value of $C_s = 0.18$ has been shown to yield satisfactory results, but smaller values have been required for other applications. Dynamic procedures can also be employed to locally adapt the Smagorinsky constant [28], but with one exception, we adapted a static Smagorinsky constant of $C_s = 0.18$ in our simulations since the new residual-based models do not require an optimization procedure to tune model constants. The filter width was also chosen as the mesh size in all our simulations.

In the WALE model, the traceless symmetric part of the square of the velocity gradient tensor is used as an operator rather than the rate of strain tensor. This operator is chosen as it goes naturally to zero near the wall without the use of any damping functions nor dynamic procedure [43]. Using this operator, the eddy viscosity is calculated as:

$$\nu_t = (C_w \Delta)^2 \frac{\left(S^d_{ij} S^d_{ij}\right)^{\frac{3}{2}}}{\left(S_{ij} S_{ij}\right)^{\frac{5}{2}} + \left(S^d_{ij} S^d_{ij}\right)^{\frac{5}{4}}} \tag{41}$$

where $\boldsymbol{S}^d$ is the traceless symmetric part of the square of the resolved velocity gradient tensor:

$$S_{ij}^d = \frac{1}{2}\left(g_{ij}^2 + g_{ij}^2\right) - \frac{1}{3}\delta_{ij} g_{kk}^2, \tag{42}$$

δ_{ij} is the Kronecker delta symbol, the components of the resolved velocity gradient tensor $\boldsymbol{g}$ are defined as:

$$g_{ij} = \frac{\partial \bar{u}_i}{\partial x_j}, \tag{43}$$

Δ is the filter width, $\boldsymbol{S}$ is the resolved rate of strain tensor, and C_w is a constant. In the original work which introduced the model, an ideal value of C_w in the range $0.55 \leq C_w \leq 0.60$ was reported for homogeneous turbulence [43], but later studies by the authors suggest a lower value of $C_w = 0.325$ is more appropriate [1]. A value of $C_w = 0.325$ was employed in the simulations reported here.

In the Vreman model, an alternative operator is used which is designed to produce very low dissipation in transitional and near-wall regions. The eddy viscosity is calculated as:

$$\nu_t = 2.5 C_s^2 \sqrt{\frac{B_\beta}{\alpha_{ij}\alpha_{ij}}} \tag{44}$$

with:

$$\alpha_{ij} = \frac{\partial \bar{u}_j}{\partial x_i} \tag{45}$$

$$\beta_{ij} = \Delta_m^2 \alpha_{mi}\alpha_{mj} \tag{46}$$

$$B_\beta = \beta_{11}\beta_{12} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2 \tag{47}$$

where C_s is the Smagorinsky constant [55].

Finally, in the Sigma model, the subgrid stress is related to the singular values of the resolved velocity gradient tensor. The operator of this model is constructed to deliver zero subgrid stress in cases of two-component or two-dimensional flow as well as in cases of pure axisymmetric or isotropic contraction/expansion. Furthermore, it has a cubic behavior near solid boundaries and thus naturally damps near walls. The eddy viscosity is calculated as:

$$\nu_t = (C_\sigma \Delta)^2 \mathscr{D}_\sigma \tag{48}$$

with:

$$\mathscr{D}_\sigma = \frac{\sigma_3 \left(\sigma_1 - \sigma_2\right)\left(\sigma_2 - \sigma_3\right)}{\sigma_1^2} \tag{49}$$

where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ are the three singular values of the velocity gradient tensor, $\boldsymbol{g}$ [42]. The value of the constant, C_σ must be tuned based on the flow problem, the grid resolution, and numerics, but a value of $C_\sigma = 1.35$ has been reported to provide results comparable to the classical Smagorinsky model for homogeneous isotropic turbulence [42]. A value of $C_\sigma = 1.35$ was employed in the simulations reported here.

For all of the above models as well as the residual-based large eddy simulation methodologies presented in this chapter, Taylor-Green vortex flow at $Re = 1600$ was simulated. The simulations were initialized using the initial conditions defined in (39), and the simulations were advanced for 10 s. This is a long enough time for the flow to develop and transition into fully turbulent flow. A time step size of $\Delta t = 0.125$ h was used for all simulations to ensure that any temporal discretization error was dominated by spatial discretization error and modeling error. For all simulations, the Crank-Nicolson method was employed to discretize viscous terms while the Adams-Bashforth multi-step method [2] was employed to discretize nonlinear terms. All of the results were then compared with Fourier-based DNS results using between 256^3 and 512^3 modes [8, 52].

The quantities of interest that we consider in what follows are the time history of the kinetic energy dissipation rate, the Q-criterion, and the energy spectrum for the solution at the final time instance of the simulation. The kinetic energy, E_k, of the flow is given by:

$$E_k\left(t\right) = \frac{1}{|\Omega|}\int_\Omega \frac{\mathbf{u}\left(t\right)\cdot\mathbf{u}\left(t\right)}{2} d\Omega. \tag{50}$$

The kinetic energy dissipation rate, ϵ, can be computed as the time derivative of E_k:

$$\epsilon = -\frac{dE_k}{dt}. \tag{51}$$

For incompressible flow, the kinetic energy dissipation rate is directly related to the enstrophy, η:

$$\epsilon = 2\nu\eta \tag{52}$$

where the enstrophy is computed as:

$$\eta = \frac{1}{|\Omega|}\int_\Omega \frac{\boldsymbol{\omega}\cdot\boldsymbol{\omega}}{2} d\Omega, \tag{53}$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity. For large eddy simulation, application of (52) to the resolved velocity field results in the *resolved dissipation rate*. This represents the amount of kinetic energy dissipation that is present in the coarse scales. The difference of the *total dissipation rate* and the *resolved dissipation rate* allows for the calculation of the *model dissipation rate*, which gives a measure of how much dissipation a particular subgrid scale model provides. Finally, the Q-criterion represents the local balance between vorticity magnitude and shear strain rate, and identifies vortices as regions where the vorticity magnitude is greater than the magnitude of the rate of strain. The Q-criterion is calculated as:

$$Q = \frac{1}{2}\left(|\boldsymbol{\Omega}|^2 - |\mathbf{S}|^2\right), \tag{54}$$

where $\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i}\right)$ [37].

Before presenting the results for the residual-based models, we present a comparison of traditional eddy-viscosity models. First, the time histories of the dissipation rate are presented for simulations with $h = \frac{\pi}{32}$ and $p = 3$ and the Smagorinsky, WALE, Vreman, and Sigma eddy viscosity models in Fig. 6. Note that while $p = 3$, the velocity and pressure spaces are complete only to quadratic polynomials. Furthermore, note that the mesh is coarse and the flow is quite unresolved. This is intentional, as we would like to examine the performance of the models in the unresolved setting. All of the models appear to predict the same time evolution of the dissipation rate, and they actually demonstrate a decrease in performance from the no-model baseline case. All of the models are overly

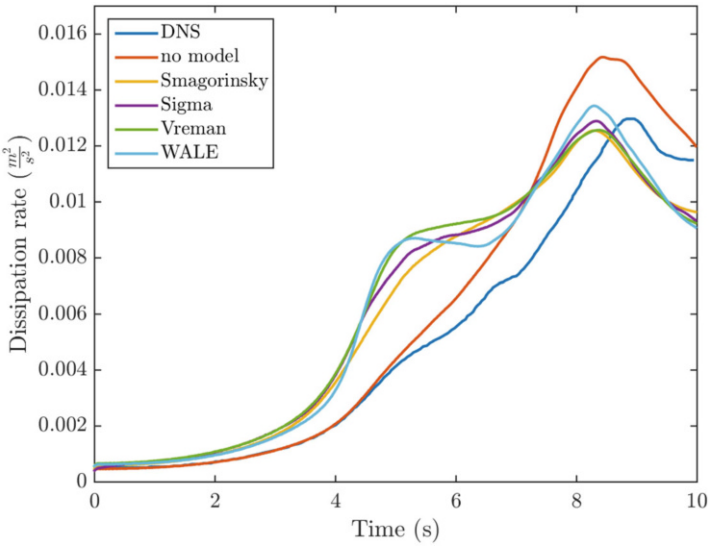


Fig. 6 Time history of total dissipation rate for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$ and and $h = \frac{\pi}{32}$. Comparison of traditional subgrid-scale models (no model, Smagorinsky, Sigma, Vreman, WALE) with DNS [8]

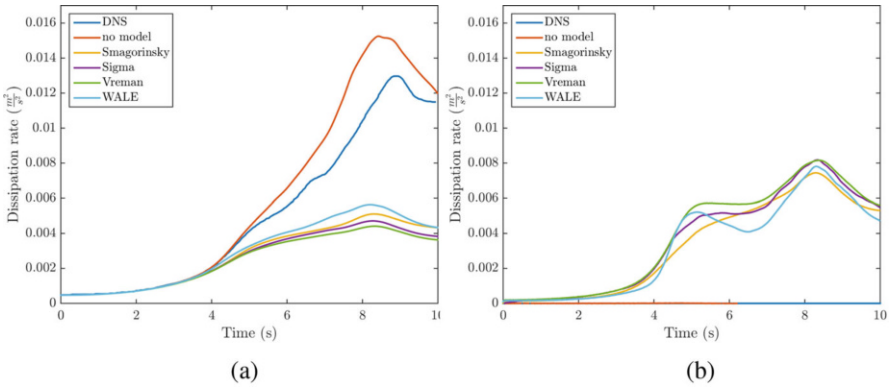


Fig. 7 Time history of resolved and model dissipation rate for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$ and $h = \frac{\pi}{32}$. Comparison of traditional subgrid-scale models (no model, Smagorinsky Sigma, Vreman, WALE) with DNS [8]. (**a**) Resolved dissipation. (**b**) Model dissipation

dissipative, and they depart from the DNS from the very beginning of the simulation. This suggests that all of the models are active even at the beginning of the simulation when the solution is purely laminar. The models also yield a prediction of the peak dissipation approximately one second earlier than the DNS data.

We can further elucidate the various models' response to turbulence production and transition by examining the resolved and model dissipation rates. The time histories of both of these parameters for the same set of simulations are presented in Fig. 7. Between the four models, there is not very much difference in the resolved dissipation. The model dissipation reflects the same characteristics that were observed with the total dissipation. At $t = 0$, all of the models have non-zero model dissipation. Interestingly, all of the models exhibit more model dissipation than resolved dissipation for almost all time.

In the next set of results, we evaluate the performance of the residual-based models. Since the Sigma, Vreman, and WALE models did not show marked improvement over the Smagorinsky model, we have selected the Smagorinsky model as a representative eddy-viscosity model by which to measure the performance of the residual-based models. First, the time histories of the dissipation rate are presented for simulations with $h = \frac{\pi}{32}$, $p = 3$, and $p_f = 2$ in Fig. 8. From the figure, it is clear that all of the residual-based models outperform the Smagorinsky model. Furthermore, the dynamic models are more accurate than the quasi-static models. It should be noted that in these simulations, (19) was employed to define τ_{uns} for the quasi-static models. Since the Smagorinsky model has no mechanism to reduce the eddy viscosity in laminar flows, it departs from the DNS solution very early in the simulation. All of the residual-based models depart from the DNS at roughly the same point in time, later than the Smagorinsky model, indicating that in all four cases, the model is appropriately "turned off" when the solution lacks subgrid-scale activity. Interestingly, at later times, both of the quasi-static residual-

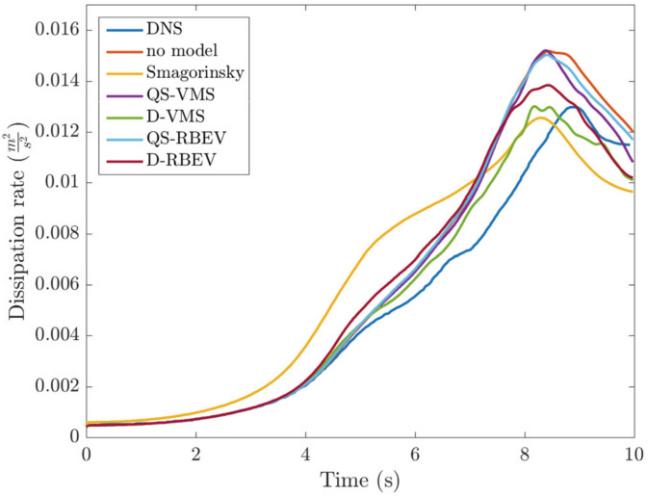


Fig. 8 Time history of total dissipation rate for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of subgrid-scale models (no model, Smagorinsky, quasi-static VMS (QS-VMS), dynamic VMS (D-VMS), quasi-static RBEV (QS-RBEV), and dynamic RBEV (D-RBEV)) with DNS [8]

based models become overly dissipative. Although both dynamic models have lower kinetic energy than the DNS after the midpoint in the simulation, by the late times both models match the DNS in terms of the dissipation rate.

In order to better understand how the residual-based models respond to turbulence production and transition, we again examine the resolved and model dissipation rates. The time histories of both of these parameters for the same set of simulations are presented in Fig. 9. It is clear that the quasi-static models are not activated until very late in the simulation, and even then they produce very little modeling. The two dynamic models show similar performance.

In Fig. 10, we present the kinetic energy spectra for the various models at the end time of the simulation. This occurs shortly after the point of maximum dissipation, so the solution is fully turbulent with structures encompassing the full range of turbulent scales present. From the spectra, it is clear that the dynamic models and the Smagorinsky model exhibit the expected energy cascade. At this point, the flow is essentially a "box of turbulence," so the Smagorinsky model performs well. Note, however, that it has lower energy overall due to its overly dissipative nature at earlier times. The quasi-static models demonstrate some slight energy pileup in the higher modes.

The displayed results so far have suggested that the quasi-static residual-based models do not produce the correct amount of dissipation required to yield acceptable results. A closer look at the construction of the quasi-static models provides some insight into what is responsible for the poor performance of the models. Namely, for the simulations reported on here, a very small time step size was employed, rendering τ_{qs} so small as to essentially "turn off" the fine-scales in the quasi-static

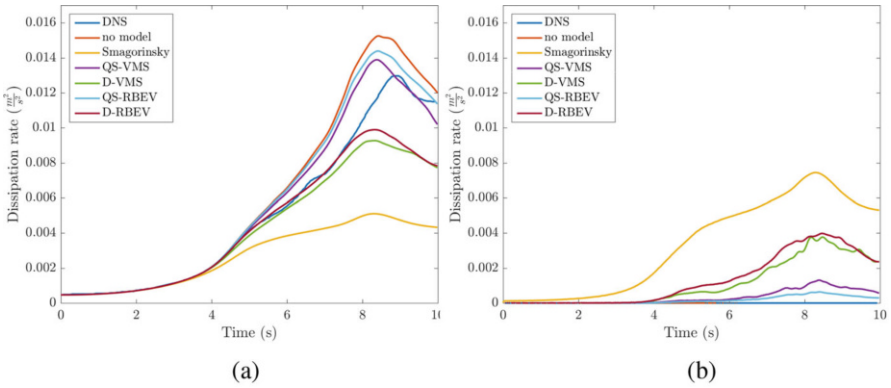


Fig. 9 Time history of resolved and model dissipation rate for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of subgrid-scale models (no model, Smagorinsky, quasi-static VMS (QS-VMS), dynamic VMS (D-VMS), quasi-static RBEV (QS-RBEV), and dynamic RBEV (D-RBEV)) with DNS [8]. (**a**) Resolved dissipation. (**b**) Model dissipation

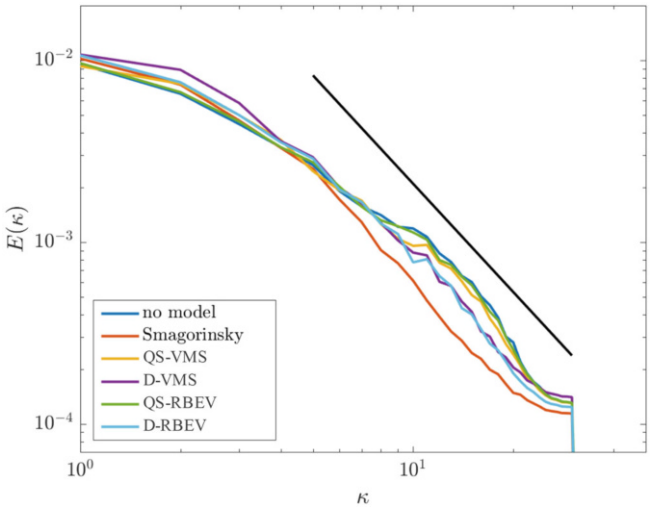


Fig. 10 Energy spectra for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of subgrid-scale models (no model, Smagorinsky, quasi-static VMS (QS-VMS), dynamic VMS (D-VMS), quasi-static RBEV (QS-RBEV), and dynamic RBEV (D-RBEV)) with DNS [8]

models. In order to explore the possibility of eliminating the sensitivity of τ_{sq} to the time step size, we now consider the quasi-static VMS model using τ_{qs} defined using the alternate formulation for τ_{uns} given by (20). The total dissipation rate for the alternate quasi-static VMS model is compared with that of the dynamic VMS model in Fig. 11. It is clear here that the quasi-static VMS model using the

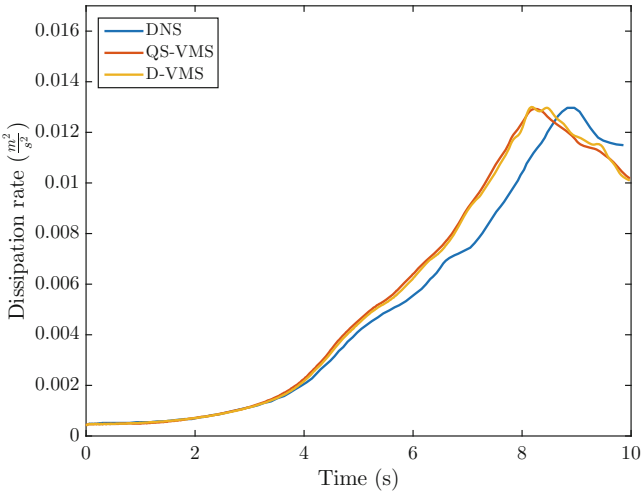


Fig. 11 Time history of total dissipation rate for simulation of the Taylor-Green vortex at Re = 1600 with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of the dynamic VMS model (D-VMS) and the quasi-static VMS model using τ_{uns} defined by (20) (QS-VMS) with DNS [8]

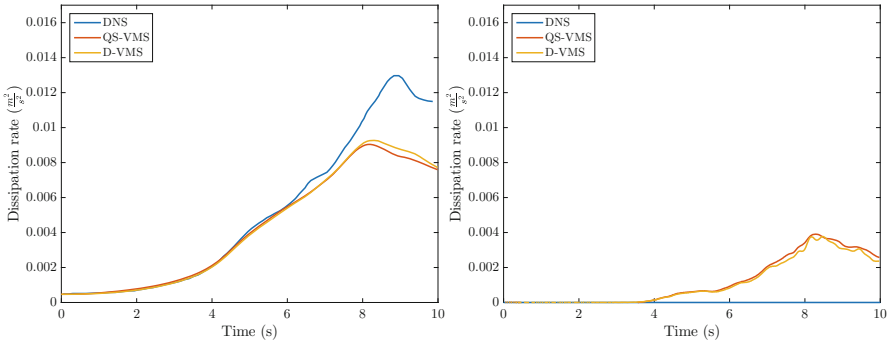


Fig. 12 Time history of resolved and model dissipation rate for simulation of the Taylor-Green vortex at Re = 1600 with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of the dynamic VMS model (D-VMS) and the quasi-static VMS model using τ_{uns} defined by (20) (QS-VMS) with DNS [8]. (**a**) Resolved dissipation. (**b**) Model dissipation

alternate formulation of τ_{uns} provides results similar to the dynamic VMS model. By examining the resolved and model dissipation rates, which are reported in Fig. 12, it is further apparent that the alternate formulation for τ_{uns} is effective in activating the fine scales. The result is an increase in the model dissipation rate and a corresponding decrease in the resolved dissipation rate.

Up to this point, we have compared simulation results directly with DNS data. However, for highly unresolved simulations, such as the ones reported here, the

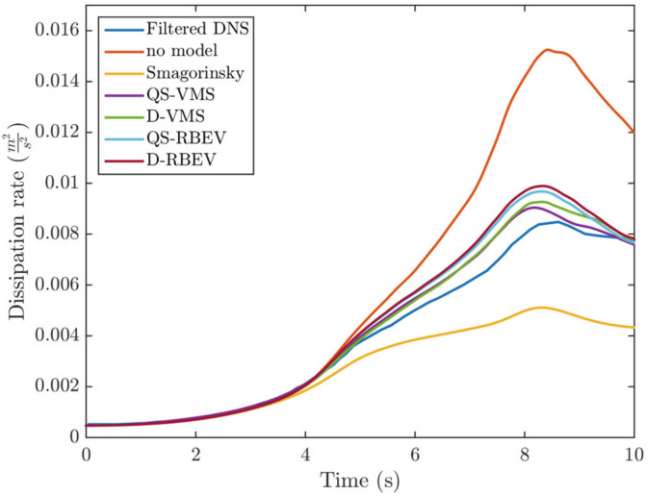


Fig. 13 Time history of resolved dissipation rate for simulation of the Taylor-Green vortex at Re = 1600 with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of subgrid-scale models (no model, Smagorinsky, quasi-static VMS (QS-VMS), dynamic VMS (D-VMS), quasi-static RBEV (QS-RBEV), and dynamic RBEV (D-RBEV)) with filtered DNS data [26]

energy in the coarse-scale velocity field (as defined by the projection of the exact velocity field into the coarse-scale space) is expected to be less than the energy in the total velocity field. As such, rather than compare the time history of the total dissipation rate for a given model with that of a DNS, it is arguably more appropriate to compare the time history of the resolved dissipation rate with that associated with filtered DNS data. With this in mind, the time histories of the resolved dissipation rate for the Smagorinsky, VMS, and RBEV models with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$ are displayed in Fig. 13 alongside the time history of the dissipation rate computed from filtered DNS data [26]. The filtered DNS data was obtained using a sharp cut-off filter of size $\Delta = \frac{\pi}{32}$. From the figure, we see that the resolved dissipation rate time histories associated with the residual-based models roughly match the resolved dissipation rate time history computed from the filtered DNS data. This indicates that the residual-based models are providing the correct amount of dissipation to yield an optimal (in the sense of projection) coarse-scale velocity field. The Smagorinsky model, on the other hand, is much too dissipative, and the resolved dissipation rate is significantly smaller than the resolved dissipation rate computed from the filtered DNS data. It should be noted that the definition of τ_{uns} given by (20) was employed here, which explains the good behavior of the quasi-static residual-based models. The VMS models perform marginally better than the RBEV models.

Another way to compare the performance of the models is to look at the character of the vortical structures present in the various solutions. Contours of $Q = 1$ colored by vorticity at a late time shortly after the time of maximum dissipation ($t = 9.57$) for the Smagorinsky and dynamic VMS models are compared in Fig. 14.

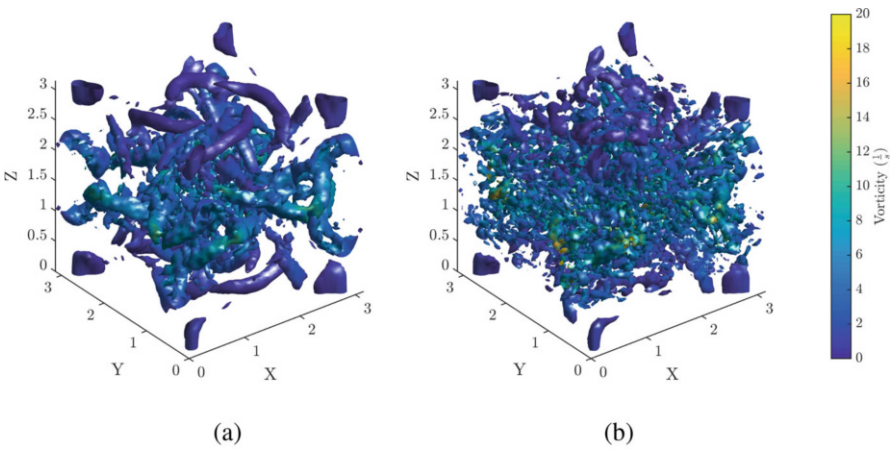


Fig. 14 Contours of $Q = 1$ colored by vorticity for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. Comparison of Smagorinsky and dynamic VMS models. (**a**) Smagorinsky model. (**b**) Dynamic VMS model

Compared to the dynamic VMS model, the Smagorinsky model damps out the smallest turbulent structures. Only the larger vortical structures, with lower vorticity, remain. In contrast, the dynamic VMS model more accurately predicts turbulent structures in the smallest length scales that can be represented on the mesh.

To illustrate how the residual-based models are activating, contours of $Q = 1$ colored by vorticity are plotted alongside the magnitude of the fine scale velocity for the dynamic VMS model in Fig. 15. Solutions at an early time ($t = 0.25$), an intermediate time ($t = 4.66$), and a late time shortly after the time of maximum dissipation ($t = 9.57$) are presented in order to provide a comparison of the character of the solutions during various phases of transition from laminar to turbulent flow. It is clear that the fine scales grow in amplitude as expected in regions where more small-scale turbulent structures develop and the fine scale velocities remain low in amplitude where only large-scale structures that can be resolved by the mesh are present.

We now turn our attention to consistency of the residual-based models with respect to mesh refinement. We limit our presentation in this regard to the dynamic models since they have so far outperformed the quasi-static models using a τ_{uns} given by (19) and performed similarly to the quasi-static models using a τ_{uns} given by (20). In Fig. 16, the time histories of the total, resolved, and model dissipation rate for the dynamic VMS and RBEV models are presented for various mesh sizes. It is clear here that as the mesh is refined, the total dissipation rate for both models converge to the DNS solution. This is accompanied by an appropriate increase in the resolved dissipation rate as well as a corresponding decrease in the model dissipation rate. It is also noted that the dynamic VMS model outperforms the dynamic RBEV model for both coarse and fine meshes.

We conclude by examining the effectiveness of the residual-based models at capturing the dissipation rate for Taylor-Green vortex flow for a range of Reynolds

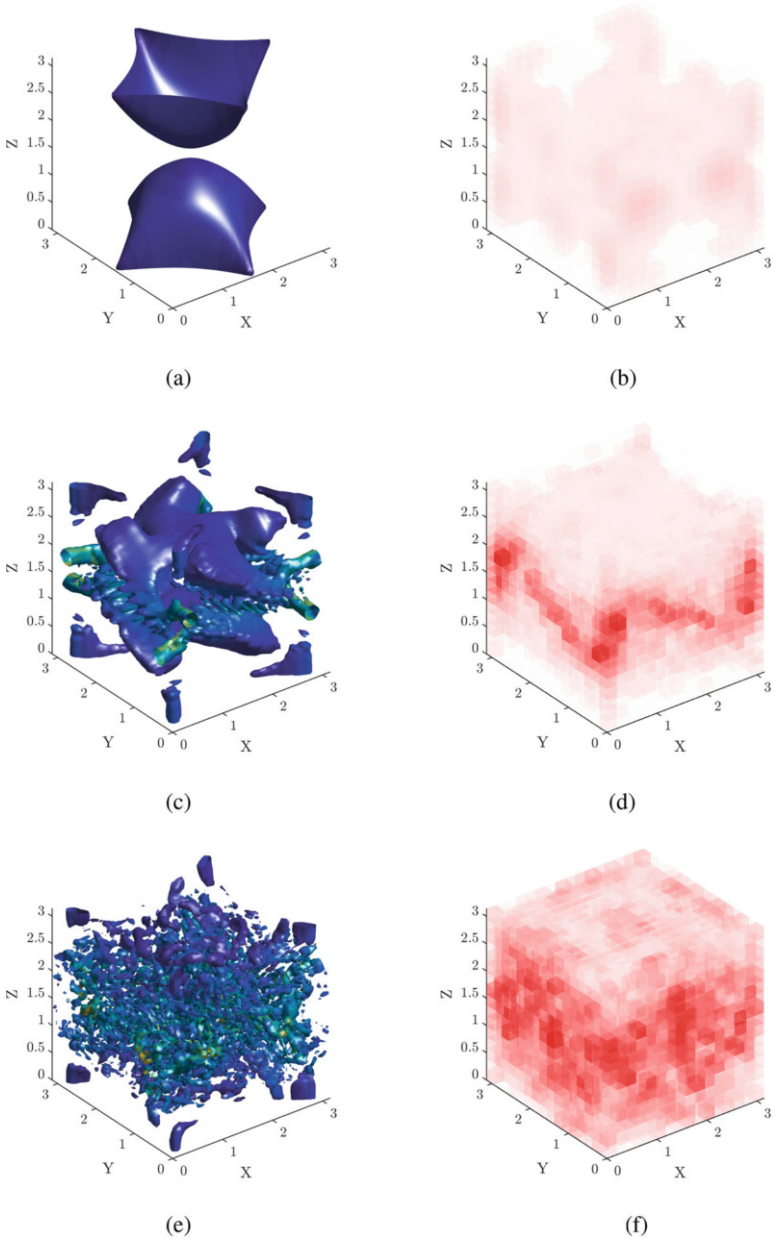


Fig. 15 Q-criterion contours colored by vorticity and magnitude of fine scale velocity for simulation of the Taylor-Green vortex at $Re = 1600$ with $p = 3$, $p_f = 2$, and $h = \frac{\pi}{32}$. (**a**) Contours of $Q = 1$ at $t = 0.25$. Dynamic VMS model. (**b**) Fine scales at $t = 0.25$. Dynamic VMS model. (**c**) Contours of $Q = 1$ at $t = 4.66$. Dynamic VMS model. (**d**) Fine scales at $t = 4.66$. Dynamic VMS model. (**e**) Contours of $Q = 1$ at $t = 9.57$. Dynamic VMS model. (**f**) Fine scales at $t = 9.57$. Dynamic VMS model

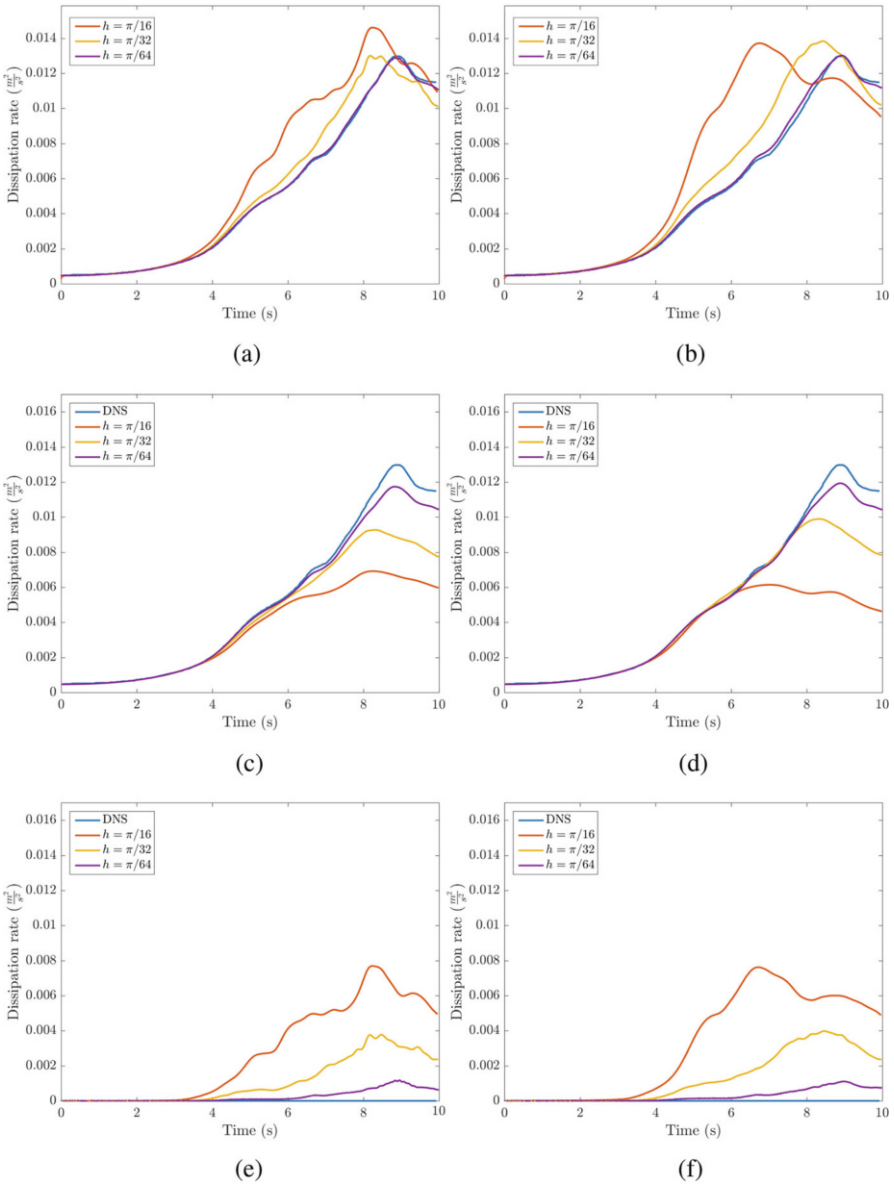


Fig. 16 Time history of total, resolved, and model dissipation rate for simulation of the Taylor-Green vortex at $Re = 1600$ under mesh refinement with $p = 3$ and $p_f = 2$. (**a**) Total dissipation rate. Dynamic VMS model. (**b**) Total dissipation rate. Dynamic RBEV model. (**c**) Resolved dissipation rate. Dynamic VMS model. (**d**) Resolved dissipation rate. Dynamic RBEV model. (**e**) Model dissipation rate. Dynamic VMS model. (**f**) Model dissipation rate. Dynamic RBEV model

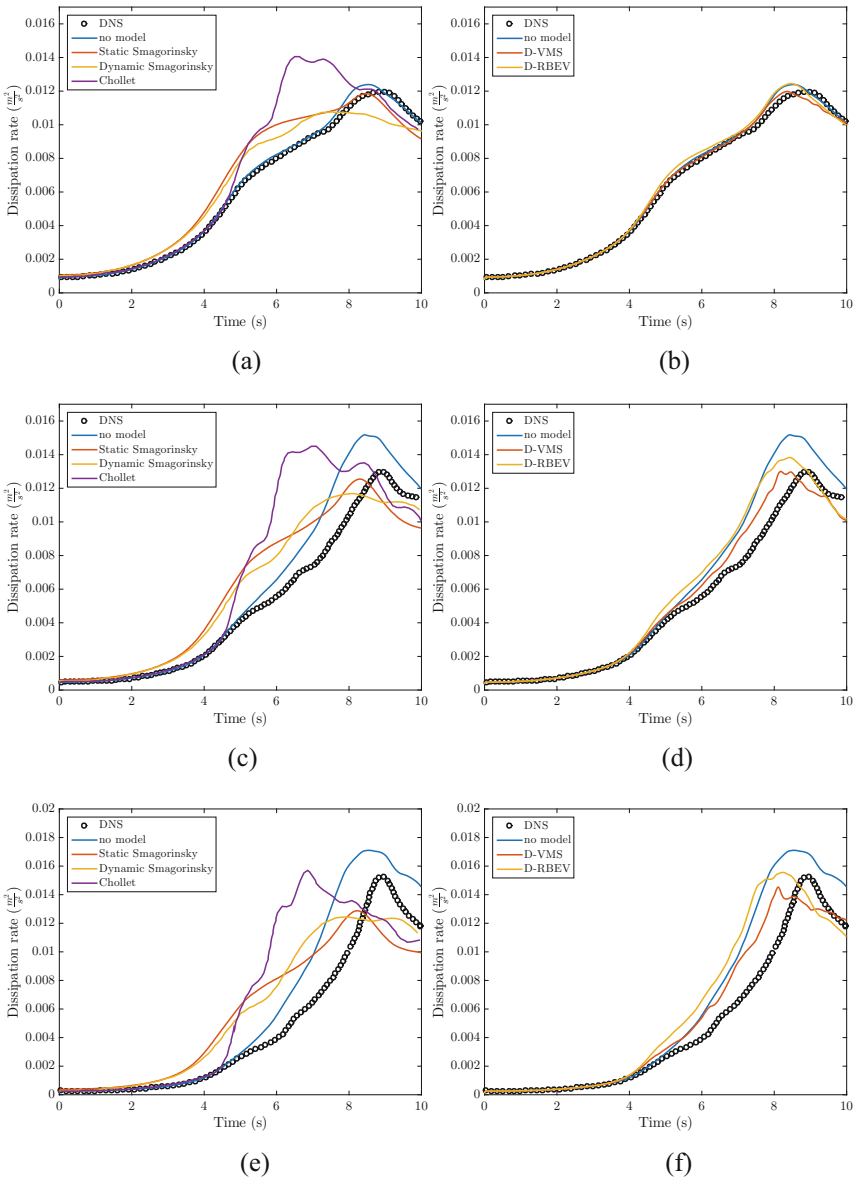


Fig. 17 Time history of total dissipation rate for simulation of the Taylor-Green vortex at various Re with $h = \frac{\pi}{32}$, $p = 3$, and $p_f = 2$ for a variety of subgrid models and DNS [8, 52]. Left: Assessment of static Smagorinsky, dynamic Smagorinsky, and Chollet subgrid. Right: Assessment of dynamic VMS and RBEV subgrid models. (**a**) Total dissipation rate for $Re = 800$. Standard eddy viscosity models. (**b**) Total dissipation rate for $Re = 800$. Residual-based models. (**c**) Total dissipation rate for $Re = 1600$. Standard eddy viscosity models. (**d**) Total dissipation rate for $Re = 1600$. Residual-based models. (**e**) Total dissipation rate for $Re = 3000$. Standard eddy viscosity models. (**f**) Total dissipation rate for $Re = 3000$. Residual-based models

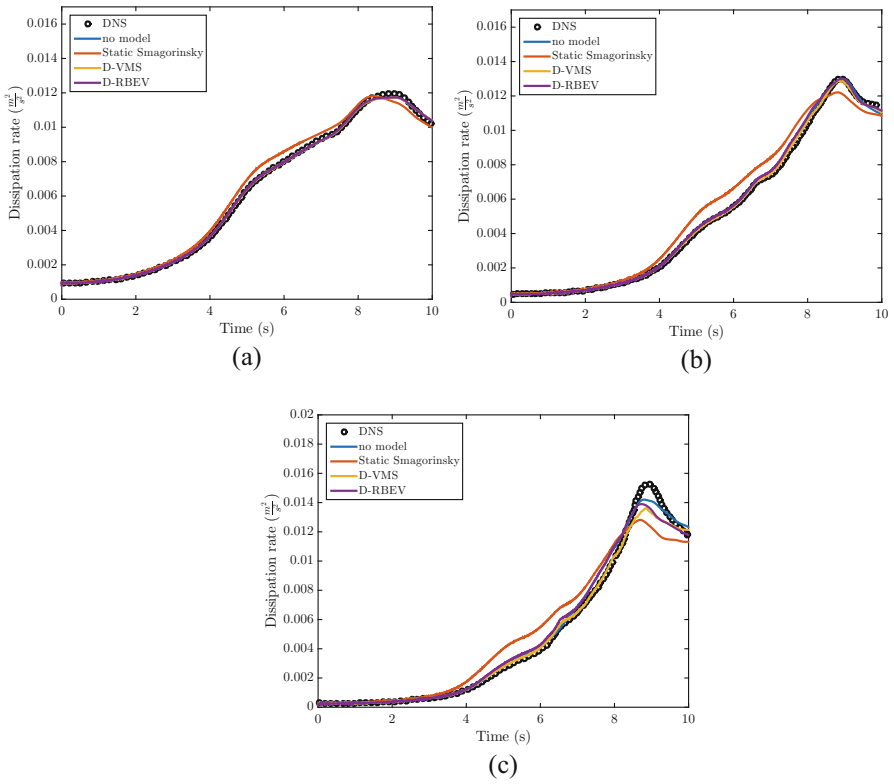


Fig. 18 Time history of total dissipation rate for simulation of the Taylor-Green vortex at various Re with $h = \frac{\pi}{64}$, $p = 3$, and $p_f = 2$ for the dynamic VMS and RBEV subgrid models and the static Smagorinsky model versus DNS [8, 52]. (**a**) Total dissipation rate for $Re = 800$. (**b**) Total dissipation rate for $Re = 1600$. (**c**) Total dissipation rate for $Re = 3000$

numbers. We first consider $h = \frac{\pi}{32}$, $p = 3$, and $p_f = 2$. In Fig. 17, we display the total dissipation rate for both the dynamic VMS and RBEV models for $Re = 800$, $Re = 1600$, and $Re = 3000$. We also display the total dissipation rate for the dynamic Smagorinsky [28] model, which improves upon the classical Smagorinsky model when the flow is transitional, and the Chollet [12] model. Note that the dynamic VMS and RBEV models match the DNS data far better than the dynamic Smagorinsky and the Chollet models for each Reynolds number considered. Impressively, the results for the dynamic VMS and RBEV models are indistinguishable from the DNS results for $Re = 800$. The results for the dynamic VMS and RBEV models at higher Re are not quite as impressive, though the dynamic VMS and RBEV models still perform much better than the dynamic Smagorinsky and Chollet models. Moreover, at these Reynolds numbers, we do not necessarily expect the dynamic VMS and RBEV models to exactly match the DNS data as the energy in the filtered scales is likely less than the total energy.

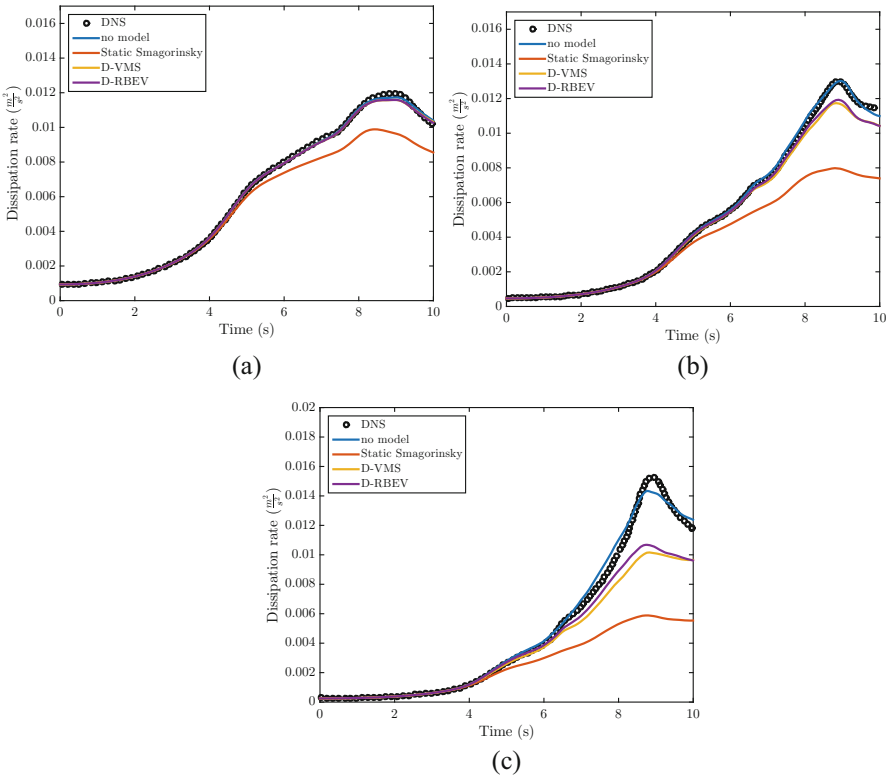


Fig. 19 Time history of resolved dissipation rate for simulation of the Taylor-Green vortex at various Re with $h = \frac{\pi}{64}$, $p = 3$, and $p_f = 2$ for the dynamic VMS and RBEV subgrid models and the static Smagorinsky model versus DNS [8, 52]. (**a**) Resolved dissipation rate for $Re = 800$. (**b**) Resolved dissipation rate for $Re = 1600$. (**c**) Resolved dissipation rate for $Re = 3000$

We next consider a refined mesh, namely, $h = \frac{\pi}{64}$, $p = 3$, and $p_f = 2$. In Fig. 18, we display the total dissipation rate for both the dynamic VMS and RBEV models as well as the (static) Smagorinsky model for $Re = 800$, $Re = 1600$, and $Re = 3000$. Note that the results for the dynamic VMS and RBEV models are indistinguishable from the DNS results for $Re = 800$ and $Re = 1600$, and they nearly match the DNS results for $Re = 3000$ except near the time of max dissipation rate which they slightly underpredict. The Smagorinsky model performs considerably worse by comparison, and it overpredicts the dissipation rate at early times, underpredicts the time of max dissipation rate, and underpredicts the max dissipation rate for each of the three Reynolds numbers. The results in Fig. 18 seem to suggest that the dynamic VMS and RBEV models are not active for any of the Reynolds numbers since the no model, dynamic VMS, and dynamic RBEV total dissipation rates are so similar. However, an examination of the resolved dissipation rate time histories for each of the models, which are depicted in Fig. 19, reveals this is not the case. From this

figure, we see that the dynamic models are quite active near the time of maximum dissipation, especially for the $Re = 3000$ case, so it is likely that energy pileup in the no model case is responsible for the similar total dissipation rate time histories between the no model, dynamic VMS, and dynamic RBEV cases.

10 Conclusions

In this chapter, a novel paradigm for the large eddy simulation of turbulent flows utilizing residual-based subgrid modeling has been presented. By combining isogeometric divergence-conforming discretizations with structure-preserving variational multiscale subgrid modeling, a large eddy simulation simulation capability which addresses many of the shortcomings of classical large eddy simulation approaches has been developed. In particular, the new subgrid models appropriately turn off when the flow is either laminar or fully resolved in contrast with standard eddy viscosity approaches. Furthermore, the new subgrid models contain no "tunable" parameters, and thus the models can be applied across a wide range of flows. The models come in two general categories, namely residual-based variational multiscale models and residual-based eddy viscosity models, and quasi-static and dynamic models were developed for each of these categories. For a transitional flow problem, the models were seen to outperform not only static eddy viscosity models such as the Smagorinsky, WALE, Vreman, and Sigma models but also the dynamic Smagorinsky model. Of all of the presented residual-based models, the dynamic variational multiscale model yielded the most promising results.

The models are currently being applied to the large eddy simulation of wall-bounded and rotational flows, and initial results are very encouraging. In these simulations, no-slip boundary conditions are being enforced in a weak manner using a combination of Nitsche's method for diffusive terms and upwinding for convection terms [4]. Such an approach may be interpreted as a variationally consistent wall model [5]. In the future, the models presented here will be further enhanced by improving the modeling of the fine-scale problem, in light of promising recent work in the arena of scalar transport [16].

Acknowledgements This material is based upon work supported by the Air Force Office of Scientific Research under Grant No. FA9550-14-1-0113.

References

1. Wall-adapting local eddy viscosity (WALE) model. In *ANSYS Fluent Theory Guide*.
2. F. Bashforth and J. C. Adams. *Theories of Capillary Action*. Cambridge University Press, London, 1883.
3. 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*, 197:173–201, 2007.

4. Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.
5. Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.
6. L. Biferale, S. Musacchio, and F. Toschi. Inverse energy cascade in three-dimensional isotropic turbulence. *Physical Review Letters*, 108:164501, 2012.
7. M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.
8. M. Brachet, D. Meiron, B. Nickel, and R. Morf. Small-scale structure of the Taylor-Green vortex. *Journal of Fluid Mechanics*, 130:411–452, 1983.
9. D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Applied Numerical Mathematics*, 23:3–19, 1997.
10. A. Buffa, C. de Falco, and R. Vázquez. Isogeometric analysis: Stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65:1407–1422, 2011.
11. A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric discrete differential forms in three dimensions. *SIAM Journal on Numerical Analysis*, 49:818–844, 2011.
12. J.-P. Chollet and M. Lesieur. Parameterization of small scales of three-dimensional isotropic turbulence utilizing spectral closures. *Journal of the Atmospheric Sciences*, 38:2747–2757, 1981.
13. R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196:2413–2430, 2007.
14. C. Coley. *Residual-based large eddy simulation of turbulent flows using divergence-conforming discretizations*. PhD thesis, University of Colorado Boulder, 2017.
15. C. Coley, J. Benzaken, and J. A. Evans. A geometric multigrid method for isogeometric compatible discretizations of the generalized Stokes and Oseen problems. *Numerical Linear Algebra with Applications*, 25:e2145, 2018.
16. C. Coley and J.A. Evans. Variational multiscale modeling with discontinuous subscales: Analysis and application to scalar transport. *Meccanica*, 53:1241–1269, 2018.
17. J. A. Cottrell. *Isogeometric analysis and numerical modeling of the fine scales within the variational multiscale method*. PhD thesis, University of Texas at Austin, 2007.
18. J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009.
19. J. A. Evans. *Divergence-free B-spline discretizations for viscous incompressible flows*. PhD thesis, The University of Texas at Austin, 2011.
20. J. A. Evans and T. J. R. Hughes. Discrete spectrum analyses for various mixed discretizations of the Stokes eigenproblem. *Computational Mechanics*, pages 1–8, 2012.
21. J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the Darcy–Stokes–Brinkman equations. *Mathematical Models and Methods in Applied Sciences*, 23:671–741, 2013.
22. J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations. *Mathematical Models and Methods in Applied Sciences*, 23:1421–1478, 2013.
23. J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.
24. J. A. Evans and T. J. R. Hughes. Isogeometric compatible discretizations for viscous incompressible flow. In *IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs*, pages 155–193. Springer, 2016.
25. R.S. Falk and M. Neilan. Stokes complexes and the construction of stable finite elements with pointwise mass conservation. *SIAM Journal on Numerical Analysis*, 51:1308–1326, 2013.
26. D. Fauconnier. *Development of a dynamic finite difference method for large-eddy simulation*. PhD thesis, Ghent University, 2008.

27. F. Gaspar, Y. Notay, C. Oosterlee, and C. Rodrigo. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM Journal on Scientific Computing*, 36:A1187–A1206, 2014.
28. M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids*, 3:1760–1765, 1991.
29. I. Harari and T.J.R. Hughes. What are C and h?: Inequalities for the analysis and design of finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 97:157–192, 1992.
30. M. C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.
31. 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*, 127:387–401, 1995.
32. T. J. R. Hughes, G. R. Feijó, L. Mazzei, and J. B. Quincy. The variational multiscale method–a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.
33. T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.
34. T. J. R. Hughes, L. Mazzei, and K. J. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization Science*, 3:47–59, 2000.
35. T. J. R. Hughes and G. Sangalli. Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods. *SIAM Journal on Numerical Analysis*, 45:539–557, 2007.
36. T.J.R. Hughes, G. Scovazzi, and L.P. Franca. Multiscale and stabilized methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–64, 2018.
37. J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flows. *Center for Turbulence Research Proceedings of the Summer Program 1988*, 1988.
38. K. A. Johannessen, M. Kumar, and T. Kvamsdal. Divergence-conforming discretization for Stokes problem on locally refined meshes using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 293:38–70, 2015.
39. D. Kamensky, M.-C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.
40. D. K. Lilly. On the application of the eddy viscosity concept in the inertial sub-range of turbulence. *NCAR Manuscript 123*, 1966.
41. J.-C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35:315–341, 1980.
42. F. Nicoud, H. Baya Toda, O. Cabrit, S. Bose, and J. Lee. Using singular values to build a subgrid-scale model for large eddy simulations. *Physics of Fluids*, 23, 2011.
43. F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Turbulence and Combustion*, 62:183–200, 1999.
44. A. A Oberai, J. Liu, D. Sondak, and T. J. R. Hughes. A residual based eddy viscosity model for the large eddy simulation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 282:54–70, 2014.
45. S. B. Pope. *Turbulent Flows*. IOP Publishing, 2001.
46. P. Raviart and J. Thomas. A mixed finite element method for second order elliptic problems. *Mathematical aspects of finite element methods*, pages 292–315, 1977.
47. A. F. Sarmiento, A. M. A. Cortes, D. A. Garcia, L. Dalcin, N. Collier, and V. M. Calo. PetIGA-MF: A multi-field high-performance toolbox for structure-preserving B-splines spaces. *Journal of Computational Science*, 18:117–131, 2017.

48. P. W. Schroeder and G. Lube. Pressure-robust analysis of divergence-free and conforming fem for evolutionary incompressible navier–stokes flows. *Journal of Numerical Mathematics*, 2017.
49. F. Shakib, T. J. R Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. the compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89:141–219, 1991.
50. T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.
51. T. M. van Opstal, J. Yan, C. Coley, J. A. Evans, T. Kvamsdal, and Y. Bazilevs. Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 316:859–879, 2017.
52. W. van Rees, A. Leonard, D. Pullin, and P. Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *Journal of Computational Physics*, 230:2794–2805, 2011.
53. S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65:138–158, 1986.
54. P. Vignal, A. Sarmiento, A. M. A. Côrtes, L. Dalcin, and V. M. Calo. Coupling Navier-Stokes and Cahn-Hilliard equations in a two-dimensional annular flow configuration. *Procedia Computer Science*, 51:934–943, 2015.
55. A. W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, 16:3670–3681, 2004.

Interaction of Multiphase Fluids and Solid Structures

Hector Gomez and Jesus Bueno

Abstract Fluid–Structure Interaction (FSI) problems are ubiquitous in almost every branch of engineering and science. Their nonlinear and time-dependent nature makes usually the analytical solution very difficult or even impossible to obtain, requiring the use of experimental analysis and/or numerical simulations. This fact has prompted the development of a great variety of numerical methods for FSI. However, most of the efforts have been focused on classical fluids governed by the Navier–Stokes equations, which cannot capture the physical mechanisms behind multiphase fluids. Here, we present several models for the interplay of solids and multiphase flows, which we apply to particular problems such as phase-change-driven implosion, droplet motion, and elastocapillarity.

In this work, the behavior of the structure is described by the nonlinear equations of elastodynamics and treated as a hyperelastic solid. In particular, we employ a Neo-Hookean and a Saint Venant–Kirchhoff model. Our approach for the multiphase fluid is based on the diffuse-interface or phase-field method. The Navier–Stokes–Korteweg equations are used to describe compressible fluids that are composed of two phases of the same component, which may undergo phase transformation. The Navier–Stokes–Cahn–Hilliard equations are used to describe two-component immiscible flows with surface tension. As FSI technique, we adopt a boundary-fitted approach with matching discretization at the interface. This choice leads to a natural monolithic FSI coupling with strong, exact enforcement of the kinematic conditions. We use the Lagrangian description to derive the semidiscrete form of the solid equations and the Arbitrary Lagrangian–Eulerian description for the fluid domain. For the spatial discretization we adopt isogeometric analysis based on Non-Uniform Rational B-Splines. Regarding the time integration, we use a

H. Gomez (✉)
School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA
e-mail: hectorgomez@purdue.edu

J. Bueno
Departamento de Matemáticas, Universidade da Coruña, A Coruña, Spain
e-mail: jesus.bueno@udc.es

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_4

generalized-α scheme. The nonlinear system of equations is solved using a Newton–Raphson iteration procedure, which leads to a two-stage predictor-multicorrector algorithm. A quasi-direct monolithic formulation is adopted for the solution of the FSI problem.

1 Introduction

Fluid–structure interaction (FSI) encompasses a group of problems in which a deformable or movable solid interacts with an internal or surrounding fluid flow, that is, problems with an interdependence between a fluid and a solid structure. The nature of this kind of problems makes them ubiquitous in almost every branch of engineering and science, such as biomedical research [5], aerospace engineering [33, 81], or marine engineering [90] to name but a few.

In the last decades, significant progress has been made in FSI research. However, a comprehensive study of FSI problems remains a challenge due to their nonlinear and time-dependent nature. In most cases, an analytical solution is very difficult or even impossible to obtain and experimental analysis and/or numerical simulations need to be employed. The same challenges appear in the interaction of multiphase/multicomponent fluids and solid structures. The unusual mechanical nature of this kind of fluids usually produces much richer physics than classical fluids when they interact with solids, especially at small scales. Despite this fact, the computational mechanics community has traditionally focused its efforts on the interaction of solids and classical fluids governed by the Navier–Stokes equations [6, 58, 76, 80, 82], which cannot capture the physical mechanism behind multiphase and multicomponent fluids. The number of works that model the interaction of multiphase fluids and solids is very limited. The first publication we know of is [13]; more recent papers include [12, 14, 88]. We present several models for the interplay of solids and multiphase flows. The development of accurate and efficient computational methods for this kind of FSI problems could potentially benefit a number of disciplines, such as mechanobiology, microfabrication, or engineering:

– **Mechanobiology**. Cells can migrate in response to gradients in stiffness (durotaxis) and gradients in strain (tensotaxis) of the underlying substrate [57, 77]. The understanding of these mechanisms is still very limited but they seem to be controlled to a significant extent by mechanics, and it seems plausible that, at cellular scale, capillary forces play an important role. A theoretical model that allowed to study durotaxis and/or tensotaxis would contribute to a better understanding of these processes, and would allow to explore scenarios that are difficult to probe experimentally.
– **Engineering**. There is an enormous variety of engineering problems in which multiphase fluids play a prominent role. Among them, one of the most relevant problems is the implosive collapse of thin structures [48, 67]. Structures containing a compressible fluid at a pressure below the external pressure have the potential to collapse inwards. For example, an air-filled structure may suffer an implosive collapse when it is immersed underwater due to the increase of

the external pressure. In other situations, phase transformations may trigger the implosion of vapor-filled structures. Thermal variations may transform the vapor into liquid, reducing the internal pressure, which, in turn, may lead to a violent and rapid collapse of the structure, producing strong compressions and large deformations [49].

– **Microfabrication**. The growing interest for micro-devices has brought attention to the effects that surface tension may have in soft or slender structures. As the dimensions of a system are scaled down, capillary forces decrease slower than pressure and gravity forces and eventually become dominant, being able to deform elastic structures. This process is known as elastocapillarity [68, 74]. Understanding the physical mechanisms underlying elastocapillary phenomena is essential for the design of new materials and devices at the micrometer and nanometer scale.

We present different models for the interplay of solid structures and multiphase/multicomponent flows. Our computational method uses a boundary-fitted approach with a sharp fluid–solid interface. However, our approach for the fluid is based on the diffuse-interface or phase-field method. We apply the proposed models to particular problems such as phase-change-driven implosion, droplet motion, and elastocapillarity.

1.1 Phase Transitions and Computational Challenges

Traditionally, phase transition phenomena has been modeled using sharp-interface methods [3, 52]. However, this approach presents several limitations. For many classes of problems, the appropriate sharp interface model is unknown [65] and the numerical simulation turns out to be extremely difficult. The most challenging aspect is the interaction between topologically complex interfaces that undergo merging and pinch-off during the course of a phase transformation. Note that one must solve simultaneously the partial differential equations that govern each phase dynamics and the boundary conditions that hold on a moving, and a priori unknown interface, and this usually results in complex numerical treatments.

An alternative to deal with moving interface problems are phase-field models [2, 25, 41, 60]. The principal characteristic of this approach is the assumption that the interface actually represents a rapid but smooth transition of physical quantities between the bulk values—for this reason, phase-field models are also called diffuse-interface models. The idea of diffuse interfaces dates back to Poisson [64], Gibbs [36], and van der Waals [89]. However, it is in the last decades, when these models are gaining increasing interest in many fields, such as physics, material science, and engineering. They have been applied with great success to solidification dynamics [62], foams [34], brittle fracture [51], dendritic growth [46], vesicle dynamics [9], microstructure evolution in solids [35], planet formation [86], infiltration of water into a porous medium [21, 38], coalescence of bubbles [40], cancer growth [91, 92], elastocapillarity [88], or the description of partial wetting [22] to mention a few applications.

In phase-field models, the sharp description of the interface is replaced with a thin boundary layer by introducing an auxiliary field—the phase field or order parameter—that varies smoothly on the entire computational domain and that acts as marker of the location of the different phases. This idea avoids some of the problems associated with sharp interface models. The numerics are notably simplified because the governing equations can now be solved on the entire computational domain, which is usually known and fixed. In general, instead of the boundary conditions at the interfaces, we have to solve a partial differential equation for the evolution of the order parameter. Phase-field models also present important challenges that are currently being addressed (see [39, 41, 56]) and that make it difficult to use these approaches for quantitative modeling of experimentally relevant situations. For example, the presence of thin interfaces that must be captured by the computational mesh renders phase-field equations very stiff. The same way, the kinetics of the interface limits the numerical time steps that can be adopted in diffuse-interface theories, which makes difficult to model realistic time scales. Finally, from the computational point of view, another disadvantage is that the phase-field equation typically includes higher-order partial differential operators. This fact significantly limits the use of classical finite element methods (FEM) since we need to employ basis functions of high global continuity, which can be very difficult or even impossible in 3D complicated geometries. In order to use FEM it is usually necessary to adopt a mixed formulation adding new degrees of freedom. The aforementioned difficulties can be tackled by using Isogeometric Analysis (IGA) for the spatial discretization of the problem, which permits to generate basis functions with controllable continuity on non-trivial geometries. IGA is a generalization of the finite element method that was introduced in [43]. The main idea behind IGA is to use functions from computational geometry to represent both the solution and the domain of a boundary-value problem. The most frequently utilized functions are Non-Uniform Rational B-Splines (NURBS) and they are also our choice for the simulations presented here. The higher-order global continuity of NURBS leads to a more accurate and stable solution of the thin layers that naturally arise in the solution of phase-field theories [37].

Phase field models are connected to thermodynamics, in the sense that they can be derived from free-energy functionals using the classical theory of thermomechanics and Coleman–Noll approaches [20]. As an example, the classical Cahn–Hilliard equations as well as the Navier–Stokes–Korteweg (NSK) equations can be derived using this framework. The former represents a theory for two-component immiscible fluids with surface tension. The latter is the most widely accepted theory for single-component two-phase flow with phase transformations. For a more thorough discussion on phase-field methods, the reader is referred to [2, 17, 41, 71].

1.2 Fluid–Structure Interaction Technique

When solving FSI problems it is essential to choose the most suitable computational technique for the problem that we want to solve. In this regard, FSI methods may be classified into two main groups: boundary-fitted [7, 8, 78, 79] and immersed approaches [47, 63, 93]. In the former, the shape of the fluid-subdomain changes to adapt to the motion of the structure, that is, the fluid mesh tracks the interface and is updated as the flow evolves. This allows to control the mesh resolution close to the interface but it may result in important mesh distortions. In immersed approaches, the solid moves arbitrarily on top of a background fluid mesh in a non-conforming fashion. The interface is simply "captured" by the fluid mesh covering the region where it is located. This method results in a less accurate representation of the interface than boundary-fitted approaches although it is more flexible since it does not require mesh updating.

In our particular case, the idea is to conceive a model capable of reproducing the implosive collapse of vapor-filled structures as well as droplet motion and elastocapillarity. In all these problems, the accuracy at the fluid–solid interface is crucial, especially, in elastocapillary problems, as we need a rigorous control of the contact angle between the liquid–vapor interface and the solid surface as well as a precise fulfillment of the compatibility conditions. For this reason, a boundary-fitted approach or in particular an Arbitrary Lagrangian–Eulerian method (ALE) [28, 29, 44], in which the fluid and the solid discretizations match at the interface, seems to be the most suitable choice. We are aware that very strong implosion problems can lead to significant distortions in the solid and the fluid mesh and, as a consequence, require remeshing procedures. We try to tackle this challenge by focusing on moderately strong implosions and by adopting the appropriate elastic coefficients for the moving mesh problem.

2 Kinematics and Computational Domain

In Arbitrary Lagrangian–Eulerian methods, the material particles of the fluid are in relative motion with respect to a referential domain. Let us use $\widehat{\boldsymbol{x}}$ to refer to the coordinates of that reference domain $\Omega_{\widehat{\boldsymbol{x}}}$ which is fixed in time. $\Omega_{\widehat{\boldsymbol{x}}}$ is arbitrary and may take on different interpretations. We define a function $\widehat{\boldsymbol{\phi}} : \Omega_{\widehat{\boldsymbol{x}}} \times]0, T[\rightarrow \Omega_t$ that maps the reference domain $\Omega_{\widehat{\boldsymbol{x}}}$ into its spatial configuration at time t, Ω_t, where $]0, T[$ is the time interval of interest. Let us denote $\boldsymbol{x}$ the coordinates in the spatial configuration, i.e., $\Omega_t \ni \boldsymbol{x} = \widehat{\boldsymbol{\phi}}(\widehat{\boldsymbol{x}}, t)$. We also use the notation $\widehat{\boldsymbol{\phi}}_t(\widehat{\boldsymbol{x}}) = \widehat{\boldsymbol{\phi}}(\widehat{\boldsymbol{x}}, t)$. Using the mapping $\widehat{\boldsymbol{\phi}}$, we can define the displacement of a point in the referential domain $\widehat{\boldsymbol{u}}(\widehat{\boldsymbol{x}}, t) = \widehat{\boldsymbol{\phi}}(\widehat{\boldsymbol{x}}, t) - \widehat{\boldsymbol{x}}$, and its velocity

$$\widehat{\boldsymbol{v}} = \frac{\partial \widehat{\boldsymbol{u}}}{\partial t}. \tag{1}$$

Let us also define the mapping $\boldsymbol{\phi} : \Omega_X \times]0, T[\rightarrow \Omega_t$ that transforms each material particle $\boldsymbol{X}$ into its spatial coordinate at time t, that is, $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$. Note that the mapping $\boldsymbol{\phi}$ can be understood as a particular case of $\widehat{\boldsymbol{\phi}}$. In this work, we use different notations since each mapping is employed for different purposes. From the function $\boldsymbol{\phi}$, we can define the deformation gradient $\boldsymbol{F} = \partial\boldsymbol{\phi}/\partial\boldsymbol{X}$, the particle displacement $\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{\phi}(\boldsymbol{X}, t) - \boldsymbol{X}$, and the particle velocity

$$\boldsymbol{v} = \frac{\partial \boldsymbol{\phi}}{\partial t} = \frac{\partial \boldsymbol{u}}{\partial t}. \tag{2}$$

In what follows, to avoid ambiguity in our notation we will use subscripts to clarify how derivatives are to be understood. We adopt the subscript $\boldsymbol{X}$ (respectively, $\widehat{\boldsymbol{x}}$) to indicate that the derivative is taken by holding $\boldsymbol{X}$ (respectively, $\widehat{\boldsymbol{x}}$) fixed. When no subscript is specified, the derivative is assumed to be taken by holding the spatial coordinate $\boldsymbol{x}$ fixed.

Let Ω_0 denote the initial configuration of the entire domain of our problem, that is, the fluid and solid domains combined. Ω_0 will be adopted as the reference configuration and also as the material configuration. We may decompose Ω_0 as $\Omega_0 = \overline{\Omega_0^f \cup \Omega_0^s}$, with $\Omega_0^f \cap \Omega_0^s = \emptyset$, where s and f refer to the solid and the fluid domain, respectively. We may also decompose the spatial configuration of Ω_0 at time t, namely Ω_t, as $\Omega_t = \overline{\Omega_t^f \cup \Omega_t^s}$ with $\Omega_t^f \cap \Omega_t^s = \emptyset$. Let Γ_0 and Γ_t be the fluid–solid interface where the subscripts 0 and t denote the initial and the current configuration, respectively. We will denote by Γ_0^s and Γ_t^s (respectively, Γ_0^f and Γ_t^f) the boundary of the solid (respectively, the fluid) subdomain without the part of the fluid–structure interface. In Fig. 1, we show an illustration with the notation that we have adopted for the reference and the spatial domains.

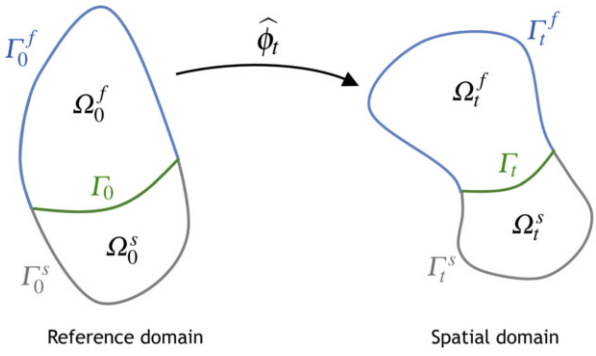


Fig. 1 Reference and spatial domain of our fluid–structure interaction problem. The superscripts "s" and "f" stand for solid and fluid, respectively. The absence of superscript makes reference to the fluid–structure interface. The subscript "0" (respectively "t") denotes the initial (current) configuration

3 Phase-Change-Driven Implosion of Thin Structures

Thin structures that enclose a compressible fluid may collapse inwards when the external pressure is larger than the internal pressure. This may happen, for example, if an air-filled structure is immersed underwater [32, 45, 49, 67, 87]. Also, thermal variations may trigger the collapse of vapor-filled structures. A decrease in the internal temperature can make vapor transform into liquid, reducing the internal pressure and triggering the collapse of the structure. We refer to the latter as phase-change-driven implosion. In order to model this phenomenon we employ the Navier–Stokes–Korteweg theory for the fluid subdomain, that is, a single-component two-phase theory in which the fluid can undergo phase transformations. The structure is a hyperelastic material governed by the nonlinear equations of elastodynamics.

3.1 Governing Equations

3.1.1 Solid Mechanics

The behavior of the solid is described by the Lagrangian form of the momentum balance equation

$$\rho_0^s \left.\frac{\partial^2 \boldsymbol{u}}{\partial t^2}\right|_X = \nabla_X \cdot \boldsymbol{P} + \rho_0^s \boldsymbol{f}^s, \tag{3}$$

where ρ_0^s is the mass density in the initial configuration, $\boldsymbol{f}^s$ represents body forces per unit mass, and $\boldsymbol{P}$ is the first Piola–Kirchhoff stress tensor. To define $\boldsymbol{P}$ we use as constitutive theory a hyperelastic model, namely, the generalized neo-Hookean model with dilatational penalty proposed by Simo and Hughes [69]. The aforementioned material model can be described by the stored elastic energy density

$$W = \frac{\mu^s}{2}\left(J^{-2/d}\mathrm{tr}(\boldsymbol{C}) - d\right) + \frac{\kappa^s}{2}\left(\frac{1}{2}\left(J^2 - 1\right) - \ln J\right), \tag{4}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator and d is the number of spatial dimensions; κ^s is the material bulk modulus and μ^s is the shear modulus, also known as the second Lamé parameter. κ^s and μ^s can be related to the Young modulus E^s and the Poisson ratio ν^s using the relations $\kappa^s = E^s/(3(1-2\nu^s))$ and $\mu^s = E^s/(2(1+\nu^s))$; J is the determinant of the deformation gradient, i.e., $J = \det(\boldsymbol{F})$, where $\boldsymbol{F}$ can be computed as $\boldsymbol{F} = \boldsymbol{I} + \nabla_X \boldsymbol{u}$. Here, $\boldsymbol{I}$ denotes the identity tensor. Finally, $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ is the Cauchy–Green deformation tensor. The second Piola–Kirchhoff stress tensor can be computed from W by taking its derivative with respect to the Green–Lagrange strain tensor $\boldsymbol{E} = (\boldsymbol{C} - \boldsymbol{I})/2$, i.e.,

$$\boldsymbol{S} = \frac{\partial W}{\partial \boldsymbol{E}} = \mu^s J^{-2/d} \left(\boldsymbol{I} - \frac{1}{d} \mathrm{tr}(\boldsymbol{C}) \boldsymbol{C}^{-1} \right) + \frac{\kappa^s}{2} \left(J^2 - 1 \right) \boldsymbol{C}^{-1}. \tag{5}$$

The first Piola–Kirchhoff stress tensor is obtained by taking $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$. The Cauchy stress tensor in the solid is given by $\boldsymbol{\sigma}^s = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T = J^{-1}\boldsymbol{P}\boldsymbol{F}^T$.

3.1.2 Fluid Mechanics

In our model, the behavior of the fluid is governed by the NSK equations, which account for mass, momentum, and energy conservation. The Navier–Stokes–Korteweg theory fits into classical thermodynamics and can be derived from a Helmholtz free-energy potential [41]. The NSK system constitutes the most widely accepted theory for the description of single-component two-phase flows and naturally allows for phase transformations in the fluid, which can happen spontaneously without precursors. In the ALE description, the NSK equations can be written as

$$\left.\frac{\partial \rho}{\partial t}\right|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla \rho + \rho \nabla \cdot \boldsymbol{v} = 0, \tag{6a}$$

$$\rho \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\widehat{x}} + \rho \left(\boldsymbol{v} - \widehat{\boldsymbol{v}}\right) \cdot \nabla \boldsymbol{v} - \nabla \cdot \boldsymbol{\sigma}^f - \rho \boldsymbol{f} = 0, \tag{6b}$$

$$\begin{aligned} &\left.\frac{\partial (\rho s)}{\partial t}\right|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla (\rho s) + \rho s \nabla \cdot \boldsymbol{v} + \nabla \cdot \left(\frac{\boldsymbol{q}}{\theta}\right) \\ &\quad = k \frac{1}{\theta^2} |\nabla \theta|^2 + \frac{1}{\theta} \boldsymbol{\tau} : \nabla \boldsymbol{v} + \frac{\rho r}{\theta} + \frac{\rho \boldsymbol{f} \cdot \boldsymbol{v}}{\theta}, \end{aligned} \tag{6c}$$

where $\otimes$ denotes the usual vector outer product and $\widehat{\boldsymbol{x}}$ is a coordinate in the reference domain (see Sect. 2); ρ is the fluid density, $\boldsymbol{v}$ denotes the fluid velocity vector, and $\widehat{\boldsymbol{v}}$ is the fluid domain velocity; $\boldsymbol{f}$ represents body forces per unit mass, θ is the temperature of the fluid, and r is the heat supply per unit mass; $\boldsymbol{q}$ denotes the heat flux, which is assumed to satisfy the isotropic Fourier law, i.e., $\boldsymbol{q} = -k\nabla\theta$, where k is the thermal conductivity. The Cauchy stress tensor of the fluid $\boldsymbol{\sigma}^f$ is defined as $\boldsymbol{\sigma}^f = \boldsymbol{\tau} - p\boldsymbol{I} + \boldsymbol{\varsigma}$. Here, p is the thermodynamic pressure, $\boldsymbol{\tau}$ is the viscous stress tensor of a Newtonian fluid, and $\boldsymbol{\varsigma}$ denotes the so-called Korteweg tensor that gives rise to capillary forces that are withstood at liquid–vapor interfaces, i.e.,

$$\boldsymbol{\tau} = \bar{\mu} \left(\nabla \boldsymbol{v} + \nabla^T \boldsymbol{v} \right) + \bar{\lambda} \nabla \cdot \boldsymbol{v} \boldsymbol{I}, \tag{7a}$$

$$\boldsymbol{\varsigma} = \lambda \left(\rho \Delta \rho + \frac{1}{2} |\nabla \rho|^2 \right) \boldsymbol{I} - \lambda \nabla \rho \otimes \nabla \rho. \tag{7b}$$

In the previous expressions, $\bar{\mu}$ and $\bar{\lambda}$ are the viscosity coefficients, which are assumed to fulfill the Stokes hypothesis, that is, $\bar{\lambda} = -2\bar{\mu}/3$; $|\cdot|$ represents the Euclidean norm of a vector and $\lambda > 0$ denotes the capillarity coefficient. λ is related to the surface tension at the liquid–vapor interface γ_{LV} through the expression $\gamma_{LV} \simeq K_0\sqrt{\lambda}$ where K_0 is a constant for a given temperature; see [27] for more details. Finally, the thermodynamic pressure p and the entropy density s are the state variables of a van der Waals fluid (see, for example, [27, 55]),

$$p = Rb\left(\frac{\rho\theta}{b-\rho}\right) - a\rho^2, \tag{8a}$$

$$s = -R\log\left(\frac{\rho}{b-\rho}\right) + c_{\boldsymbol{v}}\log\left(\frac{\theta}{\theta_c}\right), \tag{8b}$$

where R is the specific gas constant, and a and b are positive constants; $c_{\boldsymbol{v}}$ is the specific heat capacity and θ_c is the critical temperature $\theta_c = 8ab/(27R)$. Figure 2 shows a non-dimensional plot of van der Waals pressure with respect to density for three different values of temperature. For $\theta > \theta_c$, p is a monotonically increasing function of ρ, and the only phase that is stable is the vapor phase. However, for $\theta < \theta_c$ the pressure is no longer monotone, and we can differentiate three regions: $\rho \in (0, \rho_v)$ where the vapor phase is stable (red line), $\rho \in (\rho_l, b)$ that corresponds to the liquid phase (blue line), and the region $\rho \in (\rho_v, \rho_l)$ that has no physical meaning (grey line). In our theory, the latter interval may be understood as a smooth interface between the two phases that spans over a length scale $\sqrt{\lambda/a}$. For further details on the NSK equations, the reader is referred to [27, 40, 55].

3.2 Numerical Formulation

3.2.1 Continuous Problem in the Weak Form

Solid Mechanics Problem Let us assume that the solid boundary Γ_0^s can be decomposed into Neumann and Dirichlet parts denoted by $(\Gamma_0^s)_N$ and $(\Gamma_0^s)_D$, respectively. We define a trial solution functional space $\mathcal{X}^s = \mathcal{X}^s(\Omega_0^s)$ whose members verify the Dirichlet boundary conditions of the problem, and a weighting function space $\mathcal{Y}^s = \mathcal{Y}^s(\Omega_0^s)$ which is identical to $\mathcal{X}^s$, but verifies homogeneous conditions on $(\Gamma_0^s)_D$. The variational formulation can be stated as follows: Find $\boldsymbol{u} \in \mathcal{X}^s$ such that $\forall \boldsymbol{w}^s \in \mathcal{Y}^s$,

$$B^s(\boldsymbol{w}^s, \boldsymbol{u}) = F^s(\boldsymbol{w}^s), \tag{9}$$

where

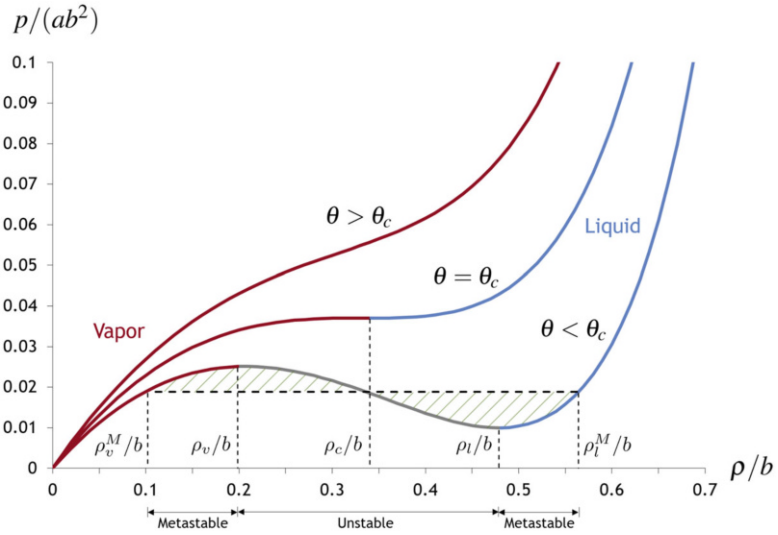


Fig. 2 Non-dimensional plot of van der Waals pressure as a function of density. The pressure is a non-monotone function of density for $\theta < \theta_c$. ρ_v^M and ρ_l^M denote the Maxwell states [27] for the vapor and the liquid phase, respectively

$$B^s(\boldsymbol{w}^s, \boldsymbol{u}) = \int_{\Omega_0^s} \left(\boldsymbol{w}^s \cdot \rho_0^s \left. \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \right|_X + \nabla_X \boldsymbol{w}^s : \boldsymbol{P} \right) \mathrm{d}\Omega_0^s \tag{10}$$

and

$$F^s(\boldsymbol{w}^s) = \int_{\Omega_0^s} \boldsymbol{w}^s \cdot \rho_0^s \, \boldsymbol{f}^s \mathrm{d}\Omega_0^s + \int_{(\Gamma_0^s)_N} \boldsymbol{w}^s \cdot \widehat{\boldsymbol{h}} \, \mathrm{d}(\Gamma_0^s)_N. \tag{11}$$

The previous variational formulation weakly enforces the Neumann boundary condition $\boldsymbol{P}\widehat{\boldsymbol{n}}^s = \widehat{\boldsymbol{h}}$ on $(\Gamma_0^s)_N$, where $\widehat{\boldsymbol{n}}^s$ is the unit outward normal to the solid boundary in the reference domain; $\widehat{\boldsymbol{h}}$ is a given traction, which takes the form of a follower load p in the direction of the inner normal to the solid boundary, i.e., $\widehat{\boldsymbol{h}} = -\mathsf{p} J \boldsymbol{F}^{-T} \widehat{\boldsymbol{n}}^s$.

Fluid Mechanics Problem For the fluid problem, let us assume solid-wall boundary conditions. Additionally, due to the third-order spatial derivative on the density in the linear momentum equations, we enforce the boundary condition $\nabla\rho \cdot \boldsymbol{n}^f = |\nabla\rho| \cos\alpha$ where $\boldsymbol{n}^f$ denotes the unit outward normal to the fluid boundary, and α is the contact angle between the liquid–vapor interface and the solid surface measured in the liquid phase; see Fig. 3. The imposition of the aforementioned boundary condition can be simplified by introducing a new variable $\Upsilon = \Delta\rho$ in the momentum equation. Note that this step is not strictly necessary with isogeometric analysis since IGA allows to use the globally $\mathcal{C}^1$-continuous basis functions that

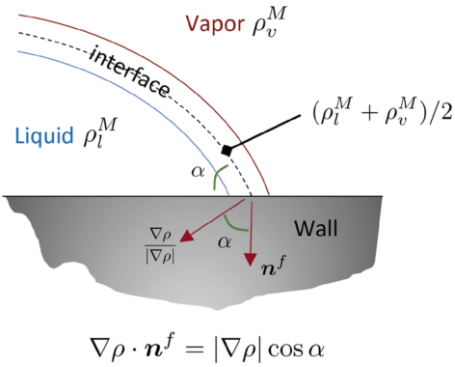


Fig. 3 Contact angle α between the liquid–vapor interface and the solid surface. The interface is assumed to be centered at the mean value of the liquid and vapor densities and spans over a length scale $\sqrt{\lambda/a}$. The boundary condition $\nabla\rho \cdot \boldsymbol{n}^f = |\nabla\rho| \cos\alpha$ imposed at the fluid–structure interface permits to control the contact angle α

are required to approximate the NSK equations in primal form; see [40]. Let us introduce the trial solution space $\mathcal{X}^f = \mathcal{X}^f(\Omega_t^f)$ whose members satisfy all Dirichlet boundary conditions. The weighting functions space $\mathcal{Y}^f = \mathcal{Y}^f(\Omega_t^f)$ is identical to $\mathcal{X}^f$, but all restrictions on the Dirichlet boundary are homogeneous. The variational formulation is stated as follows: Find $\boldsymbol{U} = \{\rho, \boldsymbol{v}, \Upsilon, \theta\} \in \mathcal{X}^f$ such that $\forall \boldsymbol{W} = \{w^1, \boldsymbol{w}^2, w^3, w^4\} \in \mathcal{Y}^f$,

$$B^f(\boldsymbol{W}, \boldsymbol{U}; \widehat{\boldsymbol{v}}) = 0, \tag{12}$$

where

$$\begin{aligned}
B^f(\boldsymbol{W}, \boldsymbol{U}; \widehat{\boldsymbol{v}}) = & \int_{\Omega_t^f} w^1 \left(\left.\frac{\partial \rho}{\partial t}\right|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla\rho + \rho \nabla \cdot \boldsymbol{v} \right) \mathrm{d}\Omega_t^f \\
& + \int_{\Omega_t^f} \boldsymbol{w}^2 \cdot \left(\rho \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\widehat{x}} + \rho\, (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v} - \rho \boldsymbol{f} \right) \mathrm{d}\Omega_t^f \\
& + \int_{\Omega_t^f} \nabla \boldsymbol{w}^2 : (\boldsymbol{\tau} - p\boldsymbol{I})\, \mathrm{d}\Omega_t^f \\
& + \int_{\Omega_t^f} \nabla \cdot \boldsymbol{w}^2 \lambda \left(\rho\Upsilon + \frac{1}{2} |\nabla\rho|^2 \right) \mathrm{d}\Omega_t^f \\
& - \int_{\Omega_t^f} \nabla \boldsymbol{w}^2 : \lambda \nabla\rho \otimes \nabla\rho \mathrm{d}\Omega_t^f \\
& - \int_{\Gamma_t^f \cup \Gamma_t} w^3 |\nabla\rho| \cos\alpha\, \mathrm{d}(\Gamma_t^f \cup \Gamma_t) \\
& + \int_{\Omega_t^f} w^3 \Upsilon \mathrm{d}\Omega_t^f + \int_{\Omega_t^f} \nabla w^3 \cdot \nabla\rho \mathrm{d}\Omega_t^f \\
& - \int_{\Omega_t^f} w^4 \left(\frac{\rho r}{\theta} + \frac{\rho \boldsymbol{f} \cdot \boldsymbol{v}}{\theta} \right) \mathrm{d}\Omega_t^f
\end{aligned}$$

$$+\int_{\Omega_t^f} w^4 \left(\frac{\partial (\rho s)}{\partial t}\bigg|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla (\rho s) \right) \mathrm{d}\Omega_t^f$$
$$-\int_{\Omega_t^f} \nabla w^4 \cdot \frac{\boldsymbol{q}}{\theta} \mathrm{d}\Omega_t^f$$
$$+\int_{\Omega_t^f} w^4 \left(\rho s \nabla \cdot \boldsymbol{v} - \frac{1}{\theta} \boldsymbol{\tau} : \nabla \boldsymbol{v} - k \frac{1}{\theta^2} |\nabla \theta|^2 \right) \mathrm{d}\Omega_t^f . \tag{13}$$

If Dirichlet boundary conditions are not set on the entire boundary for velocity or temperature, the variational formulation (13) weakly imposes the conjugate stress-free condition or vanishing heat flux at the wall, respectively.

Fluid Domain Motion The goal of this problem is to produce a smooth evolution of the fluid domain given the displacement data on the fluid–solid interface. This motion is associated to the mapping $\widehat{\boldsymbol{\phi}}$, and can be understood as a succession of fictitious linear elastic boundary-value problems. Let us define the displacement of the reference domain at time t as $\widehat{\boldsymbol{u}}_t(\widehat{\boldsymbol{x}}) = \widehat{\boldsymbol{u}}(\widehat{\boldsymbol{x}}, t) = \widehat{\boldsymbol{\phi}}_t(\widehat{\boldsymbol{x}}) - \widehat{\boldsymbol{x}}$. To determine $\widehat{\boldsymbol{\phi}}_t$, and thus, the motion of the fluid domain, we use the identity $\widehat{\boldsymbol{\phi}}_t(\widehat{\boldsymbol{x}}) = \widehat{\boldsymbol{\phi}}_{\tilde{t}}(\widehat{\boldsymbol{x}}) + \left(\widehat{\boldsymbol{u}}_t - \widehat{\boldsymbol{u}}_{\tilde{t}}\right)(\widehat{\boldsymbol{x}})$, where $\widehat{\boldsymbol{u}}_t - \widehat{\boldsymbol{u}}_{\tilde{t}}$ is obtained from a linear elastic boundary-value problem. Here, $\tilde{t} < t$ is a time instant close to t. In general, in our problems $\tilde{t}$ is the final configuration of the previous time step.

Let us define $\boldsymbol{u}^m$ as $\boldsymbol{u}^m = \widehat{\boldsymbol{u}}_t - \widehat{\boldsymbol{u}}_{\tilde{t}}$. $\boldsymbol{u}^m$ is obtained by solving fictitious linear elastic boundary-value problems, subject to the Dirichlet boundary conditions $\boldsymbol{u}^m = \boldsymbol{u}_t \circ \boldsymbol{\phi}_t^{-1} - \boldsymbol{u}_{\tilde{t}} \circ \boldsymbol{\phi}_{\tilde{t}}^{-1}$ on $\Gamma_{\tilde{t}}$, where $\boldsymbol{u}_t$ and $\boldsymbol{u}_{\tilde{t}}$ are the particle displacement at times t and $\tilde{t}$, respectively. These Dirichlet boundary conditions are strongly built into the trial solution space $\mathcal{V}^m = \mathcal{V}^m(\Omega_{\tilde{t}}^f)$. The corresponding weighting function space—which satisfies homogeneous conditions on the boundary—is denoted by $\mathcal{W}^m = \mathcal{W}^m(\Omega_{\tilde{t}}^f)$. The variational formulation can be stated as follows: Find $\boldsymbol{u}^m \in \mathcal{V}^m$ such that $\forall \boldsymbol{w}^m \in \mathcal{W}^m$,

$$B^m(\boldsymbol{w}^m, \boldsymbol{u}^m) = 0, \tag{14}$$

where

$$B^m(\boldsymbol{w}^m, \boldsymbol{u}^m) = \int_{\Omega_{\tilde{t}}^f} \nabla_{\tilde{x}}^{sym} \boldsymbol{w}^m : 2\mu^m \nabla_{\tilde{x}}^{sym} \boldsymbol{u}^m \mathrm{d}\Omega_{\tilde{t}}^f + \int_{\Omega_{\tilde{t}}^f} \nabla_{\tilde{x}} \cdot \boldsymbol{w}^m \lambda^m \nabla_{\tilde{x}} \cdot \boldsymbol{u}^m \mathrm{d}\Omega_{\tilde{t}}^f . \tag{15}$$

Here, μ^m and λ^m denote the Lamé parameters of a fictitious elastic problem; $\nabla_{\tilde{x}}$ denotes the gradient operator on $\Omega_{\tilde{t}}$ and $\nabla_{\tilde{x}}^{sym}$ is its symmetric part, that is, $\nabla_{\tilde{x}}^{sym} = 1/2 \left(\nabla_{\tilde{x}} + \nabla_{\tilde{x}}^T\right)$.

Coupled FSI Problem Now we have all the ingredients to define the coupled fluid–structure interaction problem. The variational formulation of the problem may be stated as follows: Find $\boldsymbol{U} = \{\rho, \boldsymbol{v}, \Upsilon, \theta\} \in \mathcal{X}^f$, $\boldsymbol{u} \in \mathcal{X}^s$ and $\boldsymbol{u}^m \in \mathcal{V}^m$ such that

$\forall \boldsymbol{W} = \{w^1, \boldsymbol{w}^2, w^3, w^4\} \in \mathcal{Y}^f$, $\forall \boldsymbol{w}^s \in \mathcal{Y}^s$ and $\forall \boldsymbol{w}^m \in \mathcal{W}^m$,

$$B^f(\boldsymbol{W}, \boldsymbol{U}; \widehat{\boldsymbol{v}}) + B^s(\boldsymbol{w}^s, \boldsymbol{u}) + B^m(\boldsymbol{w}^m, \boldsymbol{u}^m) = F^s(\boldsymbol{w}^s), \tag{16}$$

with the following constraints at the fluid–structure interface:

$$\boldsymbol{v} = \frac{\partial \boldsymbol{u}}{\partial t} \circ \widehat{\boldsymbol{\phi}}^{-1} \quad \text{on} \quad \Gamma_t, \tag{17}$$

$$\boldsymbol{w}^2 = \boldsymbol{w}^s \circ \widehat{\boldsymbol{\phi}}^{-1} \quad \text{on} \quad \Gamma_t. \tag{18}$$

The purpose of Eq. (17) is to enforce the compatibility of kinematics at the fluid–solid interface. Equation (18), however, leads to a weak enforcement of traction compatibility at the interface, that is, $\boldsymbol{\sigma}^f \boldsymbol{n}^f + \boldsymbol{\sigma}^s \boldsymbol{n}^s = 0$ on Γ_t. Here, $\boldsymbol{n}^s$ denotes the unit outward normal to the solid in the spatial configuration.

3.2.2 Semidiscrete Formulation

We use NURBS-based isogeometric analysis for the spatial discretization of the FSI problem. We define finite-dimensional approximations of the functional spaces, i.e, $\mathcal{X}_h^f$, $\mathcal{X}_h^s$ and $\mathcal{V}_h^m$ such that $\mathcal{X}_h^f \subset \mathcal{X}^f$, $\mathcal{X}_h^s \subset \mathcal{X}^s$, and $\mathcal{V}_h^m \subset \mathcal{V}^m$. We also define $\mathcal{Y}_h^f \subset \mathcal{Y}^f$, $\mathcal{Y}_h^s \subset \mathcal{Y}^s$, and $\mathcal{W}_h^m \subset \mathcal{W}^m$. The coupled FSI problem (16) can be approximated by the following variational problem over the finite element spaces: Find $\boldsymbol{U}_h = \{\rho_h, \boldsymbol{v}_h, \Upsilon_h, \theta_h\} \in \mathcal{X}_h^f$, $\boldsymbol{u}_h \in \mathcal{X}_h^s$ and $\boldsymbol{u}_h^m \in \mathcal{V}_h^m$ such that $\forall \boldsymbol{W}_h = \{w_h^1, \boldsymbol{w}_h^2, w_h^3, w_h^4\} \in \mathcal{Y}_h^f$, $\forall \boldsymbol{w}_h^s \in \mathcal{Y}_h^s$ and $\forall \boldsymbol{w}_h^m \in \mathcal{W}_h^m$,

$$B^f(\boldsymbol{W}_h, \boldsymbol{U}_h; \widehat{\boldsymbol{v}}_h) + B^s(\boldsymbol{w}_h^s, \boldsymbol{u}_h) + B^m(\boldsymbol{w}_h^m, \boldsymbol{u}_h^m) = F^s(\boldsymbol{w}_h^s), \tag{19}$$

where

$$\rho_h(\boldsymbol{x}, t) = \sum_{A \in I_f} \rho_A(t) N_A(\boldsymbol{x}, t), \qquad w_h^1(\boldsymbol{x}, t) = \sum_{A \in I_f} w_A^1 N_A(\boldsymbol{x}, t), \tag{20a}$$

$$\boldsymbol{u}_h(\boldsymbol{X}, t) = \sum_{A \in I_s} \boldsymbol{u}_A(t) \widehat{N}_A(\boldsymbol{X}), \qquad \boldsymbol{w}_h^s(\boldsymbol{X}) = \sum_{A \in I_s} \boldsymbol{w}_A^s \widehat{N}_A(\boldsymbol{X}), \tag{20b}$$

$$\boldsymbol{u}_h^m(\tilde{\boldsymbol{x}}, \tilde{t}) = \sum_{A \in I_f} \widehat{\boldsymbol{u}}_A(\tilde{t}) \tilde{N}_A(\tilde{\boldsymbol{x}}, \tilde{t}), \qquad \boldsymbol{w}_h^m(\tilde{\boldsymbol{x}}, \tilde{t}) = \sum_{A \in I_f} \boldsymbol{w}_A^m \tilde{N}_A(\tilde{\boldsymbol{x}}, \tilde{t}), \tag{20c}$$

$$\widehat{\boldsymbol{v}}_h(\boldsymbol{x}, t) = \sum_{A \in I_f} \frac{\partial \widehat{\boldsymbol{u}}_A}{\partial t}(t) N_A(\boldsymbol{x}, t). \tag{20d}$$

The rest of the variables ($\boldsymbol{v}_h$, θ_h and Υ_h) and the weighting functions ($\boldsymbol{w}_h^2$, w_h^3 and w_h^4) are defined analogously to ρ_h and w_h^1, respectively. In Eq. (20), the $\widehat{N}_A$'s are a

set of basis functions defined on Ω_0 that are fixed in time. In the context of IGA, these functions are splines with controllable continuity. I_s denotes the global-index set of the $\widehat{N}_A$'s that correspond to the solid domain. N_A is the push forward of $\widehat{N}_A$ to the spatial domain Ω_t, i.e., $N_A(\boldsymbol{x}, t) = \widehat{N}_A \circ \widehat{\boldsymbol{\phi}^h}^{-1}(\boldsymbol{x}, t)$, where $\widehat{\boldsymbol{\phi}^h}^{-1}$ is the discrete counterpart of $\widehat{\boldsymbol{\phi}}^{-1}$. Likewise, $\tilde{N}_A$ is the push forward of $\widehat{N}_A$ to the spatial domain at time $\tilde{t}$, i.e., $\tilde{N}_A(\tilde{\boldsymbol{x}}, \tilde{t}) = \widehat{N}_A \circ \widehat{\boldsymbol{\phi}^h}^{-1}(\tilde{\boldsymbol{x}}, \tilde{t})$. I_f is the global-index set of the N_A's that correspond to the fluid domain.

We enforce the compatibility conditions at the fluid–structure interface by using a unique set of trial and test functions for the velocity at the fluid–structure interface. This leads to strong (pointwise) satisfaction of the kinematics condition and weak satisfaction of the traction compatibility constraint.

3.2.3 Time Discretization

We use the generalized-α method [19] as a time integration scheme. The nonlinear system of equations is solved using a Newton–Raphson iteration procedure, which leads to a two-stage predictor-multicorrector algorithm. The resulting linear system is solved using a preconditioned GMRES method. To solve the FSI equations, we adopt a quasi-direct solution strategy (see [85]), where the fluid and solid equations are solved in a coupled fashion. The mesh motion problem is solved separately, using input data from the fluid–solid solve.

Let $\boldsymbol{U}$, $\dot{\boldsymbol{U}}$, $\ddot{\boldsymbol{U}}$ denote the vectors of control variable degrees of freedom of the fluid–structure system, and its first and second time derivatives, respectively. Let $\boldsymbol{V}$, $\dot{\boldsymbol{V}}$, $\ddot{\boldsymbol{V}}$ denote the vectors of control variable degrees of freedom of mesh displacements, velocities, and accelerations. We define the residual vectors

$$\boldsymbol{R}^{cont} = \{R_A^{cont}\}, \tag{21a}$$

$$\boldsymbol{R}^{mom} = \{R_{A,i}^{mom}\}, \tag{21b}$$

$$\boldsymbol{R}^{aux} = \{R_A^{aux}\}, \tag{21c}$$

$$\boldsymbol{R}^{ener} = \{R_A^{ener}\}, \tag{21d}$$

$$\boldsymbol{R}^{mesh} = \{R_{A,i}^{mesh}\}, \tag{21e}$$

where A is a control-variable index and i is an index associated to the spatial dimensions. The residual components are defined as

$$R_A^{cont} = B^f(\{N_A, 0, 0, 0\}, \{\rho_h, \boldsymbol{v}_h, \Upsilon_h, \theta_h\}; \widehat{\boldsymbol{v}}_h), \tag{22a}$$

$$\begin{aligned} R_{A,i}^{mom} &= B^f(\{0, N_A \boldsymbol{e}_i, 0, 0\}, \{\rho_h, \boldsymbol{v}_h, \Upsilon_h, \theta_h\}; \widehat{\boldsymbol{v}}_h) \\ &\quad + B^s(\widehat{N}_A \boldsymbol{e}_i, \boldsymbol{u}_h) - F^s(\widehat{N}_A \boldsymbol{e}_i), \end{aligned} \tag{22b}$$

$$R_A^{aux} = B^f(\{0, 0, N_A, 0\}, \{\rho_h, \boldsymbol{v}_h, \Upsilon_h, \theta_h\}; \widehat{\boldsymbol{v}}_h), \tag{22c}$$

$$R_A^{ener} = B^f(\{0,0,0,N_A\},\{\rho_h,\boldsymbol{v}_h,\Upsilon_h,\theta_h\};\widehat{\boldsymbol{v}}_h), \tag{22d}$$

$$R_{A,i}^{mesh} = B^m(\tilde{N}_A\boldsymbol{e}_i,\boldsymbol{u}_h^m), \tag{22e}$$

where $\boldsymbol{e}_i$ is the ith cartesian basis vector. The time stepping scheme can be stated as follows: Given the discrete approximation to the global vectors of control variables at time t_n, namely, $\boldsymbol{U}_n, \dot{\boldsymbol{U}}_n, \ddot{\boldsymbol{U}}_n, \boldsymbol{V}_n, \dot{\boldsymbol{V}}_n, \ddot{\boldsymbol{V}}_n$ and the current time step $\Delta t = t_{n+1} - t_n$ find $\boldsymbol{U}_{n+1}, \dot{\boldsymbol{U}}_{n+1}, \ddot{\boldsymbol{U}}_{n+1}, \boldsymbol{V}_{n+1}, \dot{\boldsymbol{V}}_{n+1}, \ddot{\boldsymbol{V}}_{n+1}$ such that

$$\begin{aligned}
&\boldsymbol{R}^{cont}(\boldsymbol{U}_{n+\alpha_f},\dot{\boldsymbol{U}}_{n+\alpha_f},\ddot{\boldsymbol{U}}_{n+\alpha_m},\boldsymbol{V}_{n+\alpha_f},\dot{\boldsymbol{V}}_{n+\alpha_f},\ddot{\boldsymbol{V}}_{n+\alpha_m}) = 0,\\
&\boldsymbol{R}^{mom}(\boldsymbol{U}_{n+\alpha_f},\dot{\boldsymbol{U}}_{n+\alpha_f},\ddot{\boldsymbol{U}}_{n+\alpha_m},\boldsymbol{V}_{n+\alpha_f},\dot{\boldsymbol{V}}_{n+\alpha_f},\ddot{\boldsymbol{V}}_{n+\alpha_m}) = 0,\\
&\boldsymbol{R}^{aux}(\boldsymbol{U}_{n+\alpha_f},\dot{\boldsymbol{U}}_{n+\alpha_f},\ddot{\boldsymbol{U}}_{n+\alpha_m},\boldsymbol{V}_{n+\alpha_f},\dot{\boldsymbol{V}}_{n+\alpha_f},\ddot{\boldsymbol{V}}_{n+\alpha_m}) = 0,\\
&\boldsymbol{R}^{ener}(\boldsymbol{U}_{n+\alpha_f},\dot{\boldsymbol{U}}_{n+\alpha_f},\ddot{\boldsymbol{U}}_{n+\alpha_m},\boldsymbol{V}_{n+\alpha_f},\dot{\boldsymbol{V}}_{n+\alpha_f},\ddot{\boldsymbol{V}}_{n+\alpha_m}) = 0,\\
&\boldsymbol{R}^{mesh}(\boldsymbol{U}_{n+\alpha_f},\dot{\boldsymbol{U}}_{n+\alpha_f},\ddot{\boldsymbol{U}}_{n+\alpha_m},\boldsymbol{V}_{n+\alpha_f},\dot{\boldsymbol{V}}_{n+\alpha_f},\ddot{\boldsymbol{V}}_{n+\alpha_m}) = 0,
\end{aligned} \tag{23}$$

$$\begin{aligned}
\dot{\boldsymbol{U}}_{n+1} &= \dot{\boldsymbol{U}}_n + \Delta t((1-\gamma)\ddot{\boldsymbol{U}}_n + \gamma\ddot{\boldsymbol{U}}_{n+1}),\\
\boldsymbol{U}_{n+1} &= \boldsymbol{U}_n + \Delta t\dot{\boldsymbol{U}}_n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\boldsymbol{U}}_n + 2\beta\ddot{\boldsymbol{U}}_{n+1}\right),\\
\dot{\boldsymbol{V}}_{n+1} &= \dot{\boldsymbol{V}}_n + \Delta t((1-\gamma)\ddot{\boldsymbol{V}}_n + \gamma\ddot{\boldsymbol{V}}_{n+1}),\\
\boldsymbol{V}_{n+1} &= \boldsymbol{V}_n + \Delta t\dot{\boldsymbol{V}}_n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\boldsymbol{V}}_n + 2\beta\ddot{\boldsymbol{V}}_{n+1}\right),
\end{aligned} \tag{24}$$

where

$$\boldsymbol{U}_{n+\alpha_f} = \boldsymbol{U}_n + \alpha_f(\boldsymbol{U}_{n+1} - \boldsymbol{U}_n), \tag{25a}$$

$$\boldsymbol{U}_{n+\alpha_m} = \boldsymbol{U}_n + \alpha_m(\boldsymbol{U}_{n+1} - \boldsymbol{U}_n), \tag{25b}$$

We choose the parameters α_f, α_m, γ, and β according to [4]. The nonlinear system of equations (23) may be solved using a Newton–Raphson iteration procedure; see [13] for more details.

3.3 Numerical Examples

The solution of very strong implosions at full scale is extremely difficult for two main reasons. On the one hand, strong implosions usually result in important fluid mesh distorsions that may require remeshing procedures. On the other hand, very strong implosions are generally associated to strong temperature reductions in the fluid. However, the van der Waals equation is significantly less accurate for low temperatures than for high temperatures. For these reasons, the numerical examples that we are showing in this work only focus on moderately strong implosions.

The problems that we are presenting here were solved in a non-dimensional form. We rescaled the units of measurement of length, mass, time, and temperature by L_0, bL_0^3, $L_0/\sqrt{ab}$, and θ_c, respectively, where $L_0 = 1$ denotes a length scale of the computational domain size. Using this non-dimensionalization, our FSI problem can be rewritten using seven dimensionless numbers: $\widehat{\mu} = \bar{\mu}/(L_0 b\sqrt{ab})$, $\widehat{\gamma} = (\sqrt{\lambda/a})/L_0$, $Pr = L_0 ab^2\sqrt{ab}/(\theta_c k)$, $c = 8c_v/(27R)$, $\widehat{E} = (E^s/\rho_0^s ab)$, $\widehat{\boldsymbol{f}}^s = \boldsymbol{f}^s/(ab/L_0)$ and the Poisson ratio ν^s. The dimensionless numbers $\widehat{\gamma}$ and $\widehat{\mu}$ were chosen according to the methodology proposed in [40], which relates these parameter to the mesh size. In particular, we take

$$\widehat{\gamma} = h/L_0 \quad \text{and} \quad \widehat{\mu} = h/(\eta L_0), \tag{26}$$

where h is a characteristic length scale of the spatial mesh and η is an $O(1)$ constant.

3.3.1 Implosion of a Vapor-Filled Ring

The first numerical example of phase-change-driven implosion is the collapse of a vapor-filled ring; see Fig. 4. The circular structure has the external radius $L_0 = 1$, and the thickness $L_0/40$. The system is initially in thermal and mechanical equilibrium since we impose along the external surface a follower load of identical value than the initial vapor pressure. We trigger the implosion by imposing a low temperature [θ_D in Fig. 4a] at the fluid–solid interface using the temperature boundary condition. The temperature reduction makes the vapor condensate, reducing the internal pressure and increasing the density.

Figure 4 shows the time evolution of the density (middle column) and temperature (right column) in the fluid subdomain. Initially, all the fluid is in gaseous state (red color in the density plots). The thin grey color that encloses the fluid represents the structure. The initial value of the fluid density is the one corresponding to the Maxwell state at $\theta_0 = 0.85$. On the fluid–structure interface we apply the temperature boundary condition $\theta_D = 0.5 < \theta_0$. As a consequence, the temperature of the gas close to the structure is reduced and the vapor turns into liquid (blue); see Fig. 4d. The pressure decreases inside and the structure is deformed due to the external pressure. The process continues so that all the gas is transformed into liquid and the structure is completely collapsed.

3.3.2 Three-Dimensional Implosion of Thin Structures

The second numerical example is the collapse of a three-dimensional box of side $2L_0$ and thickness $L_0/40$; see Fig. 5. We adopt the same parameters and initial conditions than in the two-dimensional case, however the collapse is stronger because the structure is geometrically softer. Figure 5 shows the evolution of the density (left column) and temperature (middle column). We can observe that the

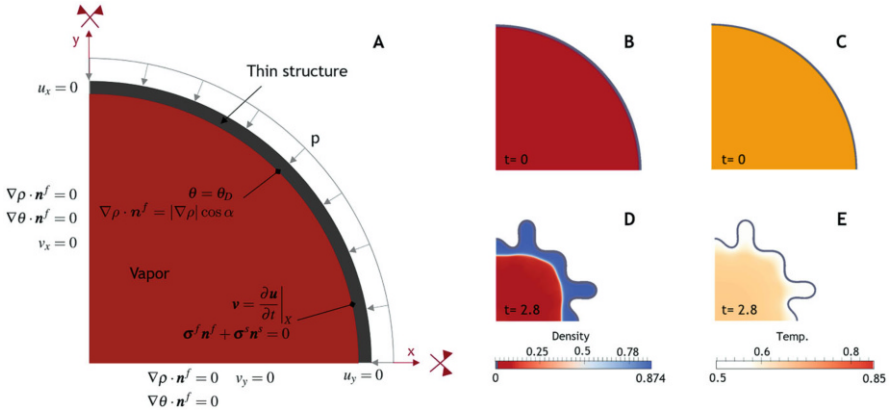


Fig. 4 Two-dimensional implosion of a vapor-filled ring. (**a**) Computational domain. The symmetry of the problem permits to compute a quarter of the circular domain. The symmetry conditions for $\boldsymbol{v}$, ρ, θ, and $\boldsymbol{u}$ are indicated on the left and bottom boundaries. The external pressure p is modeled as a follower load acting on the structure. We impose the temperature θ_D and the contact angle α along the fluid–structure interface. (**b**) and (**c**) Initial configuration for the density and the temperature variables. (**b**) Initially, the structure (grey) is filled with water-vapor (red). (**c**) A low temperature $\theta_D = 0.5$ (white) is applied at the fluid–structure interface. (**d**) and (**e**) plot of density and temperature at time $t = 2.8$. The vapor close to the boundaries reduces its temperature and increases its density, turning into liquid (blue). The pressure inside is reduced and the structure deforms due to the external load. We have used the parameters $\widehat{\mu} = 1/256$, $\widehat{\gamma} = 0.0078125$, $Pr = 0.013$, $c = 0.73$, $\widehat{E} = 2.25$, $\nu^s = 0.125$

temperature boundary condition triggers the transformation of vapor into liquid creating a blue area in the density plot that represents liquid water. Panel (e) in Fig. 5 presents the streamlines of the fluid velocity. As it was expected, the velocity magnitude is higher close to the solid structure and reduces as we converge to the central point of the computational domain.

4 Droplet Motion: Tensotaxis

One of the fields that could benefit the most from a computational model for the interaction of multicomponent fluids and solid structures is droplet dynamics. A model that allows a better understanding in this area could have important implications in a wide range of disciplines, such as the design of new materials and devices at small scales [26, 66, 83], droplet-based microfluidic platforms [1, 31, 70], or geologic carbon sequestration [18] to name but a few.

Significant efforts have been made in the last decade to understand droplet motion in deformable substrates. Recently, Style et al. [75] showed that droplets could undergo durotaxis, that is, motion due to gradients in the stiffness of the underlying substrate. The experiments conducted by Style revealed that wetting

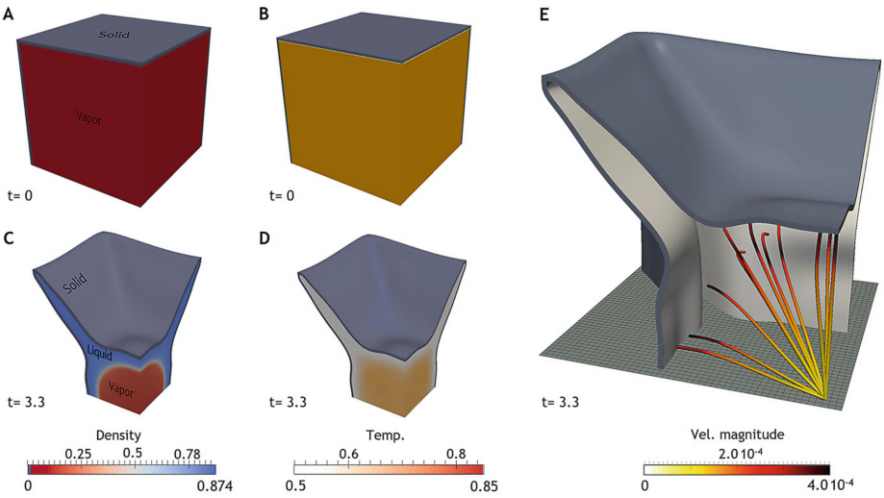


Fig. 5 Evolution of density (left column) and temperature (middle column) in a 3D phase-change-driven implosion. Initially, water-vapor (red) is filling up a hollow structure (**a**) and a low temperature $\theta_D = 0.5$ is applied at the fluid–structure interface (**b**). The vapor close to the boundaries increases its density and turns into liquid (blue) (**c**), while the temperature decreases close to the solid walls (**d**). The pressure inside is reduced and the structure deforms due to the external load. (**e**) shows the streamlines of the fluid velocity colored with the velocity magnitude. We have used the parameters $\widehat{\mu} = 1/256$, $\widehat{\gamma} = 0.0078125$, $Pr = 0.013$, $c = 0.73$, $\widehat{E} = 2.25$, $\nu^s = 0.125$

droplets deposited on substrates with non-uniform stiffness move toward the softer parts of the substrate, in contrast to the behavior previously observed in cells [57]. This finding has revived the debate about the relevance of mechanics in cell migration. It seems plausible that at cellular scale capillary forces play an important role. A theoretical model that allows studying droplet motion on soft substrates would contribute to a better understanding of the process and would permit to explore scenarios that are difficult to probe experimentally.

Here, we focus on the case of tensotaxis, that is, motion along strain gradients. This mechanism is still not well understood although new models have been proposed in recent years [11, 61]. The main reason is that experimental studies of this mechanism are particularly challenging because prestraining the substrate is often followed by an increase in the substrate stiffness due to the nonlinear material response [54], thus producing a combination of tensotaxis and durotaxis. We develop a model that allows to isolate the effects of tensotaxis and we carry out several numerical experiments, which show that liquid droplets on soft substrates move toward areas of higher compressive strains in agreement with the behavior observed in cells.

The static configuration that droplets adopt on infinitely rigid and flat substrates is governed by the Young–Dupré equation [24], $\gamma_{SL} + \gamma_{LV}\cos\alpha = \gamma_{SV}$. Here, α is the static contact angle, and γ_{LV}, γ_{SV}, and γ_{SL} denote the interfacial tension at the liquid–vapor, solid–vapor, and solid–liquid interfaces, respectively; see Fig. 6a.

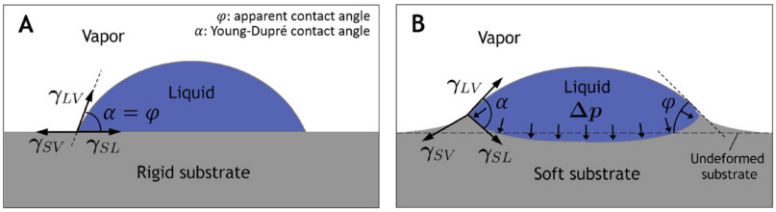


Fig. 6 Wetting on rigid and deformable substrates at small scales. (**a**) Liquid droplet (blue) deposited on a rigid substrate (gray). The shape of the droplet and thus, the static contact angle α, is given by the Young–Dupré equation. (**b**) Liquid droplet on a soft substrate (gray). The Laplace pressure Δp dimples the solid below the droplet and the surface tension γ_{LV} produces a ridge at the contact line. The apparent contact angle φ differs from the contact angle α predicted by the Young–Dupré equation

When the substrate is soft, the Laplace pressure inside the liquid may create a dimple under the droplet and the elastocapillary forces may produce a ridge at the contact line. There is a rotation of the liquid–vapor interface [72] and the angle formed by the liquid–vapor interface and the undeformed surface of the substrate (apparent contact angle φ) differs from α; see Fig. 6b. Elastocapillary forces are relevant when the elastocapillary length $l_{ec} = \gamma_{LV}/E^s$ (E^s is the Young modulus of the solid) is comparable to the droplet radius. For values of l_{ec} much smaller than the droplet radius, the deformation of the solid is negligible.

We hypothesize that, for non-zero Poisson ratio and in absence of gravity, forces applied on the plane of the substrate may cause a strain gradient, which in turn, has an impact on the apparent contact angle causing an imbalance of forces that may trigger the motion of the droplet towards the compressed part of the solid. To verify our hypothesis, we have adapted the model presented in Sect. 3. Most models for the interaction of droplets and deformable substrates that have been presented so far are based on thin film descriptions of the fluid problem and linear elastic solids [50, 73]. The approaches are variational and allow computing a minimum-energy configuration. We use a three-dimensional model that couples the nonlinear dynamics of a solid with a fluid composed of a liquid and a gaseous phase. Using this computational approach, we mimic the cell locomotion experiments reported in [57]. A microneedle is inserted into the substrate and exerts a force either toward the droplet or away from the droplet.

4.1 Governing Equations

In order to avoid the combination of tensotaxis and durotaxis, we propose an FSI model in which the solid is defined by a Saint Venant–Kirchhoff theory. This choice allows to consider geometric nonlinearities with a linear material response. The fluid is governed by the isothermal Navier–Stokes–Korteweg equations (see Sect. 3.1) allowing for the stable coexistence of a liquid and a gaseous phase.

4.1.1 Solid Mechanics Equations

The behavior of the solid is described by the Lagrangian form of the momentum balance equation; see Eq. (3). In this case, to define the first Piola–Kirchhoff stress tensor $\boldsymbol{P}$, we adopt a Saint Venant–Kirchhoff constitutive theory [7, 59]. The stored elastic energy density takes the form

$$W = \frac{\lambda^s}{2} (\mathrm{tr}(\boldsymbol{E}))^2 + \mu^s \, \mathrm{tr}(\boldsymbol{E}^2). \tag{27}$$

Here, λ^s and μ^s are the first and second Lamé parameters, which can be written as a function of the Young modulus E^s and the Poisson ratio ν^s: $\lambda^s = \nu^s E^s/((1 + \nu^s)(1 - 2\nu^s))$ and $\mu^s = E^s/(2(1 + \nu^s))$.

4.2 Numerical Formulation

The computational approach adopted for this problem is identical to that used for the implosion example; see Sect. 3.2. The reader is referred to [12] for more details.

4.3 Numerical Examples

The results presented in this section are in non-dimensional form. Our FSI problem can be defined using six dimensionless numbers: a dimensionless viscosity $\widehat{\mu} = \bar{\mu}/(L_0 b\sqrt{ab})$, a dimensionless surface tension $\widehat{\gamma} = (\sqrt{\lambda/a})/L_0$, a dimensionless temperature $\widehat{\theta} = \theta/\theta_c$, a dimensionless Young modulus $\widehat{E} = (E^s/\rho_0^s ab)$, a dimensionless body force $\widehat{\boldsymbol{f}}^s = \boldsymbol{f}^s/(ab/L_0)$, and the Poisson ratio ν^s.

4.3.1 Unraveling Droplet Tensotaxis

The first numerical example that we carried out mimics the experiments conducted for cells [57]; see Fig. 7. The left panels in Fig. 7 correspond to the case in which

Fig. 7 (continued) moves in the direction of the applied force. The dashed black line represents the position of the droplet at the initial time. (**e**) and (**f**) Time evolution of the apparent contact angles at the left (blue line) and right (red line) contact lines of the droplet. The difference in apparent contact angles between the two contact lines is responsible for the motion of the droplet. (**g**) and (**h**) Droplet velocity with respect to time for different Poisson ratios ν^s. (**i**) and (**j**) Droplet velocity with respect to the position of the droplet center X_c. In all cases the velocity is reduced as the Poisson ratio drops, and for $\nu^s = 0$ the applied force induces no droplet motion. The computational domain is the rectangle $\Omega = [0, 1.0] \times [0, 0.5]$, which is discretized with a uniform mesh of 128×64 $\mathcal{C}^1$-quadratic elements. The static contact angle is $\alpha = 75°$. We have used the parameters $\nu^s = 0.45$, $\widehat{\mu} = 1/256$, $\widehat{\gamma} = 1/64$, $\widehat{E} = 0.7554$, $\widehat{\theta} = 0.39$, and $\left|\widehat{\boldsymbol{f}}^s\right| = 0.16215$

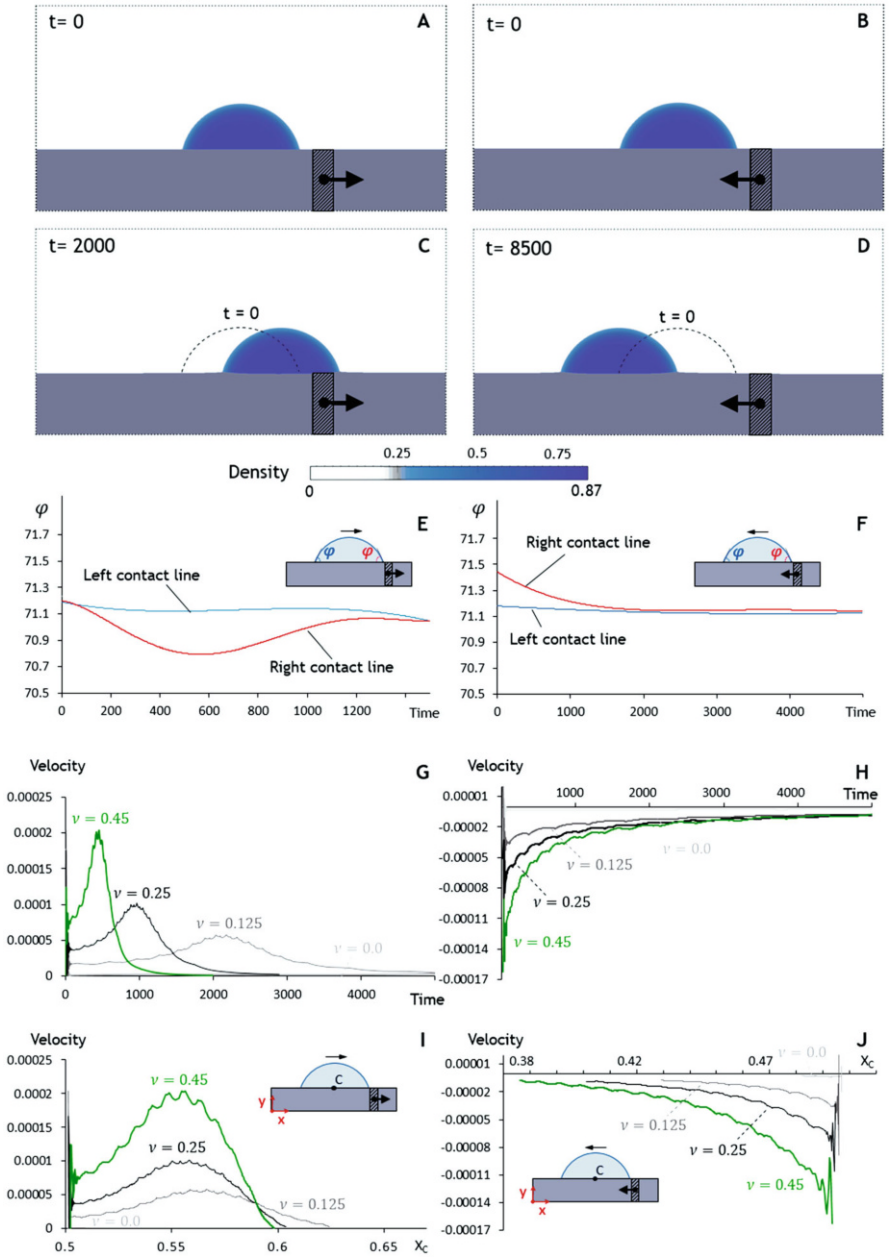


Fig. 7 Mechanism of tensotaxis. Left panels refer to the case in which the applied force pulls the droplet. In the right panels the applied force pushes the droplet. (**a**) and (**b**) Initial configuration of a droplet on a deformable substrate. We apply a force per unit mass that points away from the droplet (**a**) and toward the droplet (**b**) in the marked rectangular region. (**c**) and (**d**) The droplet

we pull the droplet. The right panels present the results when we push the droplet. The effect of the needle is modeled as a horizontal force per unit mass applied on a localized area; see striped area in panels (a) and (b). Our results show that droplets move in the direction of the applied force, in agreement with the behavior observed in cells; see panels (c) and (d) in which the black dashed line represents the initial position of the droplets. The applied load produces vertical displacements in the solid, as it would be expected in a material with a nonzero Poisson ratio. The vertical displacements are positive where the load induces compressive stresses, and negative on the side where the load induces tensile stresses. As a consequence, there is a rotation of the contact lines, that is, the solid deformation produces different apparent contact angles at each of the contact lines [panels (e) and (f)], leading to an imbalance of horizontal forces. The tensotaxis mechanism is independent of gravity, which is neglected in our simulations.

From the velocity plots in Fig. 7 we can see the different behavior of the droplets depending on the direction of the applied force. Pulling produces increasing droplet velocities as the contact line approaches the region where the external force is applied. However, at certain point the velocity decreases and the droplet gets trapped in the loaded area; see panels (g) and (i). On the other hand, pushing repels the droplet monotonically with a time-decreasing velocity magnitude; see panels (h) and (j). We have repeated the same computations for different Poisson ratios; see grey curves in panels (g)–(j). The results show that the maximum droplet velocity increases with the Poisson ratio ν^s. When $\nu^s = 0$ the droplet remains immobile, which is consistent with the proposed mechanism.

4.3.2 Three-Dimensional Tensotaxis

We have also performed three-dimensional simulations of tensotaxis in order to obtain a more faithful reproduction of the reality. In the left column of Fig. 8, we present the 3D analogue of the needle experiment. The external load applied at the center of the substrate induces droplet motion in the direction of the force; see panel (b). Panel (c) shows the vertical displacements induced by the external load, the surface tension at the liquid–vapor interface and the Laplace pressure inside the droplet. The vertical displacements are positive where the load induces compressive stresses. Negative displacements occur under the droplet and on the side where the load induces tensile stresses. In the right column of Fig. 8 we use tensotaxis to induce the coalescence of two liquid droplets of different sizes. The droplets are "pushed" toward each other by the action of two localized forces; see panel (d). The droplets approach each other and eventually coalesce in a single droplet. The resulting droplet is trapped by the forces of different sign, which illustrates the potential of tensotaxis to manipulate liquid droplets. Note that the presence of

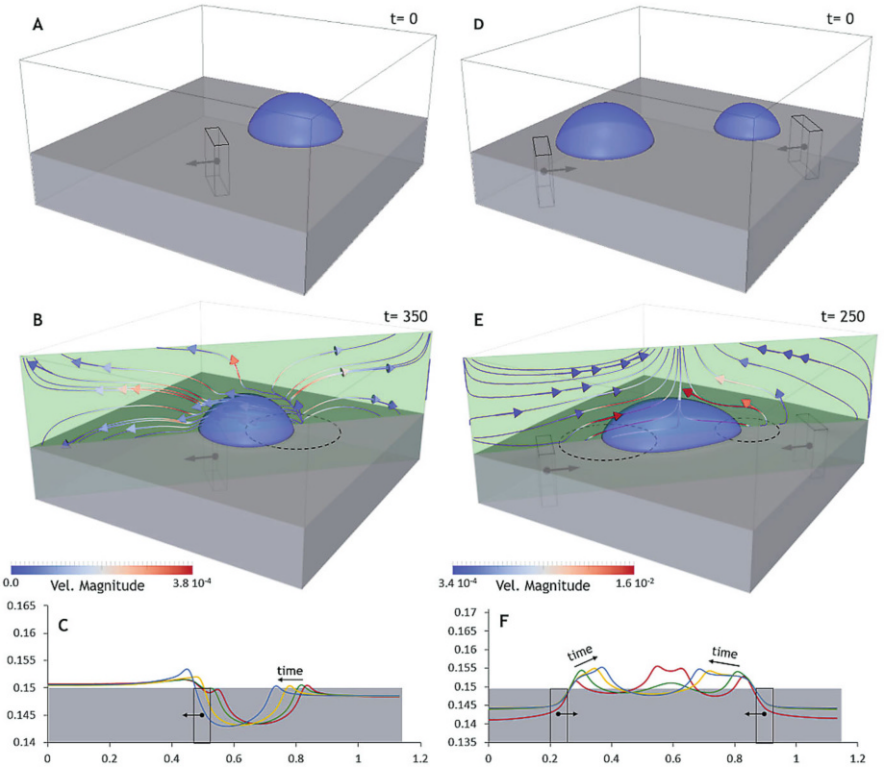


Fig. 8 Three-dimensional droplet motion triggered by tensotaxis. Left panels present the 3D analogue of the needle experiment. Right panels show how tensotaxis can be used to induce the coalescence of liquid droplets. (**a**) A liquid droplet is deposited on a deformable substrate. A horizontal force is applied at the center of the substrate. (**b**) The droplet moves in the direction of the applied force. (**d**) Two forces of the same magnitude are applied on the substrate, pushing the droplets towards the center of the domain. (**e**) When the droplets are sufficiently close, capillary forces promote coalescence of the two droplets. The black, dashed line indicates the initial position of the droplet. (**c**) and (**f**) Vertical displacement of the solid–liquid interface at the diagonal plane [green color in panels (**b**) and (**e**)] at different times. In both experiments the computational domain $\Omega = [0, 0.8] \times [0, 0.8] \times [0, 0.4]$ is discretized with $80 \times 80 \times 40$ $\mathcal{C}^1$-quadratic elements. We have adopted $\nu^s = 0.45$, $\widehat{\mu} = 1/200$, $\widehat{\gamma} = 1/50$, $\widehat{E} = 0.7554$, $\widehat{\theta} = 0.39$, and $\alpha = 75°$. The load that triggers droplet motion is a body force per unit mass of value $\left|\widehat{\boldsymbol{f}}^s\right| = 1.376$ and $\left|\widehat{\boldsymbol{f}}^s\right| = 2.7519$ for the left and right columns, respectively

multiple droplets increases the complexity of the problem, as we must consider their mechanical interactions through the deformable substrate, as well as through the fluid domain.

5 Ellastocapillarity

The FSI models presented so far describe the physics of single-component two-phase flows interacting with hyperelastic structures. However, there are a number of physical and engineering problems in which the fluid is not formed by phases of the same component. It is the case, for example, of gas injection for oil recovering, fuel sprays in combustion processes, air-water flows, etc. For modeling this kind of problems, we propose a computational model that couples nonlinear hyperelastic solids with two-component immiscible fluids described by the Navier–Stokes–Cahn–Hilliard (NSCH) equations.

We use the proposed theory to study a variety of problems in which capillary forces at fluid–fluid interfaces deform soft and/or slender structures. This phenomenon—elastocapillarity [68, 74]—can be observed, for example, when water interacts with hair and other flexible fibers, which tend to assemble into bundles [10, 30]. However, it is in micro and nanotechnologies, where the effect of elastocapillary forces is sparking greatest interest. Understanding elastocapillarity is essential for the design of new materials and devices at small scales. Capillary forces can damage microelectromechanical structures [66, 83] and carbon nanotube carpets [16]. However, under control, they can be used as a fabrication technique to deform straight pillars and build complex and robust 3D geometries at the micrometer and nanometer scales [26, 84]. We think that our theory could be a useful tool to understand elastocapillary phenomena and thus, contribute to a better design of manufacturing processes at small scales.

5.1 Governing Equations

In our model, the behavior of the structure will be governed by the momentum balance equation, which has been previously presented in Sect. 3.1; see also [14]. The dynamics of the fluid system is governed by the Navier–Stokes–Cahn–Hilliard equations, which describe the behavior of two immiscible and incompressible fluids with matched density and viscosity. The ALE form of the NSCH equations can be written as follows:

$$\nabla \cdot \boldsymbol{v} = 0, \tag{28a}$$

$$\rho \left(\left. \frac{\partial \boldsymbol{v}}{\partial t} \right|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla \boldsymbol{v} \right) - \nabla \cdot \boldsymbol{\sigma}^f = 0, \tag{28b}$$

$$\left. \frac{\partial \psi}{\partial t} \right|_{\widehat{x}} + (\boldsymbol{v} - \widehat{\boldsymbol{v}}) \cdot \nabla \psi - \nabla \cdot \left(M_\psi \gamma_{LG} \nabla \left(\frac{1}{\epsilon} W'_\psi - \epsilon \Delta \psi \right) \right) = 0, \tag{28c}$$

where $\widehat{\boldsymbol{v}}$ is the velocity of the fluid domain [4], ρ is the fluid density that we consider to be constant, $\boldsymbol{f}$ represents body forces per unit mass, and $\psi \in [-1, 1]$ is the phase-field variable. W_ψ is a double-well potential, which we define as $W_\psi = \left(\psi^2 - 1\right)^2/4$. The constant γ_{LG} denotes the liquid–gas surface tension and ϵ is the interface thickness. M_ψ represents the mobility, which is assumed to be constant for the examples presented herein. The fluid Cauchy stress tensor $\boldsymbol{\sigma}^f$ is given by $\boldsymbol{\sigma}^f = \boldsymbol{\tau} - p\boldsymbol{I} - \gamma_{LG}\epsilon\nabla\psi \otimes \nabla\psi$, Here, p represents the mechanical pressure and $\boldsymbol{\tau}$ is the viscous stress tensor of an incompressible Newtonian fluid, i.e., $\boldsymbol{\tau} = \bar{\mu}\left(\nabla\boldsymbol{v} + \nabla^T\boldsymbol{v}\right)$, where $\bar{\mu}$ is the viscosity coefficient. We will assume that the viscosity $\bar{\mu}$ is a constant parameter for the problems presented in this work.

In the fluid mechanics problem we focus on solid-wall boundary conditions by setting the velocity to a given value. Additionally, and with the purpose of attaining well-posedness in the NSCH system, we impose the following boundary conditions on the phase-field variable,

$$\nabla\psi \cdot \boldsymbol{n}^f = |\nabla\psi|\cos(\alpha) \qquad \text{on} \quad \Gamma_t, \tag{29a}$$

$$\nabla\psi \cdot \boldsymbol{n}^f = 0 \qquad \text{on} \quad \Gamma_t^f, \tag{29b}$$

and

$$M_\psi\gamma_{LG}\nabla\left(\frac{1}{\epsilon}W'_\psi - \epsilon\Delta\psi\right)\cdot\boldsymbol{n}^f = 0 \quad \text{on} \quad \Gamma_t^f \cup \Gamma_t, \tag{30}$$

where $\boldsymbol{n}^f$ is the unit outward normal to the fluid boundary and α denotes the contact angle between the fluid–fluid interface and the solid, measured in the phase of value $\psi = 1$. The imposition of boundary condition (29) can be notably simplified by introducing in our problem a new variable $\Upsilon = \Delta\psi$. This implies that the phase-field equation (28c) is split into two lower-order equations, allowing to use the classical finite element method for the spatial discretization of the problem.

5.2 Numerical Formulation

The numerical scheme for our FSI problem is similar to those presented in [4] and [13]. We solve the coupled system composed by Eqs. (3) and (28) subject to the kinematic compatibility and traction balance constraints. We compute the variational form of the system and discretize in space using isogeometric analysis. The reader is referred to [14] for more details.

5.2.1 Semidiscrete Formulation

We use NURBS-based isogeometric analysis for the spatial discretization of the problem. Additionally, in order to stabilize the NSCH equations, we make use of the VMS method and split the weighting and the trial solution spaces corresponding to the pressure and the velocity field into a coarse and a fine scale subspaces. Let H^1 be the Sobolev space of square-integrable functions with square-integrable first derivatives. We denote by $\mathcal{X}_h^\psi$ and $\mathcal{X}_h^\Upsilon$ a finite-dimensional aproximation of the trial solution spaces for Υ and ψ, respectively. $\mathcal{X}_h^\psi$ and $\mathcal{X}_h^\Upsilon$ are subsets of $H^1(\Omega_t^f)$. Let $\mathcal{Y}_h^\psi$ and $\mathcal{Y}_h^\Upsilon$ be the corresponding finite-dimensional weigthing function spaces. $\mathcal{Y}_h^\psi$ and $\mathcal{Y}_h^\Upsilon$ are identical to $\mathcal{X}_h^\psi$ and $\mathcal{X}_h^\Upsilon$, respectively. Analogously, let us define a finite-dimensional trial solution space for the coarse scales of $\boldsymbol{v}$ and p, which we denote by $\mathcal{X}_h^v$ and $\mathcal{X}_h^p$, respectively. $\mathcal{X}_h^v$ is a subset of $H^1(\Omega_t^f)$ whose members satisfy all Dirichlet boundary conditions. The weighting function space for p, i.e., $\mathcal{Y}_h^p$ is identical to $\mathcal{X}_h^p$. Let $\mathcal{X}_h^s$ and $\mathcal{V}_h^m$ be a finite-dimensional approximation of the trial solution spaces for the solid and mesh displacement variables $\boldsymbol{u}$ and $\boldsymbol{u}^m$, respectively. $\mathcal{X}_h^s$ and $\mathcal{V}_h^m$ are subsets of $H^1\left(\Omega_0^s\right)$ and $H^1(\Omega_t^f)$, respectively, whose members satisfy all Dirichlet boundary conditions; see Sect. 3.2 for more details on the fluid mesh motion problem. Finally, we denote by $\mathcal{Y}_h^s$ and $\mathcal{W}_h^m$ the corresponding finite-dimensional weighting function spaces, which are identical to $\mathcal{X}_h^s$ and $\mathcal{V}_h^m$, except that all restrictions on the Dirichlet boundary are homogeneous.

With the previous considerations, we can approximate our FSI problem by the following variational formulation over the finite element spaces: Find $p_h \in \mathcal{X}_h^p$, $\boldsymbol{v}_h \in \mathcal{X}_h^v$, $\psi_h \in \mathcal{X}_h^\psi$, $\Upsilon_h \in \mathcal{X}_h^\Upsilon$, $\boldsymbol{u}_h \in \mathcal{X}_h^s$ and $\boldsymbol{u}_h^m \in \mathcal{V}_h^m$ such that $\forall w_h^1 \in \mathcal{Y}_h^p$, $\boldsymbol{w}_h^2 \in \mathcal{Y}_h^v$, $w_h^3 \in \mathcal{Y}_h^\psi$, $w_h^4 \in \mathcal{Y}_h^\Upsilon$, $\boldsymbol{w}_h^s \in \mathcal{Y}^s{}_h$ and $\boldsymbol{w}_h^m \in \mathcal{W}_h^m$,

$$B_{MS}^f(w_h^1, \boldsymbol{w}_h^2, w_h^3, w_h^4, p_h, \boldsymbol{v}_h, \psi_h, \Upsilon_h; \widehat{\boldsymbol{v}_h}) + B^s(\boldsymbol{w}_h^s, \boldsymbol{u}_h) + B^m(\boldsymbol{w}_h^m, \boldsymbol{u}_h^m) = F^s(\boldsymbol{w}_h^s), \tag{31}$$

where B^s and F^s constitute the weak form of the solid equations defined in Eqs. (10) and (11), respectively. B^m denotes the weak form of the mesh motion problem; see Eq. (15). Finally, B_{MS}^f stands for the stabilized weak form of the fluid problem, i.e.,

$$\begin{aligned}
B_{MS}^f(w_h^1, \boldsymbol{w}_h^2, w_h^3, w_h^4, p_h, \boldsymbol{v}_h, \psi_h, \Upsilon_h; \widehat{\boldsymbol{v}_h}) &= \int_{\Omega_t^f} w_h^1 \nabla \cdot \boldsymbol{v}_h \mathrm{d}\Omega_t^f \\
&+ \int_{\Omega_t^f} \boldsymbol{w}_h^2 \cdot \rho \left(\left.\frac{\partial \boldsymbol{v}_h}{\partial t}\right|_{\widehat{x}} + (\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) \cdot \nabla \boldsymbol{v}_h \right) \mathrm{d}\Omega_t^f + \int_{\Omega_t^f} \nabla \boldsymbol{w}_h^2 : \boldsymbol{\sigma}_h^f \mathrm{d}\Omega_t^f \\
&+ \int_{\Omega_t^f} w_h^3 \left(\left.\frac{\partial \psi_h}{\partial t}\right|_{\widehat{x}} + (\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) \cdot \nabla \psi_h \right) \mathrm{d}\Omega_t^f \\
&+ \int_{\Omega_t^f} \nabla w_h^3 \cdot M_\psi \gamma_{LG} \nabla \left(\frac{1}{\epsilon} W_\psi' - \epsilon \Upsilon_h \right) \mathrm{d}\Omega_t^f
\end{aligned}$$

$$
\begin{aligned}
&+ \int_{\Omega_t^f} w_h^4 \Upsilon_h \mathrm{d}\Omega_t^f + \int_{\Omega_t^f} \nabla w_h^4 \cdot \nabla \psi_h \mathrm{d}\Omega_t^f - \int_{\Gamma_t} w_h^4 |\nabla \psi_h| \cos(\alpha) \mathrm{d}\Gamma_t \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_t^{f,e}} \tau_{SUPS} \left((\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) \cdot \nabla \boldsymbol{w}_h^2 + \frac{\nabla w_h^1}{\rho} \right) \cdot \boldsymbol{r}_M \mathrm{d}\Omega_t^f \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_t^{f,e}} \nu_{LSIC} \rho \nabla \cdot \boldsymbol{w}_h^2 r_C \mathrm{d}\Omega_t^f - \sum_{e=1}^{n_{el}} \int_{\Omega_t^{f,e}} \tau_{SUPS} \boldsymbol{w}_h^2 \cdot (\boldsymbol{r}_M \cdot \nabla \boldsymbol{v}_h) \, \mathrm{d}\Omega_t^f \\
&- \sum_{e=1}^{n_{el}} \int_{\Omega_t^{f,e}} \frac{\nabla \boldsymbol{w}_h^2}{\rho} : (\tau_{SUPS} \boldsymbol{r}_M \otimes \tau_{SUPS} \boldsymbol{r}_M) \, \mathrm{d}\Omega_t^f \\
&- \sum_{e=1}^{n_{el}} \int_{\Omega_t^{f,e}} \frac{\tau_{SUPS}}{\rho} w_h^3 \boldsymbol{r}_M \cdot \nabla \psi_h \mathrm{d}\Omega_t^f
\end{aligned} \tag{32}
$$

Here, n_{el} is the total number of elements on the fluid mesh and $\Omega_t^{f,e}$ the region of the physical space occupied by element e at time t. In Eq. (32), $\boldsymbol{r}_M$ and r_C are given by

$$
\boldsymbol{r}_M = \rho \left(\left. \frac{\partial \boldsymbol{v}_h}{\partial t} \right|_{\widehat{x}} + (\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) \cdot \nabla \boldsymbol{v}_h \right) - \nabla \cdot \boldsymbol{\sigma}_h^f, \tag{33a}
$$

$$
r_C = \nabla \cdot \boldsymbol{v}_h, \tag{33b}
$$

where $\boldsymbol{\sigma}_h^f = \bar{\mu} \left(\nabla \boldsymbol{v}_h + \nabla^T \boldsymbol{v}_h \right) - p_h \boldsymbol{I} - \gamma_{LG} \epsilon \nabla \psi_h \otimes \nabla \psi_h$. For the stabilization parameters τ_{SUPS} and ν_{LSIC} we use the expressions

$$
\tau_{SUPS} = \left(\frac{4}{\Delta t^2} + (\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) \cdot \boldsymbol{G} \, (\boldsymbol{v}_h - \widehat{\boldsymbol{v}}_h) + C_I \nu^2 \boldsymbol{G} : \boldsymbol{G} \right)^{-1/2}, \tag{34a}
$$

$$
\nu_{LSIC} = \left(\mathrm{tr}(\boldsymbol{G}) \, \tau_{SUPS} \right)^{-1}, \tag{34b}
$$

where Δt is, in a time-discrete context, the time step; $C_I = 1/12$ is a positive constant; ν is the kinematic viscosity; and $\boldsymbol{G}$ denotes the element metric tensor [7].

5.2.2 Time Integration

For the time integration we adopt the generalized-α method. The nonlinear system of equations is solved using a Newton–Raphson iteration procedure; see [14].

5.3 Numerical Examples

5.3.1 Static Wetting on Soft Substrates

The static configuration that single droplets adopt at small scales on flat and infinitely rigid substrates is relatively well understood. It is known that droplets tend to form a spherical cap with an equilibrium contact angle α given by Young–Dupré equation [23], i.e., $\alpha = \arccos[(\gamma_{SG} - \gamma_{SL})/\gamma_{LG}]$; γ_{SG} and γ_{SL} denote the interfacial tension at the solid–gas and solid–liquid interfaces, respectively. However, when droplets interact with sufficiently soft solids, Young–Dupré equation breaks down. This happens for droplets smaller than the elastocapillary length scale $l_{ec} = \gamma_{LG}/E^s$. In these cases, the Laplace pressure inside the droplet creates a dimple in the wet area and capillary forces produce a ridge at the contact line. We show that our model can successfully predict this phenomenon by reproducing one of the experiments reported in [72]; see Fig. 9. A glycerol droplet of radius $R = 225.5\,\mu\text{m}$ is deposited on a soft substrate with Young modulus $E^s = 3.0\,\text{kPa}$. The droplet deforms the substrate as shown on the left panel of Fig. 9. The contact angle α imposed at the fluid–structure interface is larger than the apparent contact angle (measured with respect to the undeformed surface of the substrate). Analogously to [88], we compare our measurements of the vertical displacements at the fluid–solid interface close to the contact line with the experimental data reported in [72]; see right panel in Fig. 9. The results show a good agreement between both curves.

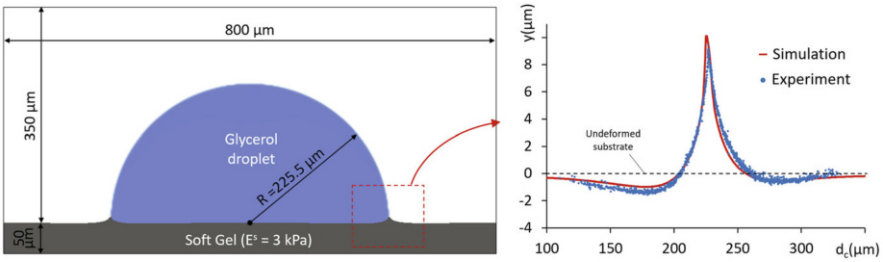


Fig. 9 Glycerol droplet deposited on a soft substrate. Stationary configuration (left) and vertical displacements of the fluid–solid interface y close to the contact line (right). d_c in the horizontal axis represents the distance to the center of the droplet. The blue points are the experimental measurements reported in [72]. The computational results are reported at $t = 21.0\,ms$ (red), when the solution is considered to be stationary. The horizontal dashed line $y = 0$ corresponds to the fluid–solid interface before deformation. We use a uniform mesh of 400×200 quadratic elements. We adopted the values of surface tension reported in [72], i.e., $\gamma_{LG} = 46.0\,\text{mN/m}$, $\gamma_{SA} = 31\,\text{mN/m}$ and $\gamma_{SL} = 36\,\text{mN/m}$. For the viscosity and density parameters, we use values for glycerol: $\mu = 1412.0\,\text{mPa s}$ and $\rho = 1.26\,\text{pg}/\mu\text{m}^3$. We took $\epsilon = 2.0\,\mu\text{m}$ and $M_\psi = 0.1\,\mu\text{m}^3\mu\text{s/pg}$. The parameters for the solid correspond to a silicone gel with $E^s = 3.0\,\text{kPa}$, $\nu^s = 0.499$ and $\rho^s = 12.6\,\text{pg}/\mu\text{m}^3$

5.3.2 Capillary Origami

We use our theory to simulate the spontaneous wrapping of glycerol droplets with elastic membranes in two and three dimensions; see Fig. 10. We place a semicircular droplet (blue) of radius $R = 146.0\,\mu\text{m}$ on an elastic membrane (grey) and let the capillary forces fold the membrane in absence of gravity forces; panels (a) and (d) show the setup of the problem. In the two-dimensional problem we impose different contact angles α along the fluid–structure interface. In particular, we show the results for two cases, $\alpha = 60°$ [panel (b)] and $\alpha = 120°$ [panel (C)]. The capillary forces are able to deform the structure, which gradually wraps the droplet. The configurations adopted by the system differ considerably depending on the contact angle at the fluid–solid interface. The results suggest that droplet encapsulation occurs faster for more wetting fluids. This observation can be used to have better control in the process of self-assembly of micro- and nano-structures. Figure 10b, c also show the streamlines of the fluid velocity at times $t = 0.065\,\text{s}$ and $t = 0.302\,\text{s}$, respectively. The flow inside the droplet is predominantly vertical. Outside the droplet, the membrane's motion produces well-defined vortices.

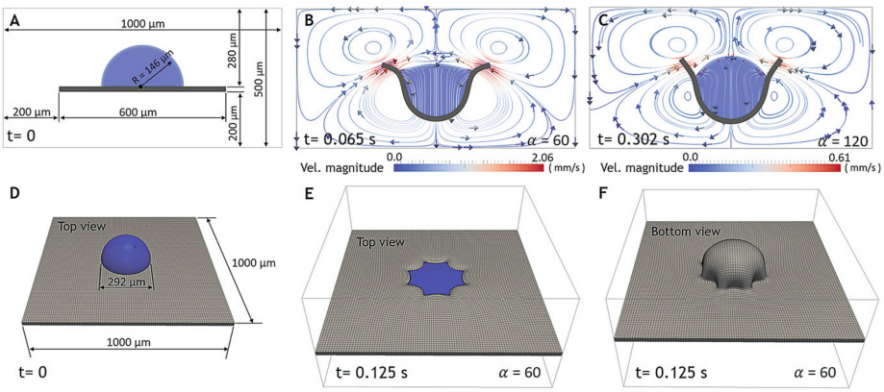


Fig. 10 Spontaneous wrapping of glycerol droplets in 2D (upper panels) and 3D (lower panels). (**a**) Initial condition. A semicircular droplet (blue) is deposited on an elastic membrane (grey). (**b**) and (**c**) Current configuration for two different contact angles: $\alpha = 60°$ and $\alpha = 120°$ at $t = 0.065\,\text{s}$ and $t = 0.302\,\text{s}$, respectively. Panels (**b**) and (**c**) also show the streamlines of the fluid velocity colored with the velocity magnitude. The arrows present the direction of the velocity field. (**d**) Initial configuration of the 3D problem. The elastic membrane is clamped to the lateral boundaries of the computational domain. (**e**) Capillary forces fold the membrane and lead to the partial wrapping of the liquid droplet. (**f**) A wrinkling instability is developed on the structure, triggered by the non-axisymmetric shape of the membrane. We use a uniform mesh of 200×100 and $100 \times 100 \times 50$ quadratic elements for the 2D and 3D problems, respectively. The parameters employed in this computation for the fluid correspond to a glycerol droplet, i.e., $\mu = 1412.0\,\text{mPa}\,\text{s}$, $\gamma_{LG} = 46.0\,\text{mN/m}$, and $\rho = 1.26\,\text{pg}/\mu\text{m}^3$. We also take $M_\psi = 0.1\,\mu\text{m}^3\mu\text{s/pg}$, $\nu^s = 0.45$, and $\rho^s = 12.6\,\text{pg}/\mu\text{m}^3$. For the 2D case, we have adopted $\epsilon = 5.0\,\mu\text{m}$ and $E^s = 30.0\,\text{kPa}$, $\nu^s = 0.45$. In the 3D problem, $\epsilon = 10.0\,\mu\text{m}$ and $E^s = 0.6\,\text{kPa}$

In the three-dimensional problem [Fig. 10d–f] the elastic membrane covers an entire horizontal plane of the computational domain. The solid structure is clamped to the lateral boundaries, where we are preventing the motion in normal and vertical directions. The droplet is a wetting liquid that forms a static contact angle of $\alpha = 60°$ with the elastic membrane. Panels (e) and (f) show an upper and bottom view of the problem, respectively. The surface tension at the interface folds the solid resulting in the partial wrapping of the droplet; see panel (e). The initial square shape of the elastic membrane leads to a non-axisymmetric deformation of the structure and results in the formation of wrinkles; see panel (f). Different patterns of the wrinkling instability could be obtained for different values of surface tension and solid stiffness [15, 42].

5.3.3 Wetting of Elastic Micropillars

At small scales, capillary forces may deform or even collapse slender structures. Surface tension may cause important damage in microelectromechanical systems [66], destroy micropatterns in photoresist materials [68], or produce the collapse of carbon nanotube carpets [53]. For this reason, wetting of fibrous media has been widely studied in recent years. However, most of the efforts focused on rigid fibers. Here, we use our FSI model to study quantitatively the deformation of elastic micropillars produced by capillary forces; see Fig. 11. We place a small amount of liquid (blue) between two micropillars (grey) and let capillary forces deform the fibers. The micropillars are clamped at the bottom, where we are preventing the displacements in horizontal and vertical directions. Panels (b) and (c) show how surface tension deforms the fibers and brings them closer to each other, which, in turn, produces the rise of the liquid. The insets in panels (b) and (c) present the mesh

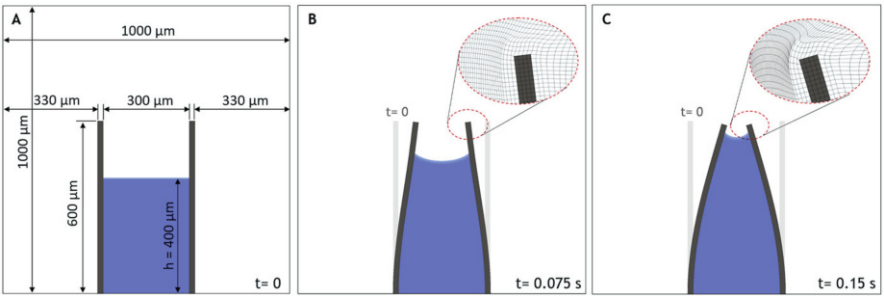


Fig. 11 Wetting of elastic micropillars. (**a**) Setup of the problem. A small amount of liquid (blue) is placed between two elastic micropillars (grey). (**b**) and (**c**) show the configuration of the problem at two different time instants. The capillary forces deform the fibers producing the rise of the liquid. The computational domain is discretized with a uniform mesh of 200^2 quadratic elements. We use a wetting liquid with contact angle $\alpha = 60°$. The parameters are $\mu = 1412.0\,\text{mPa s}$, $\gamma_{LG} = 46.0\,\text{mN/m}$, $\epsilon = 2.5\,\mu\text{m}$, $\rho = 1.26\,\text{pg}/\mu\text{m}^3$, $M_\psi = 0.1\,\mu\text{m}^3\mu\text{s/pg}$, $E^s = 320.0\,\text{kPa}$, $\nu^s = 0.45$ and $\rho^s = 12.6\,\text{pg}/\mu\text{m}^3$

distortion close to the tip of the micropillars. We have compared the deformation of the fibers for different amounts of liquid (not shown here) and we observed that the volume of liquid plays a key role in the dynamics of the problem. For a given geometrical configuration, larger volumes of liquid produce larger and faster deformations of the structure.

References

1. J. J. Agresti, E. Antipov, A. R. Abate, K. Ahn, A. C. Rowat, J.-C. Baret, M. Marquez, A. M Klibanov, A. D. Griffiths, and D. A. Weitz. Ultrahigh-throughput screening in drop-based microfluidics for directed evolution. *Proceedings of the National Academy of Sciences*, 107(9):4004–4009, 2010.
2. D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30:139–165, 1998.
3. G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, 2000.
4. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric Fluid-Structure Interaction: Theory, algorithms, and computations. *Computational Mechanics*, 43(1):3–37, 2008.
5. Y. Bazilevs, M.-C. Hsu, D.J. Benson, S. Sankaran, and A.L. Marsden. Computational Fluid-Structure Interaction: Methods and application to a total cavopulmonary connection. *Computational Mechanics*, 45(1):77–89, 2009.
6. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Challenges and directions in computational fluid-structure interaction. *Mathematical Models and Methods in Applied Sciences*, 23:215–221, 2013.
7. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid-Structure Interaction. Methods and Applications*. Wiley, 2013.
8. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre. Aerodynamic and FSI Analysis of Wind Turbines with the ALE-VMS and ST-VMS Methods. *Archives of Computational Methods in Engineering*, 21(4):359–398, 2014.
9. T. Biben, K. Kassner, and C. Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72:041921, Oct 2005.
10. J. Bico, B. Roman, L. Moulin, and A. Boudaoud. Adhesion: elastocapillary coalescence in wet hair. *Nature*, 432(7018):690–690, 2004.
11. I. B. Bischofs and U. S. Schwarz. Cell organization in soft media due to active mechanosensing. *Proceedings of the National Academy of Sciences*, 100(16):9274–9279, 2003.
12. J. Bueno, Y. Bazilevs, R. Juanes, and H. Gomez. Droplet motion driven by tensotaxis. *Extreme Mechanics Letters*, 13:10–16, 2017.
13. J. Bueno, C. Bona-Casas, Y. Bazilevs, and H. Gomez. Interaction of complex fluids and solids: Theory, algorithms and application to phase-change-driven implosion. *Computational Mechanics*, 55(6):1105–1118, 2015.
14. J. Bueno, H. Casquero, Y. Bazilevs, and H. Gomez. Three-dimensional dynamic simulation of elastocapillarity. *Meccanica*, 53(6):1221–1237, 2018.
15. E. Cerda and L. Mahadevan. Geometry and physics of wrinkling. *Physical Review Letters*, 90(7):074302, 2003.
16. N. Chakrapani, B. Wei, A. Carrillo, P. M. Ajayan, and R. S. Kane. Capillarity-driven assembly of two-dimensional cellular carbon nanotube foams. *Proceedings of the National Academy of Sciences*, 101(12):4009–4012, 2004.
17. L.Q. Chen. Phase-field models for microstructure evolution. *Annual Review of Materials Research*, 32(1):113–140, 2002.
18. P. Chiquet, D. Broseta, and S. Thibeau. Wettability alteration of caprock minerals by carbon dioxide. *Geofluids*, 7(2):112–122, 2007.

19. 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*, 60:371–375, 1993.
20. B. D. Coleman and W. Noll. The thermodynamics of elastic materials with heat conduction and viscosity. *Archive for Rational Mechanics and Analysis*, 13(1):167–178, 1963.
21. L. Cueto-Felgueroso and R. Juanes. Nonlocal interface dynamics and pattern formation in gravity-driven unsaturated flow through porous media. *Physical Review Letters*, 101:244504, 2008.
22. L. Cueto-Felgueroso and R. Juanes. Macroscopic phase-field modeling of partial wetting: bubbles in a capillary tube. *Physical Review Letters*, 108:144502, 2012.
23. P. G. de Gennes. Wetting: statics and dynamics. *Review of Modern Physics*, 57:827–863, 1985.
24. P.-G. De Gennes, F. Brochard-Wyart, and D. Quéré. *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*. Springer Science & Business Media, 2004.
25. L. Dedè, M.J. Borden, and T.J.R. Hughes. Isogeometric analysis for topology optimization with a phase field model. *Archives of Computational Methods in Engineering*, 19(3):427–465, 2012.
26. M. DeVolder and A. J. Hart. Engineering hierarchical nanostructures by elastocapillary self-assembly. *Angewandte Chemie International Edition*, 52(9):2412–2425, 2013.
27. D. Diehl. *Higher Order Schemes for Simulation of Compressible Liquid-Vapor Flow with Phase Change*. PhD thesis, Albert-Ludwigs-Universitt Freiburg, 2007.
28. J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, 2003.
29. J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodrguez-Ferran. *Encyclopedia of Computational Mechanics. Arbitrary Lagrangian-Eulerian Methods.*, volume 1, chapter 14. John Wiley & Sons, Ltd, 2004.
30. C. Duprat, A. D. Bick, P. B. Warren, and H. A. Stone. Evaporation of drops on two parallel fibers: Influence of the liquid morphology and fiber elasticity. *Langmuir*, 29(25):7857–7863, 2013. PMID: 23705986.
31. A. Fallah-Araghi, J.-C. Baret, M. l. Ryckelynck, and A. D. Griffiths. A completely in vitro ultrahigh-throughput droplet-based microfluidic screening system for protein engineering and directed evolution. *Lab on a Chip*, 12(5):882–891, 2012.
32. C. Farhat, A. Rallu, K. Wang, and T. Belytschko. Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible Fluid-Structure Interaction problems. *International Journal for Numerical Methods in Engineering*, 84(1):73–107, 2010.
33. C. Farhat, K. G. Van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17):1973–2001, 2006.
34. I. Fonseca, M. Morini, and V. Slastikov. Surfactants in foam stability: A phase-field model. *Archive for Rational Mechanics and Analysis*, 183(3):411–456, 2007.
35. E. Fried and M.E. Gurtin. Dynamic solid-solid transitions with phase characterized by an order parameter. *Physica D: Nonlinear Phenomena*, 72(4):287–308, 1994.
36. J.W. Gibbs. *On the Equilibrium of Heterogeneous Substances*. 1876.
37. H. Gomez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric Analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:43334352, 2008.
38. H. Gomez, L. Cueto-Felgueroso, and R. Juanes. Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium. *Journal of Computational Physics*, 238:217–239, 2013.
39. H. Gomez and T.J.R. Hughes. Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models. *Journal of Computational Physics*, 230(13):5310–5327, 2011.
40. H. Gomez, T.J.R. Hughes, X. Nogueira, and V.M. Calo. Isogeometric Analysis of the isothermal Navier-Stokes-Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199(25–28):1828–1840, 2010.

41. H. Gomez and K. van der Zee. *Encyclopedia of Computational Mechanics. Computational phase-field modeling*. John Wiley & Sons, Ltd, 2017.
42. J. Huang, M. Juszkiewicz, W. H. de Jeu, E. Cerda, T. Emrick, N. Menon, and T. P. Russell. Capillary wrinkling of floating thin polymer films. *Science*, 317(5838):650–653, 2007.
43. 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(39–41):4135–4195, 2005.
44. 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(3):329–349, 1981.
45. C.M. Ikeda. *Fluid-Structure Interactions. Implosions of shell structures and wave impact on a flat plate*. PhD thesis, University of Maryland, 2012.
46. J.H. Jeong, N. Goldenfeld, and J.A. Dantzig. Phase field model for three-dimensional dendritic growth with fluid flow. *Physical Review E*, 64:041602, 2001.
47. D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T.J.R. Hughes. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
48. K. Kamran, R. Rossi, E. Oñate, and S.R. Idelshon. 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(02):339–367, 2013.
49. K. Kamran, R. Rossi, E. Oñate, and S.R. Idelsohn. A compressible lagrangian framework for the simulation of the underwater implosion of large air bubbles. *Computer Methods in Applied Mechanics and Engineering*, 255:210–225, 2013.
50. S. Karpitschka, A. Pandey, L. A. Lubbers, J. H. Weijs, L. Botto, S. Das, B. Andreotti, and J. H. Snoeijer. Liquid drops attract or repel by the inverted cheerios effect. *Proceedings of the National Academy of Sciences*, 113(27):7403–7407, 2016.
51. J. Kiendl, M. Ambati, L. De Lorenzis, H. Gomez, and A. Reali. Phase-field description of brittle fracture in plates and shells. *Computer Methods in Applied Mechanics and Engineering*, 312:374–394, 2016.
52. H. Lamb. *Hydrodynamics*. Courier Corporation, 1945.
53. K. K. Lau, J. Bico, K. B. Teo, M. Chhowalla, G. A. Amaratunga, W. Milne, G. H McKinley, and K. Gleason. Superhydrophobic carbon nanotube forests. *Nano letters*, 3(12):1701–1705, 2003.
54. S.-L. Lin, J.-C. Yang, K.-N. Ho, C.-H. Wang, C.-W. Yeh, and H.-M. Huang. Effects of compressive residual stress on the morphologic changes of fibroblasts. *Medical & Biological Engineering & Computing*, 47(12):1273–1279, 2009.
55. J. Liu. *Thermodynamically consistent modeling and simulation of multiphase flows*. PhD thesis, The University of Texas at Austin, 2014.
56. J. Liu, H. Gomez, J.A. Evans, T.J.R. Hughes, and C.M. Landis. Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier-Stokes-Korteweg equations. *Journal of Computational Physics*, 248:47–86, 2013.
57. C.-M. Lo, H.-B. Wang, M. Dembo, and Y.-L. Wang. Cell movement is guided by the rigidity of the substrate. *Biophysical Journal*, 79(1):144–152, 2000.
58. C.C. Long, M. Esmaily-Moghadam, A.L. Marsden, and Y. Bazilevs. Computation of residence time in the simulation of pulsatile ventricular assist devices. *Computational Mechanics*, 54:911–919, 2014.
59. J. E. Marsden and T. J. R. Hughes. *Mathematical Foundations of Elasticity*. Prentice-Hall, Englewood Cliffs, NJ, 1983. Reprinted with corrections, Dover, New York, 1994.
60. C Miehe, F Welschinger, and M Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field fe implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311, 2010.

61. P. Moreo, J. M. García-Aznar, and M. Doblaré. Modeling mechanosensing and its effect on the migration and proliferation of adherent cells. *Acta Biomaterialia*, 4(3):613–621, 2008.
62. O. Penrose and P.C. Fife. Thermodynamically consistent models of phase-field type for the kinetic of phase transitions. *Physica D: Nonlinear Phenomena*, 43(1):44–62, 1990.
63. C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.
64. S.D. Poisson. *Nouvelle théorie de l'action capillaire*. Bachelier père et fils, 1831.
65. N. Provatas and K. Elder. *Phase-Field Methods in Materials Science and Engineering*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010.
66. O. Raccurt, F. Tardif, F. A. d'Avitaya, and T. Vareine. Influence of liquid surface tension on stiction of SOI MEMS. *Journal of Micromechanics and Microengineering*, 14(7):1083, 2004.
67. A.S.D. Rallu. *A Multiphase Fluid-Structure Computational Framework for Underwater Implosion Problems*. PhD thesis, Standford University, 2009.
68. B. Roman and J. Bico. Elasto-capillarity: deforming an elastic structure with a liquid droplet. *Journal of Physics: Condensed Matter*, 22(49):493101, 2010.
69. J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer-Verlag, New Yoirk, 1998.
70. H. Song, D.L. Chen, and R.F. Ismagilov. Reactions in droplets in microfluidic channels. *Angewandte Chemie International Edition*, 45(44):7336–7356, 2006.
71. I. Steinbach. Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001, 2009.
72. R. W. Style, R. Boltyanskiy, Y. Che, J. S. Wettlaufer, L. A. Wilen, and E. R. Dufresne. Universal deformation of soft substrates near a contact line and the direct measurement of solid surface stresses. *Physical Review Letters*, 110:066103, Feb 2013.
73. R. W. Style and E. R. Dufresne. Static wetting on deformable substrates, from liquids to soft solids. *Soft Matter*, 8(27):7177–7184, 2012.
74. R. W. Style, A. Jagota, C.-Y. Hui, and E. R. Dufresne. Elastocapillarity: Surface tension and the mechanics of soft solids. *Annual Review of Condensed Matter Physics*, 8(0), 2016.
75. R.W. Style, Y. Che, S.J. Park, B.M. Weon, J.H. Je, C. Hyland, G.K. German, M.P. Power, L.A. Wilen, J.S. Wettlaufer, et al. Patterning droplets with durotaxis. *Proceedings of the National Academy of Sciences*, 110(31):12541–12544, 2013.
76. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, and T. Ueda. FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta. *Computational Mechanics*, 54:1035–1045, 2014.
77. R. Sunyer, V. Conte, J. Escribano, A. Elosegui-Artola, A. Labernadie, L. Valon, D. Navajas, J. M. García-Aznar, J. J. Muñoz, P. Roca-Cusachs, and X. Trepat. Collective cell durotaxis emerges from long-range intercellular force transmission. *Science*, 353(6304):1157–1161, 2016.
78. K. Takizawa. Computational engineering analysis with the new-generation space–time methods. *Computational Mechanics*, 54:193–211, 2014.
79. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt. ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling. *Mathematical Models and Methods in Applied Sciences*, 24:2437–2486, 2014.
80. K. Takizawa, H. Takagi, T. E. Tezduyar, and R. Torii. Estimation of element-based zero-stress state for arterial FSI computations. *Computational Mechanics*, 54:895–910, 2014.
81. K. Takizawa, T. E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel. Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes. *Computational Mechanics*, 54(6):1461–1476, 2014.
82. K. Takizawa, R. Torii, H. Takagi, T.E. Tezduyar, and X.Y. Xu. Coronary arterial dynamics computation with medical-image-based time-dependent anatomical models and element-based zero-stress state estimates. *Computational Mechanics*, 54:1047–1053, 2014.
83. T. Tanaka, M. Morigami, and N. Atoda. Mechanism of resist pattern collapse during development process. *Japanese Journal of Applied Physics*, 32(12S):6059, 1993.
84. S. H. Tawfick, J. Bico, and S. Barcelo. Three-dimensional lithography by elasto-capillary engineering of filamentary materials. *MRS Bulletin*, 41(02):108–114, 2016.

85. 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(17–18):2002–2027, 2006.
86. S. Tremaine. On the origin of irregular structure in saturn's rings. *The Astronomical Journal*, 125(2):894, 2003.
87. S.E. Turner. Underwater implosion of glass spheres. *The Journal of the Acoustical Society of America.*, 121(2):844–852, 2007.
88. E. H. van Brummelen, M. Shokrpour-Roudbari, and G. J. van Zwieten. *Elasto-Capillarity Simulations Based on the Navier–Stokes–Cahn–Hilliard Equations*, pages 451–462. Springer International Publishing, Cham, 2016.
89. J.D. van der Waals. The thermodynamic theory of capillarity under the hypothesis of a continuous variation of density. *Journal of Statistical Physics*, 20(2):200–244, 1893.
90. C. Wang, M.C.H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs. Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis. *Computers and Fluids*, 142:3–14, 2017.
91. X. Wu, G. J. van Zwieten, and K. G. van der Zee. Stabilized second-order convex splitting schemes for Cahn-Hilliard models with application to diffuse-interface tumor-growth models. *International Journal for Numerical Methods in Biomedical Engineering*, 30(2):180–203, 2014.
92. J. Xu, G. Vilanova, and H. Gomez. A mathematical model coupling tumor growth and angiogenesis. *PloS one*, 11(2):e0149422, 2016.
93. L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21):2051–2067, 2004.

Immersogeometric Analysis of Bioprosthetic Heart Valves, Using the Dynamic Augmented Lagrangian Method

Ming-Chen Hsu and David Kamensky

Abstract In the mid-2010s, we began applying a combination of isogeometric analysis and immersed boundary methods to the problem of bioprosthetic heart valve (BHV) fluid–structure interaction (FSI). This chapter reviews how our research on BHV FSI (1) crystallized the emerging concept of *immersogeometric analysis*, (2) introduced a new semi-implicit numerical method for weakly enforcing constraints in time dependent problems, which we refer to as the *dynamic augmented Lagrangian* approach, and (3) clarified the important role of mass conservation in immersed FSI analysis. We illustrate these contributions with selected numerical results and discuss future improvements to, and applications of, the computational FSI techniques we have developed.

1 Introduction

Heart valves are passive anatomical structures driven by hemodynamic forces. They ensure proper unidirectional blood flow through the heart. At least 280,000 diseased valves are replaced annually [1, 2]. The most popular replacements are bioprosthetic heart valves (BHVs), fabricated from biologically derived materials [3]. Like native valves, BHVs consist of flexible leaflets. BHVs have more natural hemodynamics than the older "mechanical" prosthesis designs, which consist of rigid moving parts [2]. However, the lifespans of typical BHVs remain limited to ∼10–15 years, with structural deterioration mediated by fatigue and tissue mineralization [1, 2, 4, 5]. Much research has sought to prevent mineralization, but methods to

M.-C. Hsu (✉)
Department of Mechanical Engineering, Iowa State University, Ames, IA, USA
e-mail: jmchsu@iastate.edu

D. Kamensky
Department of Structural Engineering, University of California, San Diego, La Jolla, CA, USA
e-mail: dmkamensky@eng.ucsd.edu

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_5

extend durability remain less explored. Central to such efforts is an understanding of the stresses in BHV leaflets over the cardiac cycle.

Computational methods may be used for stress analysis of heart valves. Some previous computational studies on heart valve mechanics used (quasi-)static [6, 7] and dynamic [8] structural analysis, with assumed pressure loads. However, pure structural analysis is only accurate for static pressurization of a closed valve, which represents just part of the full cardiac cycle. It is therefore important to simulate the dynamics of heart valves interacting with blood, using computational fluid–structure interaction (FSI).

1.1 Computational FSI Analysis of Heart Valves

Heart valves present several challenges for FSI analysis. Most notably, the valve leaflets contact one another, changing the fluid subdomain's topology. This section updates the literature review of [3] to cover some additional recent work. Standard arbitrary Lagrangian–Eulerian (ALE) [9–11] or deforming-spatial-domain/space–time (DSD/ST) [12, 13] formulations, which continuously deform the fluid domain from some reference configuration, are no longer directly applicable. One must augment these methods with special techniques to handle extreme deformations. One solution is to generate a new mesh of finite elements or volumes for the fluid domain whenever its deformation becomes too extreme [14–17]. This allows computations to proceed, but introduces additional computational cost and numerical errors. Some recent work by Takizawa and collaborators [18] introduced a novel space–time with topology change (ST-TC) method that permits topology change without re-meshing. Takizawa et al. [19] applied the ST-TC approach to CFD analysis of a heart valve, and later extended the approach to include sliding interfaces in [20, 21], rendering it suitable for future full FSI analysis.

In light of the aforementioned difficulties, the majority of work to-date on heart valve FSI analysis has invoked Peskin's immersed boundary concept [22]. While it is not a universal convention, we follow [23–25] in applying the term "immersed boundary method" broadly, to describe any numerical method for approximating partial differential equations (PDEs) that allows boundaries of the PDE domain to cut arbitrarily through a computational mesh. Researchers may have varying interpretations of the term "immersed boundary method," and we recommend that writers clarify its meaning within a particular document.

Immersed boundary methods for FSI greatly simplify treatment of large structural deformations, but engender several disadvantages relative to ALE and DSD/ST techniques [26]. In particular, they struggle to efficiently capture boundary layer solutions near fluid–structure interfaces. Takizawa et al. [27] found that resolving such layers is essential to obtaining accurate shear stresses in hemodynamic analysis. A comprehensive overview of various immersed boundary methods and their properties is beyond the scope of this literature review; we refer the interested reader to [23, 24].

Peskin introduced the immersed boundary concept specifically to meet the demands of heart valve FSI analysis [22]. The numerical method proposed by

Peskin has found little if any direct application by bioengineers, though, due to its crude representation of the heart valve as a collection of markers connected by elastic fibers. However, deficient modeling of the structure subproblem is not an inherent feature of immersed boundary methods. In the early 2000s, [28–33] used an immersed boundary method introduced in [34] to couple finite element discretizations of heart valves and blood flow. This allowed investigation of various constitutive models, but numerical instabilities prevented analysis at realistic Reynolds numbers and transvalvular pressures. Increasing availability of parallel computing resources in the 2010s led to higher resolution simulations of heart valves. Griffith [35] adapted Peskin's original method to modern distributed-memory computer architectures and included adaptive mesh refinement for the fluid subproblem, to compute FSI of a native aortic valve throughout a full cardiac cycle, with physiological flow velocities and pressure differences. Borazjani [36] applied the curvilinear immersed boundary (CURVIB) method [37, 38] to simulate systolic ejection through a bioprosthetic aortic valve, using nearly 10 million grid points in the fluid domain. The valve leaflet models in the studies by Griffith and Borazjani suffered from deficiencies, though, with [35] modeling the leaflets in the style of Peskin, as markers connected by elastic fibers, and [36] omitting bending stiffness. The CURVIB method was recently extended to include fluid–shell structure interaction in [39, 40].

The immersed analyses cited above relied on academic research codes. As early as the late 1990s, immersed methods in the commercial software LS-DYNA [41] were used for FSI simulations of bioprosthetic and native aortic valves [42–45]. The time-explicit procedures used by LS-DYNA result in severe Courant–Friedrichs–Lewy conditions [46, 47], limiting stable time step size in hemodynamic computations, because blood is nearly incompressible. References [44, 45] circumvented this difficulty by artificially reducing the sound speed by a significant factor, reporting that the fluid density variations introduced by this deliberate modeling error were negligible. Other commercial analysis software for heart valve FSI analysis may be usable through "black box" coupling algorithms [48] that connect independent finite element analysis and CFD programs without access to their internal details. Specialized methods are required for stable and efficient black box coupling of fluids to thin, light structures such as heart valve leaflets [49, 50]. Astorino et al. [51] applied a novel black box coupling algorithm to FSI analysis of an idealized aortic valve. Remeshing functionality in ANSYS software has also recently allowed for boundary-fitted simulations of heart valves [52].

1.2 Immersogeometric Analysis

Following the majority of the studies cited in Sect. 1.1, our own work has employed an immersed approach to heart valve FSI analysis. The goal of immersed methods has always been to simplify the construction of analysis-suitable computational models from available geometric data specifying the domain of a PDE system. Traditional immersed boundary analysis eases this process by allowing subproblems to be discretized separately, then coupled through a numerical method.

Another technology for simplifying computational model generation is isogeometric analysis (IGA) [53]. IGA is based on the insight that many geometries in engineering design are specified in spline spaces that can be enriched, then used to approximate PDE solutions. These spline spaces also have desirable mathematical properties, including control over smoothness, improved approximation power [54], and straightforward constructions of discrete de Rham complexes [55, 56]. Benefits of these properties are evident in fluid and structural analyses, including studies of incompressible flow [57–60], thin shells [61–64], extreme mesh distortion [65], and contact [66, 67]. IGA encounters difficulties, though, when faced with realistic engineering designs. Foremost among these are:

1. Many designs of volumes are specified in terms of bounding spline surfaces. If analysts wish to solve PDEs in such volumes, then IGA, as originally conceived, is inapplicable.
2. Spline surfaces in designs are frequently trimmed along curves that do not conform to the parametric supports of the spline space's basis functions. The analysis space suggested by standard IGA is therefore not fitted to the boundaries of the PDE domain.

These challenges could be addressed by changing the way in which engineering products are designed: designers could transition to geometry representations that are analysis-suitable. Changing the habits of designers throughout industry, though, would require an incredible feat of mass persuasion. Undeterred, creators of analysis-suitable design technologies (e.g., [68–70]) have succeeded at incorporating their work into some major commercial design platforms. It remains doubtful, though, that analysis-suitable design will become standard any time soon. Further, many designs specified in past formats will remain relevant long into the future.

One way to make IGA practical without changing the design process is to incorporate immersed boundary methods. Difficulty 1 can be alleviated by creating a convenient unfitted analysis space covering the volume of interest, then using an immersed boundary method to enforce the desired boundary conditions on the spline surfaces. Difficulty 2 can be addressed by using the natural isogeometric solution space, and treating the trim curves as immersed boundaries. Promising work in both of these directions has been carried out using an immersed boundary approach called the finite cell method [71–75]. In addition to patching weaknesses of IGA, direct application of immersed boundary techniques to design geometries can eliminate the meshing and consequent geometrical approximation[1] from traditional immersed boundary analysis. In [77], we introduced the term *immersogeometric analysis* (IMGA) to describe this symbiotic union of immersed boundary and isogeometric technologies.[2]

[1] In practice, immersogeometric methods must frequently approximate integrals over the domain geometry, which may be considered a type of geometrical approximation [76, Sections 4.3 and 4.4], but this is conceptually distinct from the direct alteration of domain geometry that occurs in traditional mesh generation.

[2] The word "immersogeometric" was originally coined in 2014 by T. J. R. Hughes, while traveling in Italy; it is derived from the Italian word *immerso*, meaning "immersed."

1.3 Structure and Content of This Chapter

Section 2 states the coupled PDEs that we use to model the FSI system. Section 3 describes the isogeometric spatial discretizations for the fluid and structure subproblems. Section 4 completes the discretization with a semi-implicit coupling scheme that we call the *dynamic augmented Lagrangian* (DAL) method. Section 5 demonstrates the accuracy of the proposed methods, looking at both norm convergence and quantities of interest in nonlinear benchmark problems. Section 6 applies DAL-based IMGA to BHV FSI simulation and compares the results to in vitro experimental work. Finally, Sect. 7 sketches some future developments that may improve on the technology described in this chapter, connect it to clinical practice, and apply it to other FSI problems.

2 Mathematical Model of FSI

We model BHV leaflets as Kirchhoff–Love thin shells. We model the surrounding fluid as incompressible and Newtonian. The subproblems are coupled through kinematic and dynamic conditions on the fluid–solid interface. The thin structure is modeled geometrically as a 2D surface embedded in the 3D fluid domain. We state the model in a weak form, which is both suggestive of finite-dimensional approximations and conducive to including distributional forces from immersed boundaries.

Remark 1 We do not include a condition that the structure cannot intersect itself. Inclusion of such a constraint would be redundant in light of FSI kinematics, since a continuous velocity field is defined throughout the fluid–structure continuum [78]. While it is, in practice, useful to include some specialized treatment of structure-on-structure contact in a numerical method, we consider that a feature of the discretization, not the mathematical model.

2.1 Augmented Lagrangian Formulation of FSI

We start from the augmented Lagrangian framework for FSI introduced by Bazilevs et al. [79], and specialize to the case of thin structures. The region occupied by fluid at time t is $(\Omega_1)_t \subset \mathbb{R}^d$. The structure geometry at time t is modeled by the hypersurface $\Gamma_t \subset (\Omega_1)_t$. Let $\mathbf{u}_1$ denote the fluid's velocity and p denote its pressure. Let $\mathbf{y}$ denote the structure's displacement from some reference configuration, Γ_0, and $\mathbf{u}_2 \equiv \dot{\mathbf{y}}$ denote the velocity of the structure. The fluid–structure kinematic constraint, $\mathbf{u}_1 = \mathbf{u}_2$ on Γ_t, is enforced by the augmented Lagrangian

$$\int_{\Gamma_t} \boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma + \frac{1}{2} \int_{\Gamma_t} \beta |\mathbf{u}_1 - \mathbf{u}_2|^2\, d\Gamma\,, \tag{1}$$

where $\boldsymbol{\lambda}$ is a Lagrange multiplier and $\beta \geq 0$ is a penalty parameter. The resulting variational problem is: Find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$, $\mathbf{y} \in \mathcal{S}_d$, and $\boldsymbol{\lambda} \in \mathcal{S}_\ell$ such that, for all test functions $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, $\mathbf{w}_2 \in \mathcal{V}_d$, and $\delta\boldsymbol{\lambda} \in \mathcal{V}_\ell$

$$
\begin{aligned}
&B_1(\{\mathbf{u}_1, p\}, \{\mathbf{w}_1, q\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) \\
&\quad + \int_{\Gamma_t} \mathbf{w}_1 \cdot \boldsymbol{\lambda}\, d\Gamma + \int_{\Gamma_t} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \qquad (2) \\
&B_2(\mathbf{y}, \mathbf{w}_2) - F_2(\mathbf{w}_2) \\
&\quad - \int_{\Gamma_t} \mathbf{w}_2 \cdot \boldsymbol{\lambda}\, d\Gamma - \int_{\Gamma_t} \mathbf{w}_2 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \qquad (3) \\
&\int_{\Gamma_t} \delta\boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \qquad (4)
\end{aligned}
$$

where $\mathcal{S}_u$, $\mathcal{S}_p$, $\mathcal{S}_d$, and $\mathcal{S}_\ell$ are the trial solution spaces for the fluid velocity, fluid pressure, structural displacement, and interface Lagrange multiplier solutions. $\mathcal{V}_u$, $\mathcal{V}_p$, $\mathcal{V}_d$, and $\mathcal{V}_\ell$ are the corresponding test spaces. B_1, B_2, F_1, and F_2 are forms defining the (weak) fluid and structure subproblems.

2.2 Fluid Subproblem

The fluid subproblem is incompressible and Newtonian:

$$
\begin{aligned}
B_1(\{\mathbf{u}, p\}, \{\mathbf{w}, q\}; \hat{\mathbf{u}}) = &\int_{(\Omega_1)_t} \mathbf{w} \cdot \rho_1 \left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} \right) d\Omega \\
&+ \int_{(\Omega_1)_t} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1(\mathbf{u}, p)\, d\Omega + \int_{(\Omega_1)_t} q \nabla \cdot \mathbf{u}\, d\Omega \\
&- \gamma \int_{(\Gamma_{1\mathbf{h}})_t} \mathbf{w} \cdot \rho_1 \left\{ (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n}_1 \right\}_- \mathbf{u}\, d\Gamma\,, \qquad (5)
\end{aligned}
$$

$$
F_1(\{\mathbf{w}, q\}) = \int_{(\Omega_1)_t} \mathbf{w} \cdot \rho_1 \mathbf{f}_1\, d\Omega + \int_{(\Gamma_{1\mathbf{h}})_t} \mathbf{w} \cdot \mathbf{h}_1\, d\Gamma\,, \qquad (6)
$$

where ρ_1 is the fluid mass density, $\boldsymbol{\varepsilon}$ is the symmetric gradient operator, $\boldsymbol{\sigma}_1(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, where μ is the dynamic viscosity, $\mathbf{f}_1$ is a prescribed body force, and $\mathbf{h}_1$ is a prescribed traction on $\Gamma_{1\mathbf{h}} \subset \partial\Omega_1$. $(\Omega_1)_t$ deforms from some reference configuration, $(\Omega_1)_0$, according to the velocity field $\hat{\mathbf{u}}$, which need not equal $\mathbf{u}_1$. $\partial(\cdot)/\partial t|_{\hat{\mathbf{x}}}$ indicates time differentiation with respect to a fixed point $\hat{\mathbf{x}}$ from $(\Omega_1)_0$. The last term of (5) is not usually considered to be part of the weak Navier–Stokes

problem, but it enhances the stability of the problem in cases where flow enters through the Neumann boundary $\Gamma_{1\mathbf{h}}$ [80]. The function $\{\cdot\}_-$ isolates the negative part of its argument. The coefficient $\gamma \geq 0$ controls the strength of this stabilizing term and $\mathbf{n}_1$ is the outward-facing normal to Ω_1.

2.3 Thin Structure Subproblem

Following the Kirchhoff–Love thin shell kinematic hypotheses (see, e.g., [61, 62, 64]), B_2 and F_2 are defined as

$$B_2(\mathbf{y}, \mathbf{w}) = \int_{\Gamma_t} \mathbf{w} \cdot \rho_2 h_{\text{th}} \left.\frac{\partial^2 \mathbf{y}}{\partial t^2}\right|_{\mathbf{X}} d\Gamma + \int_{\Gamma_0} \int_{-h_{\text{th}}/2}^{h_{\text{th}}/2} D_{\mathbf{w}}\mathbf{E} : \mathbf{S}\, d\xi^3 d\Gamma \tag{7}$$

and

$$F_2(\mathbf{w}) = \int_{\Gamma_t} \mathbf{w} \cdot \rho_2 h_{\text{th}} \mathbf{f}_2\, d\Gamma + \int_{\Gamma_t} \mathbf{w} \cdot \mathbf{h}^{\text{net}}\, d\Gamma\,, \tag{8}$$

where ρ_2 is the structure mass density, $\mathbf{f}_2$ is a prescribed body force, h_{th} is the thickness of the shell, ξ^3 is a through-thickness coordinate, and we have referred the elasticity term to the reference configuration. $\mathbf{E}$ is the Green–Lagrange strain tensor [81, (2.67)] corresponding to the displacement $\mathbf{y}$, $D_{\mathbf{w}}\mathbf{E}$ is its functional derivative in the direction $\mathbf{w}$, and $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor [81, (3.63)], depending on $\mathbf{E}$. The last term of F_2 sums the prescribed tractions on the two sides of Γ_t. $\partial(\cdot)/\partial t|_{\mathbf{X}}$ indicates time differentiation with respect to a fixed material point. The Green–Lagrange strain $\mathbf{E}$ is simplified to depend entirely on the shell structure's midsurface displacement, $\mathbf{y} : \Gamma_0 \to \mathbb{R}^d$, using Kirchhoff–Love shell kinematic assumptions [61, 64], thus reducing the dimension of the PDE domain.

Any material model that accepts a Green–Lagrange strain $\mathbf{E}$ and returns a 2nd Piola–Kirchhoff stress $\mathbf{S}$ can be used directly in the structure subproblem defined above. In the work summarized by this chapter, we model BHV leaflets as hyperelastic, meaning that $\mathbf{S} = \partial\Psi/\partial\mathbf{E}$, where Ψ maps strains to energy densities [81, Chapter 6].

3 Discretization of Subproblems

Distinct fluid and structure subproblems may be isolated from the coupled problem stated in Sect. 2 by setting the test function corresponding to the other subproblem and the test function corresponding to the kinematic constraint to zero. Each of these subproblems may be discretized by adapting existing techniques for computational fluid and structural dynamics.

3.1 Fluid Subproblem

The fluid subproblem may be isolated by setting $\mathbf{w}_2 = \delta\boldsymbol{\lambda} = \mathbf{0}$, which yields (2), in which the structure velocity $\mathbf{u}_2$ and the Lagrange multiplier $\boldsymbol{\lambda}$ should be viewed as prescribed data. We describe two ways of discretizing this subproblem: the variational multiscale (VMS) approach[3] (Sect. 3.1.1) and the divergence-conforming B-spline approach (Sect. 3.1.2).

3.1.1 Variational Multiscale Formulation

Issues of discrete stability and turbulence modeling are simultaneously addressed by the variational multiscale (VMS) [83] formulation of [58]. In short, it substitutes an ansatz for subgrid velocities and pressures into the weak fluid subproblem. This ansatz is consistent with the strong form of the Navier–Stokes equations, so that the formulation smoothly transitions to high-order-accurate direct numerical simulation as approximation spaces are refined.

The mesh-dependent VMS formulation is posed on a collection of disjoint fluid elements $\{\Omega^e\}$ such that $\overline{\Omega_1} = \cup_e \overline{\Omega^e}$. $\{\Omega^e\}$, Ω_1, and Γ remain time-dependent, but, when there is no risk of confusion, we drop the subscript t to simplify notation. The superscript h indicates association with discrete spaces defined over these elements. The mesh $\{\Omega^e\}$ deforms with velocity $\hat{\mathbf{u}}^h$. Let $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ be discrete velocity and pressure spaces defined over $\{\Omega^e\}$. The semi-discrete VMS fluid subproblem is: Find $\mathbf{u}_1^h \in \mathcal{V}_u^h$ and $p^h \in \mathcal{V}_p^h$ such that, for all $\mathbf{w}_1^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$,

$$B_1^{\text{VMS}}(\{\mathbf{u}_1^h, p^h\}, \{\mathbf{w}_1^h, q^h\}; \hat{\mathbf{u}}^h) - F_1^{\text{VMS}}(\{\mathbf{w}_1^h, q^h\}) + \int_\Gamma \mathbf{w}_1^h \cdot (\lambda \mathbf{n}_2)\, d\Gamma + \int_\Gamma \mathbf{w}_1^h \cdot \beta(\mathbf{u}_1^h - \mathbf{u}_2)\, d\Gamma = 0\,, \tag{9}$$

where

$$B_1^{\text{VMS}}(\{\mathbf{u}, p\}, \{\mathbf{w}, q\}; \hat{\mathbf{u}}) = \int_{\Omega_1} \mathbf{w} \cdot \rho_1 \left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} \right) d\Omega + \int_{\Omega_1} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1\, d\Omega + \int_{\Omega_1} q \nabla \cdot \mathbf{u}\, d\Omega - \gamma \int_{\Gamma_{1\mathbf{h}}} \mathbf{w} \cdot \rho_1 \left\{ (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n}_1 \right\}_- \mathbf{u}\, d\Gamma$$

[3] We use of the term "VMS" in this chapter to refer to the specific VMS formulation explained in Sect. 3.1.1, applied to equal-order pressure–velocity discretizations. Our choice of terminology should not be taken to mean that the concept of VMS analysis is incompatible with div-conforming B-splines, which is demonstrably [82] not true.

$$
\begin{aligned}
&- \sum_e \int_{\Omega^e} \left((\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{w} + \frac{1}{\rho_1} \nabla q \right) \cdot \mathbf{u}' \, d\Omega \\
&- \sum_e \int_{\Omega^e} p' \nabla \cdot \mathbf{w} \, d\Omega \\
&+ \sum_e \int_{\Omega^e} \mathbf{w} \cdot (\mathbf{u}' \cdot \nabla \mathbf{u}) \, d\Omega \\
&- \sum_e \int_{\Omega^e} \frac{1}{\rho_1} \nabla \mathbf{w} : \left(\mathbf{u}' \otimes \mathbf{u}' \right) \, d\Omega \\
&+ \sum_e \int_{\Omega^e} \left(\mathbf{u}' \cdot \nabla \mathbf{w} \right) \overline{\tau} \cdot \left(\mathbf{u}' \cdot \nabla \mathbf{u} \right) \, d\Omega \, , \qquad (10)
\end{aligned}
$$

and

$$
F_1^{\mathrm{VMS}}(\{\mathbf{w}, q\}) = F_1(\{\mathbf{w}, q\}) \, . \qquad (11)
$$

The forms B_1^{VMS} and F_1^{VMS} are semi-discrete counterparts of B_1 and F_1. $\mathbf{u}'$ is the fine scale velocity ansatz,

$$
\mathbf{u}' = -\tau_{\mathrm{M}} \left(\rho_1 \left(\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\hat{\mathbf{x}}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma}_1 \right) \, , \qquad (12)
$$

and p' is the fine scale pressure,

$$
p' = -\rho_1 \tau_{\mathrm{C}} \nabla \cdot \mathbf{u} \, . \qquad (13)
$$

The stabilization parameters τ_{M}, τ_{C}, and $\overline{\tau}$ are defined as

$$
\tau_{\mathrm{M}} = \left(s \left(\frac{4}{\Delta t^2} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{G} (\mathbf{u} - \hat{\mathbf{u}}) + C_I \left(\frac{\mu}{\rho_1} \right)^2 \mathbf{G} : \mathbf{G} \right) \right)^{-1/2} \, , \qquad (14)
$$

$$
\tau_{\mathrm{C}} = (\tau_{\mathrm{M}} \mathrm{tr} \mathbf{G})^{-1} \, , \qquad (15)
$$

$$
\overline{\tau} = \left(\mathbf{u}' \cdot \mathbf{G} \mathbf{u}' \right)^{-1/2} \, , \qquad (16)
$$

where Δt is a timescale associated with the (currently unspecified) temporal discretization, C_I is a dimensionless positive constant derived from element-wise inverse estimates [84, 85], and $\mathbf{G}$ generalizes element size to physical elements mapped through $\mathbf{x}(\boldsymbol{\xi})$ from a parametric parent element: $G_{ij} = \xi_{k,i} \xi_{k,j}$. s is a dimensionless field such that, in most of Ω_1, $s = 1$, but, in an $\mathcal{O}(h)$ neighborhood of Γ, $s = s^{\mathrm{shell}} \geq 1$. We introduced this field in [77] to improve mass conservation near immersed boundaries. A theoretical motivation for this scaling is given in [86], and a numerical investigation of its effect is given in [77].

3.1.2 Divergence Conforming B-splines

A way to totally eliminate mass loss and obtain pointwise divergence-free velocity solutions is to discretize the fluid in a divergence-conforming (or div-conforming) manner, such that the divergence of every vector-valued function in the discrete velocity space is a member of the discrete pressure space. If this property is satisfied, then weak mass conservation implies strong (pointwise) mass conservation. A discretization of this type was developed for Stokes and Navier–Stokes flows by Evans and Hughes [59, 60, 87]. Evans and Hughes used B-splines to construct velocity and pressure spaces with the necessary properties, then directly posed the weak problem $B_1(\{\mathbf{u}_1^h, p^h\}, \{\mathbf{w}_1^h, q^h\}; \mathbf{0}) = F_1(\{\mathbf{w}_1^h, q^h\})$ over these discrete spaces. A caveat to the above reasoning is that, to truly obtain velocities that conform to the incompressibility constraint, one would need to solve the discrete algebraic problem *exactly*, which is impractical for realistic problems. We demonstrate in the 3D numerical examples of Sects. 6.2 and 6.3, however, that the benefits of divergence-conforming discretizations persist through common approximations in the assembly and solution of the algebraic problem.

Evans and Hughes used Nitsche's method to enforce no-slip boundary conditions. For the computations of this chapter, the regularity of the fluid velocity solution is at most $H^{3/2-\epsilon}(\Omega_1)$ and we use, for simplicity, a weakly consistent penalty method, altering the problem to be

$$B_1(\{\mathbf{u}_1, p\}, \{\mathbf{w}_1, q\}; \mathbf{0}) + C_{\text{pen}} \int_{\Gamma_{\text{pen}}} (\mathbf{u}_1 - \mathbf{g}) \cdot \mathbf{w}_1 \, d\Gamma = F_1(\{\mathbf{w}_1, q\}) \ , \qquad (17)$$

where $C_{\text{pen}} > 0$ is a penalty parameter and $\mathbf{g}$ is the desired velocity on $\Gamma_{\text{pen}} \subset \partial\Omega_1$.

Construction for Rectangular Domains

Suppose, for now, that Ω_1 is an axis-aligned d-dimensional rectangle. Then physical space can serve directly as a d-variate B-spline parameter space.[4] Define a d-variate scalar B-spline space for the pressure on Ω_1. Then, for $1 \leq i \leq d$, we can k-refine the pressure space once in the ith parametric direction to obtain a scalar space for the ith Cartesian velocity component. Due to well-known properties of B-splines under differentiation [88], the ith partial derivative of the ith velocity component will then be in the pressure space. The scalar basis functions of the velocity component spaces can be multiplied by their respective unit vectors to obtain a vector-valued basis for the discrete velocity space. The divergence of a velocity will therefore be a sum of d scalar functions in the pressure space.

[4] For readers unfamiliar with the construction and basic properties of B-splines, a comprehensive explanation can be found in [88].

Precise definitions are given in [59, Section 5.2]. In the notation of the cited reference, the velocity space is $\widehat{\mathcal{RT}}_h$ and the pressure space is $\widehat{\mathcal{W}}_h$. Following the terminology of [59], if the pressure space has polynomial degree k' in all directions, the entire pressure–velocity discretization is said to be of degree k', despite the presence of $(k'+1)$-degree polynomials in the velocity component spaces.

Generalization to Non-rectangular Domains

Div-conforming B-splines are not limited to rectangular domains. A point $\mathbf{X}$ in a rectangular parametric domain $\widehat{\Omega}$ may be mapped to a point $\mathbf{x}$ in a non-rectangular physical domain Ω by $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X})$. Vector-valued velocity basis functions defined on $\widehat{\Omega}$ are then pushed forward using the Piola transform. For arbitrary parametric-space velocity function $\widehat{\mathbf{u}}$, its pushforward $\mathbf{u}$ is

$$\mathbf{u}(\mathbf{x}) = \frac{1}{J(\mathbf{X})}\mathbf{F}(\mathbf{X})\widehat{\mathbf{u}}(\mathbf{X}) \,, \tag{18}$$

where

$$\mathbf{F} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}} \quad \Longleftrightarrow \quad F_{iJ} = \frac{\partial \phi_i}{\partial X_J} = \phi_{i,J} \,, \tag{19}$$

and $J = \det \mathbf{F}$. Using Nanson's formula [81, (2.54)] and integration by parts, we get the Piola identity

$$\operatorname{div} \mathbf{u} = \frac{1}{J}\operatorname{DIV} \widehat{\mathbf{u}} \,, \tag{20}$$

where

$$\operatorname{div} \mathbf{u} = \frac{\partial u_j}{\partial x_j} = u_{j,j} \quad \text{and} \quad \operatorname{DIV} \widehat{\mathbf{u}} = \frac{\partial \widehat{u}_B}{\partial X_B} = \widehat{u}_{B,B} \,. \tag{21}$$

We would like the divergence of every pushed-forward velocity function to exist in the pushed-forward pressure space. For every $\widehat{\mathbf{u}}$ in the parametric velocity space, there exists $\widehat{q}$ in the parametric pressure space such that $\widehat{q} = \operatorname{DIV} \widehat{\mathbf{u}}$. Then, recalling (20), the parametric pressure space function should be pushed forward by

$$q(\mathbf{x}) = \frac{1}{J(\mathbf{X})}\widehat{q}(\mathbf{X}) \,. \tag{22}$$

Div-confomring B-splines may be used on even wider classes of geometries by joining deformed rectangular patches together with a discontinuous Galerkin approach [59, Section 6.5].

Stabilizing Advection

The Galerkin discretization used by Evans and Hughes can be straightforwardly augmented to include SUPG stabilization [89]. However, the pressure gradient in the momentum equation residual removes the property of the Galerkin approximation that the error in the velocity solution is independent of pressure interpolation error [60, (6.32)]. This property is valuable in the presence of immersed boundaries that induce large discontinuities in the exact pressure solution. In this work, we stabilize div-conforming discretizations with $\mathcal{O}(h)$ streamline diffusion: we add

$$+\sum_e (\tau\rho_1 \mathbf{u}_1^h \cdot \nabla \mathbf{u}_1^h, \mathbf{u}_1^h \cdot \nabla \mathbf{w}_1^h)_{L^2(\Omega^e)} \tag{23}$$

to $B_1(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\})$, where $\{\Omega^e\}_{e=1}^{N_{\text{el}}}$ are the N_{el} Bézier elements of the B-spline mesh and

$$\tau = \begin{cases} \left(\mathbf{u}_1^h \cdot \mathbf{G}\mathbf{u}_1^h\right)^{-1/2} & \mathbf{u}_1^h \cdot \mathbf{G}\mathbf{u}_1^h > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{24}$$

While this is only weakly consistent, we do not expect high convergence rates from immersed boundary discretizations of the type considered here, due to low regularity of the exact solution.

3.2 Structure Subproblem

Setting $\mathbf{w}_1 = \delta\boldsymbol{\lambda} = \mathbf{0}$ isolates the structure subproblem (3), in which $\mathbf{u}_2$ and $\boldsymbol{\lambda}$ are considered prescribed data. This problem can be stably discretized using a Bubnov–Galerkin method. However, for $B_2(\mathbf{y}, \mathbf{w})$ to remain bounded, $\mathbf{y}$ and $\mathbf{w}$ need to be in $H^2(\Gamma)$. It is sufficient for discrete spaces to be in $C^1(\Gamma)$. Traditional finite element spaces do not satisfy this requirement. However, isogeometric spline spaces can be made C^1 if geometry allows. Typical BHV leaflet geometries can be accurately modeled by C^1 spline surfaces, so, for the purposes of this chapter, the semidiscrete structure subproblem amounts to choosing $\mathcal{V}_y$ in (3) to be (and enrichment of) the smooth spline space used to model the geometry. The implementation of such discretizations is documented exhaustively in [64]. We augment this discretization with penalty-based contact, as described in [77], as mentioned in Remark 1.

4 Dynamic Augmented Lagrangian Coupling

The augmented Lagrangian coupling the subproblems is discretized using a semi-implicit time integration scheme, in which the penalty is treated implicitly and the Lagrange multiplier is updated explicitly. We call this the *dynamic augmented*

Lagrangian (DAL) method. DAL circumvents difficulties with fully implicit coupling, while forbidding leakage through the structure in steady-state solutions and retaining the stability that eludes fully explicit approaches.

4.1 Separation of Normal and Tangential Coupling

The constraint that $\mathbf{u}_1 = \mathbf{u}_2$ on Γ can be separated into no-penetration

$$\mathbf{u}_1 \cdot \mathbf{n}_2 = \mathbf{u}_2 \cdot \mathbf{n}_2 \tag{25}$$

and no-slip

$$\mathbf{u}_1 - (\mathbf{u}_1 \cdot \mathbf{n}_2)\,\mathbf{n}_2 = \mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{n}_2)\,\mathbf{n}_2\,, \tag{26}$$

where $\mathbf{n}_2$ is normal to Γ. These constraints are enforced by normal and tangential components of $\boldsymbol{\lambda}$.

No-penetration is critical to the qualitative structure of solutions. No-slip is less essential, and strong enforcement may even be detrimental to solution quality on coarse meshes [90–94]. We therefore discretize these constraints differently. For no-penetration, we discretize a scalar multiplier field, $\lambda = \boldsymbol{\lambda} \cdot \mathbf{n}_2$. For no-slip, we approximate the tangential component of $\boldsymbol{\lambda}$ by a weakly consistent penalty force. Because Γ_t can cut through the fluid domain in arbitrary ways, we do not attempt to construct inf-sup stable combinations of velocity and multiplier spaces. Instead, we circumvent the inf-sup condition by regularizing the no-penetration constraint residual:

$$(\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 \quad \rightarrow \quad (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 - \frac{r}{\beta}\lambda\,, \tag{27}$$

where $r \geq 0$ is a dimensionless constant. Much as the slip penalization can be derived as a degenerate case of Nitsche's method [77, Section 4.1], the regularization of the no-penetration constraint can be viewed as a degenerate case of strongly consistent Barbosa–Hughes stabilization [95].

The problem we discretize in time is then: Find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$, $\mathbf{y} \in \mathcal{S}_d$, and $\lambda \in \mathcal{S}_\ell$ such that, for all test functions $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, $\mathbf{w}_2 \in \mathcal{V}_d$, and $\delta\lambda \in \mathcal{V}_\ell$

$$\begin{aligned} &B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{y}) - F_2(\mathbf{w}_2) \\ &\quad + \int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \lambda \mathbf{n}_2 \, d\Gamma \\ &\quad + \int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \tau^B_{\mathrm{NOR}} \left((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2\right) \mathbf{n}_2 \, d\Gamma \end{aligned}$$

$$+\int_{\Gamma_t}(\mathbf{w}_1-\mathbf{w}_2)\cdot\tau^B_{\mathrm{TAN}}\left((\mathbf{u}_1-\mathbf{u}_2)-((\mathbf{u}_1-\mathbf{u}_2)\cdot\mathbf{n}_2)\,\mathbf{n}_2\right)\,d\Gamma$$

$$+\int_{\Gamma_t}\delta\lambda\cdot\left((\mathbf{u}_1-\mathbf{u}_2)\cdot\mathbf{n}_2-\frac{r\lambda}{\tau^B_{\mathrm{NOR}}}\right)\,d\Gamma\;=\;0, \tag{28}$$

where we split the penalty term into normal and tangential components. We propose to scale the tangential penalty like

$$\tau^B_{\mathrm{TAN}}=C_{\mathrm{TAN}}\frac{\mu}{h}\,, \tag{29}$$

where C_{TAN} is a dimensionless $\mathcal{O}(1)$ constant and h is a measure of the fluid element diameter, with units of length. We propose that the normal penalty scale like

$$\tau^B_{\mathrm{NOR}}=\max\left\{C^{\mathrm{inert}}_{\mathrm{NOR}}\frac{\rho_1 h}{\Delta t},C^{\mathrm{visc}}_{\mathrm{NOR}}\frac{\mu}{h}\right\}\,, \tag{30}$$

where $C^{\mathrm{inert}}_{\mathrm{NOR}}$ and $C^{\mathrm{visc}}_{\mathrm{NOR}}$ are dimensionless constants and Δt is a time scale from the temporal discretization.

4.2 Time Integration Algorithm

We now state the time-marching procedure for the coupled system. The algorithm computes approximate solutions discrete time levels, indexed by n and separated by steps of size Δt. At time level n, the discrete fluid velocity is defined by a coefficient vector $\mathbf{U}^n$, the fluid time derivative by $\dot{\mathbf{U}}^n$, the fluid pressure by $\mathbf{P}^n$, and the structure displacement, velocity, and acceleration by $\mathbf{Y}^n$, $\dot{\mathbf{Y}}^n$, and $\ddot{\mathbf{Y}}^n$. The multiplier at level n is λ^n, considered a function with domain Γ_t, and represented discretely as a set of samples at quadrature points of a (Lagrangian) integration rule on Γ_t. Consider solution variables at level n known. The first step of DAL is to construct a system of equations for all $(n+1)$-level fluid and structure unknowns, with λ^{n+1} kept equal to λ^n:

$$\mathrm{Res}\left(\mathbf{U}^{n+\alpha_f},\dot{\mathbf{U}}^{n+\alpha_m},\mathbf{Y}^{n+\alpha_f},\dot{\mathbf{Y}}^{n+\alpha_f},\ddot{\mathbf{Y}}^{n+\alpha_m},\mathbf{P}^{n+1},\lambda^{n+1}(=\lambda^n)\right)=\mathbf{0}\,, \tag{31}$$

$$\mathbf{U}^{n+1}=\mathbf{U}^n+\Delta t\left((1-\gamma)\dot{\mathbf{U}}^n+\gamma\dot{\mathbf{U}}^{n+1}\right)\,, \tag{32}$$

$$\dot{\mathbf{U}}^{n+\alpha_m}=\dot{\mathbf{U}}^n+\alpha_m\left(\dot{\mathbf{U}}^{n+1}-\dot{\mathbf{U}}^n\right)\,, \tag{33}$$

$$\mathbf{U}^{n+\alpha_f}=\mathbf{U}^n+\alpha_f\left(\mathbf{U}^{n+1}-\mathbf{U}^n\right)\,, \tag{34}$$

$$\mathbf{Y}^{n+1}=\mathbf{Y}^n+\Delta t\dot{\mathbf{Y}}^n+\frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{Y}}^n+2\beta\ddot{\mathbf{Y}}^{n+1}\right)\,, \tag{35}$$

$$\dot{\mathbf{Y}}^{n+1} = \dot{\mathbf{Y}}^{n} + \Delta t \left((1-\gamma)\ddot{\mathbf{Y}}^{n} + \gamma \ddot{\mathbf{Y}}^{n+1} \right) , \tag{36}$$

$$\ddot{\mathbf{Y}}^{n+\alpha_m} = \ddot{\mathbf{Y}}^{n} + \alpha_m \left(\ddot{\mathbf{Y}}^{n+1} - \ddot{\mathbf{Y}}^{n} \right) , \tag{37}$$

$$\dot{\mathbf{Y}}^{n+\alpha_f} = \dot{\mathbf{Y}}^{n} + \alpha_f \left(\dot{\mathbf{Y}}^{n+1} - \dot{\mathbf{Y}}^{n} \right) , \tag{38}$$

$$\mathbf{Y}^{n+\alpha_f} = \mathbf{Y}^{n} + \alpha_f \left(\mathbf{Y}^{n+1} - \mathbf{Y}^{n} \right) , \tag{39}$$

where α_m, α_f, β, and γ are time integration parameters. Res(...) is the algebraic residual corresponding to the discretization of (28) with $\delta\lambda = 0$. This penalty-coupled problem is resolved by block iteration, which alternates between solving for fluid and structure increments, as described further in Sect. 4.3. Equations (31)–(39) are based on the generalized-α method [96]. Following [97, Section 4.4], we work within a subset of generalized-α methods, parameterized by $\rho_\infty \in [0, 1]$, which controls numerical damping and defines the free parameters as

$$\alpha_m = \frac{1}{2}\left(\frac{3-\rho_\infty}{1+\rho_\infty} \right) , \alpha_f = \frac{1}{1+\rho_\infty} , \tag{40}$$

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f , \beta = \frac{1}{4}\left(1 + \alpha_m - \alpha_f\right)^2 . \tag{41}$$

The backward Euler method can also be conveniently implemented within the generalized-α predictor–multi-corrector scheme of [97] by setting the generalized-α parameters to $\alpha_m = \alpha_f = \gamma = \beta = 1$ and modifying the displacement predictor.

Note that, because the multiplier is fixed in (31)–(39), the (regularized) α-level constraint residual

$$R^{n+\alpha} = \left(\left(\mathbf{u}_1^h \right)^{n+\alpha_f} - \left(\mathbf{u}_2^h \right)^{n+\alpha_f} \right) \cdot \mathbf{n}_2^{n+\alpha_f} - \frac{r\lambda^{n+1}}{\tau^B_{\mathrm{NOR}}} \tag{42}$$

is not necessarily zero on $\Gamma_{t+\alpha_f}$. To motivate the development of the multiplier update step in DAL, consider the case of $r = 0$. If $R^{n+\alpha} = 0$ and $r = 0$, then the normal component of the α-level penalty force, $\tau^B_{\mathrm{NOR}} R^{n+\alpha}$, will be zero and the normal α-level fluid–structure force will be due only to the Lagrange multiplier, λ^{n+1}. This suggests the explicit update

$$\lambda^{n+1} \leftarrow \lambda^{n+1} + \tau^B_{\mathrm{NOR}} R^{n+\alpha} , \tag{43}$$

in which λ^{n+1} is set equal to the α-level fluid–structure forcing. Equations (31)–(39) are of course no longer satisfied with the updated λ^{n+1}, but one may attempt to iterate the steps

1. Solve (31)–(39) with λ^{n+1} fixed.
2. Update λ^{n+1} by (43).

until $\|R^{n+\alpha}\|_{L^2(\Gamma_I)}$ is converged to some tolerance. As explained in [77, Section 4.2.1], the $r = 0$ case of this iteration corresponds to the classic augmented Lagrangian algorithm of [98, 99]. For $r = 0$, though, the convergence criterion of $\|R^{n+\alpha}\|_{L^2(\Gamma_I)} < \epsilon$ is too strict to arrive at a non-locking solution; it effectively demands pointwise constraint satisfaction between non-matching velocity spaces of the fluid and structure. We found, accordingly, that the iteration does not typically converge, but circumvented this difficulty by truncating to a single pass, leading to the semi-implicit time marching scheme of first solving (31)–(39) with $\lambda^{n+1} = \lambda^n$, then updating λ^{n+1} by (44) and continuing directly to the next time step, i.e.,

$$\lambda^{n+1} = \lambda^n + \tau^B_{\text{NOR}} R^{n+\alpha} . \tag{44}$$

This augmented-Lagrangian-based explicit multiplier update is the distinguishing feature of DAL. Use of $r = 0$ is effective for transient problems, but may run into difficulties in the steady limit, when the Lagrange multiplier and velocity discrete spaces are not chosen stably. Choosing $r > 0$ can improve robustness. In that case, (44) is an implicit formula, but it can be recast in explicit form:

$$\lambda^{n+1} = \frac{\lambda^n + \tau^B_{\text{NOR}} \left(\left(\mathbf{u}^h_1\right)^{n+\alpha_f} - \left(\mathbf{u}^h_2\right)^{n+\alpha_f} \right) \cdot \mathbf{n}_2^{n+\alpha_f}}{1 + r} . \tag{45}$$

Some caution is warranted, however, in perturbing the kinematic constraint. Section 4.4.4 provides an illustrative example of the effects of this consistency error.

4.3 Block Iterative Solution of the Implicit Problem

The implicit step of DAL amounts to a penalty regularization of fluid–structure coupling, with a prescribed loading $\lambda^n \mathbf{n}^{n+\alpha_f}$ along $\Gamma^{n+\alpha_f}$. The penalty value can be moderate, rendering the regularized problem much easier to solve. A simple block-iterative procedure turns out to be practical, even for applications with light structures and heavy, incompressible fluids.

Consider $R_{\text{f}}(u_{\text{f}}, u_{\text{s}})$ to be the nonlinear residual for the fully discrete fluid subproblem at a particular time step, which depends on discrete fluid and structure solutions, u_{f} and u_{s}. Likewise, $R_{\text{s}}(u_{\text{f}}, u_{\text{s}})$ is the residual for the discrete structure subproblem. The block-iterative procedure to find a root of $(R_{\text{f}}, R_{\text{s}})$ is to start with guesses for u_{f} and u_{s} and repeat

1. Assemble $R_{\text{f}}(u_{\text{f}}, u_{\text{s}})$ and a(n approximate) tangent matrix, $A_{\text{f}} \approx \partial R_{\text{f}}/\partial u_{\text{f}}$.
2. Solve the linear system $A_{\text{f}} \Delta u_{\text{f}} = -R_{\text{f}}$ for the fluid solution increment.
3. Update the fluid solution: $u_{\text{f}} \leftarrow u_{\text{f}} + \Delta u_{\text{f}}$.
4. Assemble $R_{\text{s}}(u_{\text{f}}, u_{\text{s}})$ and $A_{\text{s}} \approx \partial R_{\text{s}}/\partial u_{\text{s}}$.
5. Solve $A_{\text{s}} \Delta u_{\text{s}} = -R_{\text{s}}$ for the structure solution increment.
6. Update the structure solution: $u_{\text{s}} \leftarrow u_{\text{s}} + \Delta u_{\text{s}}$.

until R_f and R_s are sufficiently converged. To ensure predictable run-times and avoid stagnation in pathological configurations, we typically choose a fixed number of iterations rather than a convergence tolerance. While it is possible that error from isolated, poorly solved time steps can pollute the future of an unsteady solution, we find this *ad hoc* procedure effective for predicting quantities of engineering interest.

4.4 Discussion

We summarize here some alternate interpretations and qualitative analysis from [77, 100] of the algorithm stated in Sect. 4.2.

4.4.1 Modified Equation Interpretation of DAL

When $r = 0$, the multiplier becomes an accumulation of penalty tractions from previous time steps. This is equivalent to replacing the multiplier and normal penalty terms

$$\int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot (\lambda \mathbf{n}_2)\, d\Gamma + \int_{\Gamma_t} ((\mathbf{w}_1 - \mathbf{w}_2) \cdot \mathbf{n}_2)\, \tau_{\mathrm{NOR}}^B\, ((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2)\; d\Gamma \tag{46}$$

by a penalization of (a backward Euler approximation of) the time integral of pointwise normal velocity differences on the immersed surface Γ_t

$$\int_{\Gamma_t} \left\{ \frac{\tau_{\mathrm{NOR}}^B}{\Delta t} (\mathbf{w}_1(\mathbf{x}, t) - \mathbf{w}_2(\mathbf{x}, t)) \cdot \mathbf{n}_2(\mathbf{x}, t) \right. \int_0^t \Big(\mathbf{u}_1(\boldsymbol{\varphi}_\tau(\boldsymbol{\varphi}_t^{-1}(\mathbf{x})), \tau) - \mathbf{u}_2(\boldsymbol{\varphi}_\tau(\boldsymbol{\varphi}_t^{-1}(\mathbf{x})), \tau) \Big) \left. \cdot \mathbf{n}_2(\boldsymbol{\varphi}_\tau(\boldsymbol{\varphi}_t^{-1}(\mathbf{x})), \tau)\, d\tau \right\}\; d\Gamma\,, \tag{47}$$

where $\boldsymbol{\varphi}_\tau(\mathbf{X})$ gives the spatial position at time τ of material point $\mathbf{X} \in \Gamma_0$ and $d\Gamma$ indicates integration over $\mathbf{x} \in \Gamma_t$. To see this, first define (at fixed $\mathbf{X}$)

$$\lambda^{\mathrm{reg}}(t) = \frac{\tau_{\mathrm{NOR}}^B}{\Delta t} \int_0^t (\mathbf{u}_1(\tau) - \mathbf{u}_2(\tau)) \cdot \mathbf{n}_2(\tau)\, d\tau\,. \tag{48}$$

Then

$$(\dot{\lambda^{\mathrm{reg}}}) = \left.\frac{\partial \lambda^{\mathrm{reg}}}{\partial t}\right|_{\mathbf{X}} = \frac{\tau_{\mathrm{NOR}}^B}{\Delta t} (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2\,. \tag{49}$$

The normal forcing on Γ in the implicit step of the semi-implicit time integrator is

$$\left(\lambda^{\text{reg}}\right)^{n+1} = \left(\lambda^{\text{reg}}\right)^{n} + \Delta t \left(\dot{\lambda^{\text{reg}}}\right)^{n+1} \tag{50}$$

where $(\lambda^{\text{reg}})^n$ is a sum of all previous approximations of λ and $\Delta t (\dot{\lambda^{\text{reg}}})^{n+1}$ is the current time step's penalty forcing. Thus the forcing (47) is accounted for in a fully implicit manner, using the stable backward Euler method.

For $r > 0$, we can draw a similar analogy, in which λ^{reg} advances through time by backward Euler integration of

$$\frac{1}{(1+r)} \left.\frac{\partial \lambda^{\text{reg}}}{\partial t}\right|_{\mathbf{X}} = \frac{\tau^B_{\text{NOR}}}{\Delta t} (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 - \frac{r}{\Delta t (1+r)} \lambda^{\text{reg}} . \tag{51}$$

Intuitively, the additional term causes a decay of λ^{reg} in the absence of constraint violation, which highlights its stabilizing effect.

4.4.2 Analogy to Artificial Compressibility

The differential equation given in (49) closely resembles the method of artificial compressibility [101]. In that scheme, the approximated Lagrange multiplier p representing pressure in an incompressible flow evolves in an analogous way to λ^{reg} (in the case $r = 0$):

$$\partial_t p = -\frac{1}{\delta} \nabla \cdot \mathbf{u}_1 , \tag{52}$$

where the constraint is $\nabla \cdot \mathbf{u}_1 = 0$ (instead of $(\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 = 0$), $1/\delta$ is the penalty parameter. A physical interpretation of DAL for FSI, similar to Chorin's original formulation of (52) in terms of a fictitious density variable, is that, for $r = 0$, DAL penalizes displacement of the fluid through the structure. This interpretation makes clear how penalizing the time integral of velocity prevents the steady creep of flow through a barrier.

4.4.3 Relation to Feedback Boundary Conditions

The time-continuous interpretation of DAL with $r = 0$ may be interpreted as a special case of an existing framework for enforcing Dirichlet boundary conditions on the unsteady Navier–Stokes equation. Goldstein et al. [102] proposed to apply concentrated surface forcing of the form [102, (3)]

$$\mathbf{f}(\mathbf{x}_s, t) = \alpha \int_0^t \mathbf{u}_1(\mathbf{x}_s, \tau) \, d\tau + \beta \mathbf{u}_1(\mathbf{x}_s, t) , \tag{53}$$

for all $\mathbf{x}_s$ on a stationary solid boundary with parameters $\alpha \leq 0$ and $\beta \leq 0$. Goldstein et al. interpreted this method, which we refer to here as the feedback method, in the context of control theory, arguing heuristically that it provides negative feedback in the case of constraint violation.

The initial implementation of [102] used a spectral fluid discretization and applied smoothing to filter concentrated forces, reducing pollution effects from the global nature of the spectral basis functions (cf. [103, Chapter I, Section 2]). Goldstein and collaborators continue to use this methodology for DNS of turbulent flows [104–109]. Saiki and Biringen [110, 111] extended the concept of feedback forcing to finite difference fluid discretizations, using bilinear interpolation within grid cells to evaluate velocity at quadrature points of the immersed boundary and also to distribute concentrated feedback forces to grid points. Reference [110] was the first application of the approach to moving boundaries, in which (53) becomes (cf. [110, (1)] and (47))

$$\begin{aligned}\mathbf{f}(\boldsymbol{\varphi}_t(\mathbf{X}_s), t) = \alpha \int_0^t \left(\mathbf{u}_1(\boldsymbol{\varphi}_\tau(\mathbf{X}_s), \tau) - \mathbf{U}_2(\mathbf{X}_s, \tau)\right)\, d\tau \\ + \beta \left(\mathbf{u}_1(\boldsymbol{\varphi}_\tau(\mathbf{X}_s), \tau) - \mathbf{U}_2(\mathbf{X}_s, \tau)\right)\,,\end{aligned} \tag{54}$$

where $\boldsymbol{\varphi}_t(\mathbf{X}_s)$ represents the position at time t of a material point $\mathbf{X}_s$ on the moving boundary with velocity $\mathbf{U}_2(\mathbf{X}_s, t)$. This naturally leads to FSI, and a recent series of papers [112–116] demonstrated that feedback forcing is robust and accurate for simulating light, flexible, immersed structures. A related approach has been used in the commercial code LS-DYNA [41] for decades, to study automobile airbag inflation and other challenging FSI problems [117–120], including heart valve simulation [42–45]. We have seen no documentation explicitly relating it to the feedback method, and assume that it was devised independently. The repeated rediscovery of this formulation by engineers studying difficult CFD and FSI problems suggests an inherent robustness to the approach.

4.4.4 Qualitative Effects of Multiplier Stabilization

The case of $r > 0$ is less physically intuitive than the $r = 0$ case. To provide some intuition for the influence of r, consider a model of plug flow through a blocked tube: a rigid barrier blocks a channel with slip boundaries, filled with a fluid assumed to have a velocity, $u\mathbf{e}_1$, that is constant across space, but may vary with time.

The ends of the channel are subject to pressures P_1 and P_2, which define the pressure drop, $\Delta P = P_1 - P_2$. Suppose that the Lagrange multiplier field takes on a single constant value across the barrier. Then the steady solution of the semi-implicit time integration procedure described in Sect. 4.2 reduces to the conditions

1. Steadiness: $\lambda^{n+1} = \lambda^n = \lambda^\infty \;\Rightarrow\; \lambda^\infty = \left(\lambda^\infty + \tau^B_{\text{NOR}} u\right)/(1 + r)$.
2. Equilibrium: $\lambda^\infty + \tau^B_{\text{NOR}} u = \Delta P$.

Leakage is then given by $u = \frac{r\Delta P}{\tau^B_{\text{NOR}}(1+r)}$, which asymptotes to inverse scaling with the penalty parameter as $r \to \infty$ and to zero as $r \to 0$. For fixed $r > 0$, steady leakage converges to zero with refinement at the same rate as it would for a pure penalty method, but, if r is an adjustable parameter, one may reduce the steady-state leakage arbitrarily without impacting the solvability of the discrete problem at each time step.

5 Numerical Experiments

We demonstrate, through numerical experiments, that the DAL method is convergent. We summarize here results from [77, 86], considering both convergence of solutions in Sobolev norms for simple problems and convergence of quantities of interest in more complicated problems.

5.1 Navier–Stokes Flow with Immersed Boundaries

Consider, first, Navier–Stokes flow with Dirichlet conditions on immersed boundaries.

5.2 Taylor–Green Vortex

The Taylor–Green vortex is a solution to the 2D Navier–Stokes equations posed on the domain $\Omega = [-\pi, \pi]^2$ with periodic boundary conditions and no external forcing:

$$\mathbf{u}_{\text{TG}}(\mathbf{x}, t) = (\sin(x_1)\cos(x_2)\mathbf{e}_1 - \cos(x_1)\sin(x_2)\mathbf{e}_2)\, e^{-2\mu t/\rho}\,. \tag{55}$$

We construct an interesting test problem by prescribing $\mathbf{u} = \mathbf{u}_{\text{TG}}$ as an initial condition at $t = 0$ and also as a time-dependent Dirichlet boundary condition on a closed immersed boundary Γ, then adding a body force $\mathbf{f}_x = \mathbf{e}_1$. The body force induces a pressure gradient in the region enclosed by Γ without perturbing the velocity solution there. The velocity outside of the region enclosed by Γ is no longer equal to $\mathbf{u}_{\text{TG}}$ for $t > 0$. There are jumps in the pressure and velocity derivatives along Γ, so the regularity of the velocity solution is representative of typical applications. We have not derived an exact solution on the entire domain, but one can easily measure error in a subset Ω^{err} of the region enclosed by Γ. In this section, we consider low Reynolds number flow, and choose $\mu = 0.01$. A high Reynolds-number test is carried out in Sect. 5.2.2.

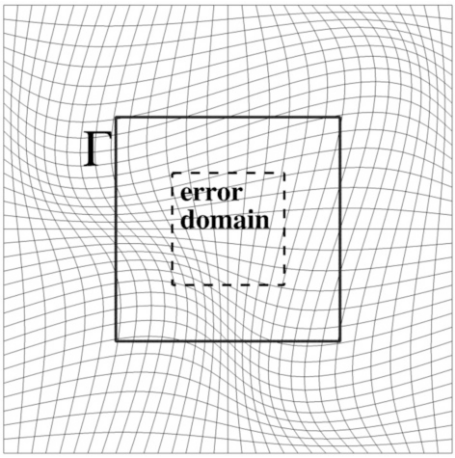


Fig. 1 The non-rectilinear mesh of Ω avoids grid alignment with Γ

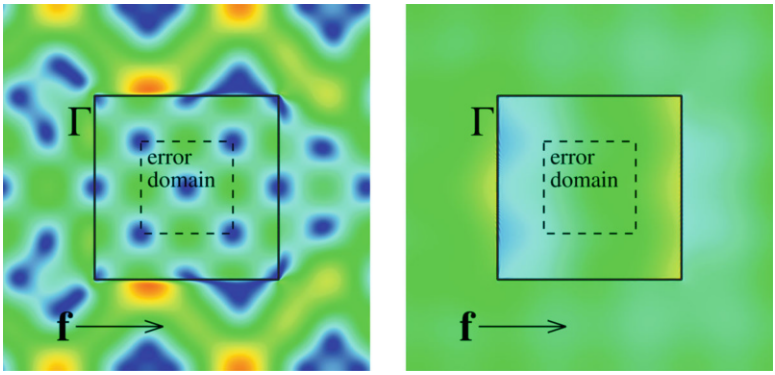


Fig. 2 Simultaneous velocity magnitude (left) and pressure (right) snapshots of the Navier–Stokes Taylor–Green problem, with annotations describing the problem setup

We choose $\Gamma = \partial\left((-\pi, \pi)^2\right)$. To avoid special behavior associated mesh-aligned immersed boundaries, we distort the background mesh, as shown in Fig. 1. Figure 2 illustrates the problem setup. Div-conforming B-splines of degree $k' = 1$ are used to discretize the velocity and pressure spaces, and backward Euler integration is applied in time. Error convergence is shown in Fig. 3, displaying nearly first-order rates.

5.2.1 Translating Taylor–Green Vortex

Adding a uniform velocity to an initial condition in a periodic domain yields a Galilean transformation of the original solution. In this section, we add $\mathbf{v} = -0.87\mathbf{e}_1 - 0.5\mathbf{e}_2$ to the initial condition of the Taylor–Green vortex and translate

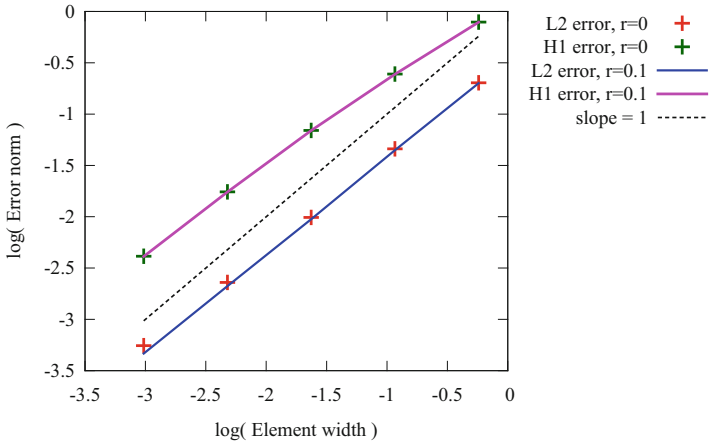


Fig. 3 Convergence of the $L^2(\Omega^{\text{err}})$ and $H^1(\Omega^{\text{err}})$ errors for $r = 0$ and $r = 0.1$ for Navier–Stokes flow with a stationary boundary and positive viscosity

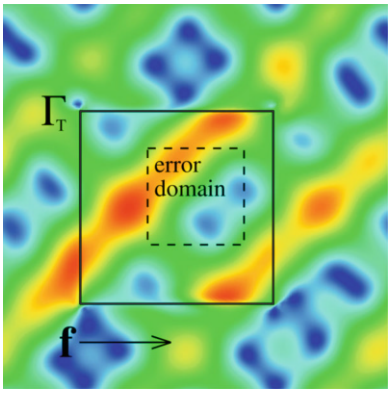


Fig. 4 Annotated snapshot of velocity magnitude at time T for Navier–Stokes flow with moving boundaries and positive viscosity

Γ_t at the same velocity. The solution at time T is shown in Fig. 4. Figure 5 indicates that convergence on Ω^{err} remains intact.

5.2.2 Infinite Reynolds Number

To demonstrate the robustness at realistic Reynolds numbers, we repeat the test of Sect. 5.2.1 with $\mu = 0$. The exact solution becomes tangentially discontinuous at Γ_t. This behavior is captured reasonably well, as shown in Fig. 6. The nearly linear convergence rates in $L^2(\Omega^{\text{err}})$ and $H^1(\Omega^{\text{err}})$ are maintained, as shown in Fig. 7, despite the fact that the $H^1(\Omega)$ norm of the exact solution is not well-defined. This example uses $r = 0$, for reasons explained in [86].

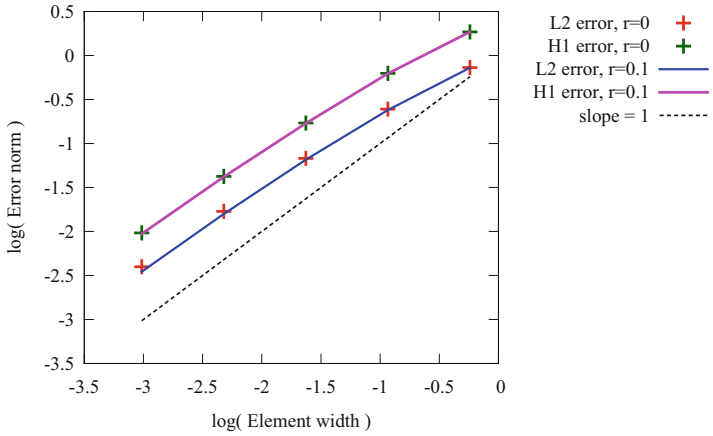


Fig. 5 Convergence of the $L^2(\Omega^{\text{err}})$ and $H^1(\Omega^{\text{err}})$ errors for $r = 0$ and $r = 0.1$ for Navier–Stokes flow with moving boundaries

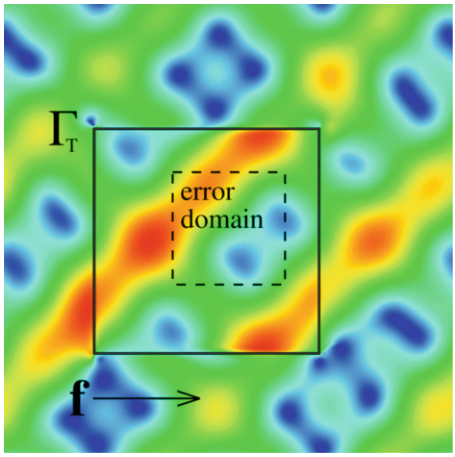


Fig. 6 Annotated snapshot of velocity magnitude at time T for Navier–Stokes flow with moving boundaries and zero viscosity

5.3 2D Non-coapting Valve

This section considers a 2D valve-inspired benchmark problem investigated previously by Gil et al. [121], Hesch et al. [122], Wick [123], and Kadapa et al. [124]. The structure does not contact itself, so it is straightforward to compute converged solutions using verified body-fitted methods, making the problem a valuable benchmark for new immersed approaches.

5.3.1 Description of the Problem

The problem consists of two cantilevered elastic beams immersed in a 2D channel filled with incompressible Newtonian fluid, as shown in Fig. 8. The fluid and

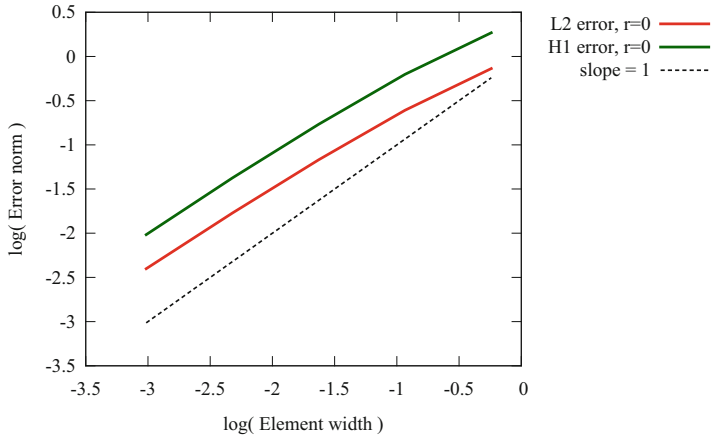


Fig. 7 Convergence of the $L^2(\Omega^{\text{err}})$ and $H^1(\Omega^{\text{err}})$ errors for $r = 0$ for Navier–Stokes flow with moving boundaries and zero viscosity

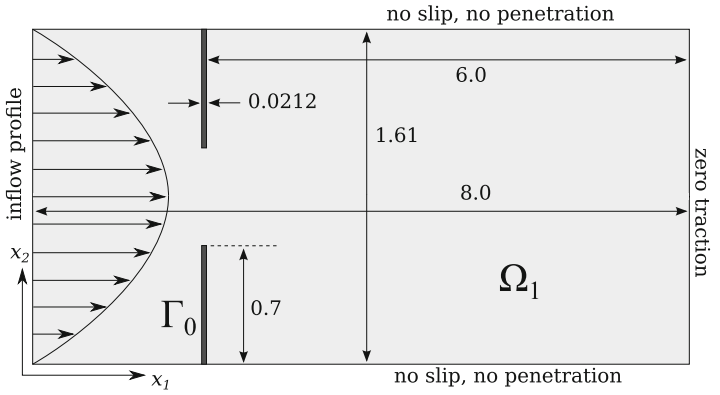


Fig. 8 Geometry and boundary conditions of the 2D heart valve benchmark

structure have equal densities of $\rho_1 = \rho_2 = 100$. The viscosity is $\mu = 10$. The structure is a St. Venant–Kirchhoff material with Young's modulus $E = 5.6 \times 10^7$ and Poisson ratio $\nu = 0.4$. The top and bottom of the channel have no-slip boundary conditions, the right end is traction-free, and the left end has a prescribed, time-dependent velocity profile,

$$\mathbf{u}_1(y\mathbf{e}_2, t) = \begin{cases} 5(\sin(2\pi t) + 1.1)y(1.61 - y)\mathbf{e}_1 & , \ t > 0 \\ \mathbf{0} & , \ \text{otherwise} \end{cases}, \tag{56}$$

where the origin of the spatial coordinate system is at the bottom left corner of the domain. The parameter γ in (5) is set to zero.

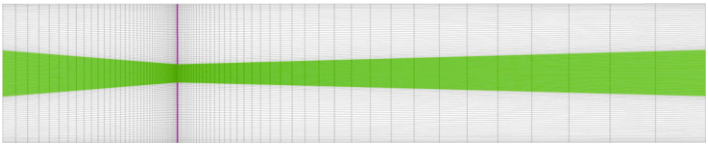


Fig. 9 The reference configuration of the body-fitted mesh for the 2D valve problem, with leaflets highlighted in magenta and areas of softened mesh highlighted in green

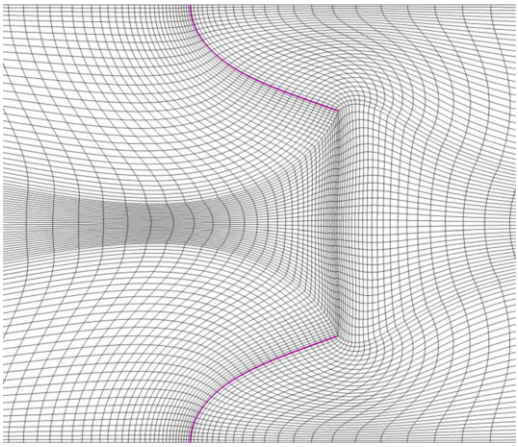


Fig. 10 The deformation of the body-fitted fluid mesh at $t = 0.5$

5.3.2 Body-Fitted Reference Computation

The mesh for the body-fitted reference computation is shown in Fig. 9. We use generalized-α time integration with $\rho_\infty = 0.5$ and a time step of $\Delta t = 0.005$. The selected resolution ensures that the displacement history of the upper beam tip changes negligibly with further refinement.

The fluid mesh deforms from one time step to the next according to the solution of a fictitious isotropic linear elastic problem that takes the location of the beam as a displacement boundary condition. The velocity of this deformation enters into (10) as $\hat{\mathbf{u}}^h$. Mesh quality is preserved throughout the deformation with Jacobian-based stiffening [97, 125–129]. In the present problem, we also find it necessary to soften the fictitious material governing the deformation of elements between the leaflets. The resulting deformed mesh at time $t = 0.5$ is shown in Fig. 10.

5.3.3 Immersogeometric Computations

We test three immersogeometric discretizations of the problem, using the VMS fluid formulation. The first, M1, evenly divides the fluid domain into 128×32 quadratic B-spline elements and each beam into 64 quadratic B-spline elements. The other

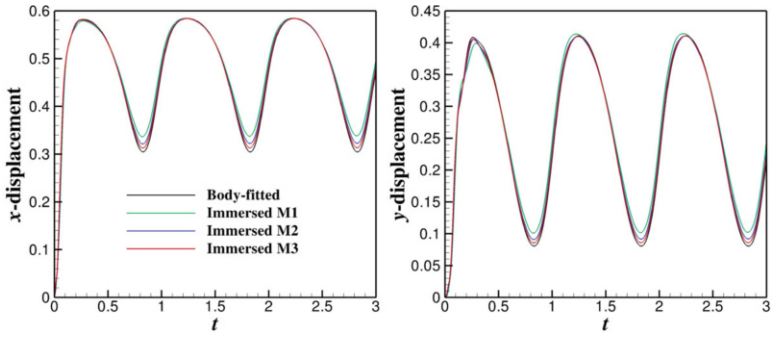


Fig. 11 The x- and y-displacements of the upper leaflet tip, computed on the immersed and body-fitted meshes

two are uniform refinements of M1. We refine also in time, using $\Delta t = 0.01$ with M1, $\Delta t = 0.005$ with M2, and $\Delta t = 0.0025$ with M3.

The time integration of the fluid–structure coupling is done using DAL with $r = 0$ and generalized-α parameters determined by $\rho_\infty = 0.5$. Following (29) and the low-Reynolds number branch of (30), we scale penalty parameters $\tau^B_{(\cdot)}$ inversely with mesh size, choosing $\tau^B_{(\cdot)} = 10^4$ on M1, $\tau^B_{(\cdot)} = 2 \times 10^4$ on M2, and $\tau^B_{(\cdot)} = 4 \times 10^4$ on M3. VMS parameters are scaled near the structure using $s^{\text{shell}} = 10^6$.

5.3.4 Comparison of Results

Figure 11 shows the x- and y-displacements of the upper beam tip for the body-fitted and immersed computations. Displacement histories from M1, M2, and M3 converge toward the body-fitted result. Comparisons of pressure contours at time $t = 0.5$ are given in Fig. 12, showing general agreement between immersogeometric and body-fitted flow fields. Velocity streamlines at $t = 0.5$ for M1 are shown in Fig. 13.

5.4 Benchmark Testing with Div-conforming B-splines

To verify the IMGA implementation using div-conforming B-splines for the fluid subproblem, we again use the 2D benchmark problem defined in Sect. 5.3. Although the problem domain is rectangular, we demonstrate convergence with distorted fluid meshes by deforming the interior of the parametric domain, as shown in Fig. 14. For the coarsest mesh, M1, the B-spline knot space is subdivided into 32×128 Bézier

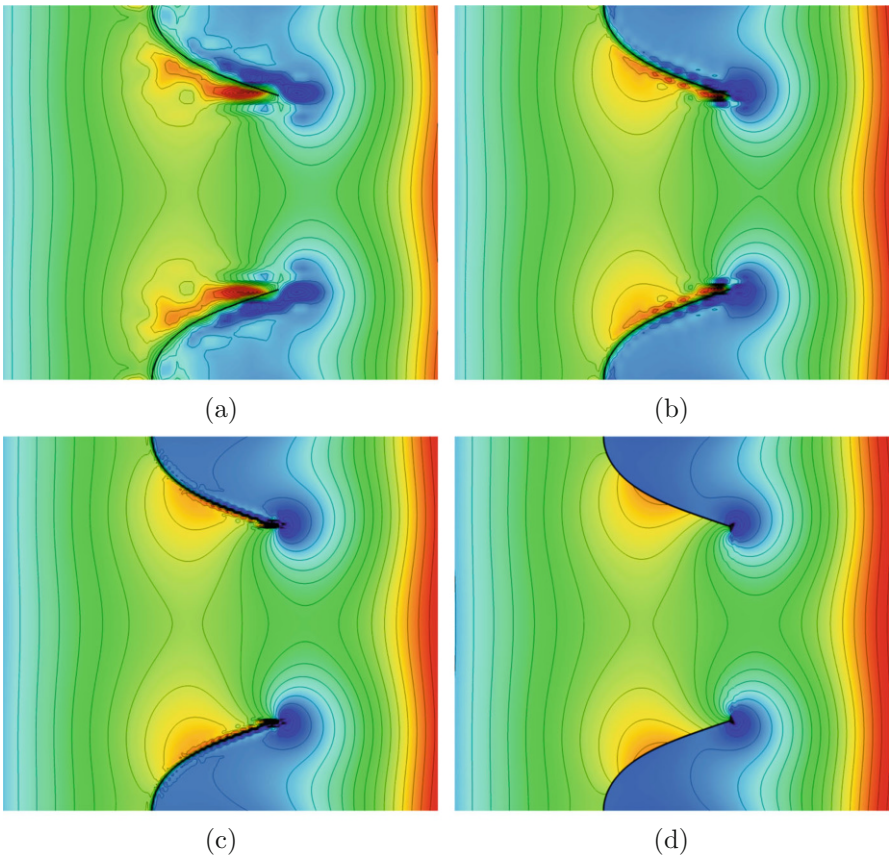


Fig. 12 Pressure contours at $t = 0.5$, from immersed boundary computations on M1, M2, and M3, along with the body-fitted reference. (**a**) Immersed M1. (**b**) Immersed M2. (**c**) Immersed M3. (**d**) Body-fitted reference

elements and div-conforming B-spline velocity and pressure spaces of degree $k' = 1$ are defined on this mesh. The meshes M2 and M3 are uniform refinements of M1.

Normal-direction Dirichlet boundary conditions on mesh boundaries are enforced strongly, while tangential boundary conditions are enforced by penalty. For computations on mesh M$(N+1)$, penalty parameters are $\tau^B_{\text{NOR}} = \tau^B_{\text{TAN}} = C_{\text{no slip}} = 1000 \times 2^N$. We use the backward Euler method in time, with $\Delta t = 1.0 \times 10^{-2} \times 2^{-N}$.

Figures 15 and 16 compare x- and y-displacement histories of the upper beam tip in the three immersogeometric computations and the body-fitted reference. Figure 17 shows snapshots of the computed pressure and velocity solutions. Refinement of immersogeometric discretizations clearly brings this quantity of interest closer to the boundary-fitted reference curve.

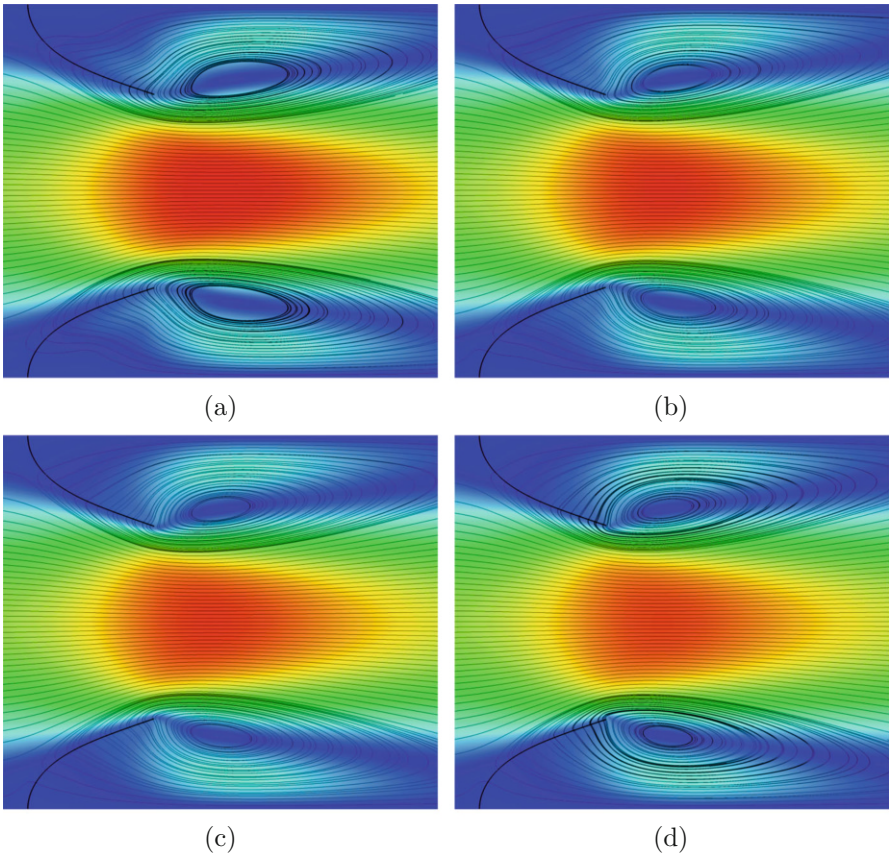


Fig. 13 Velocity streamlines superimposed on a velocity magnitude contour plot, at $t = 0.5$, from immersogeometric computations on M1, M2, and M3, and the body-fitted reference. (**a**) Immersed M1. (**b**) Immersed M2. (**c**) Immersed M3. (**d**) Body-fitted reference

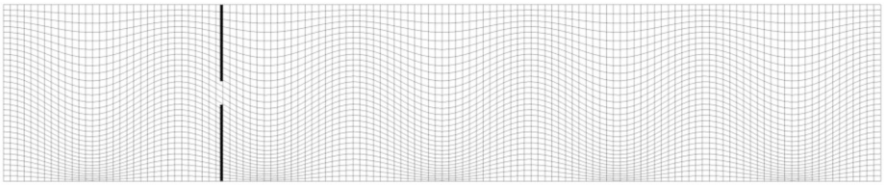


Fig. 14 The physical image of the B-spline parameter space, showing the mesh of unique knots (thin lines) for M1 in relation to the beams (thick lines)

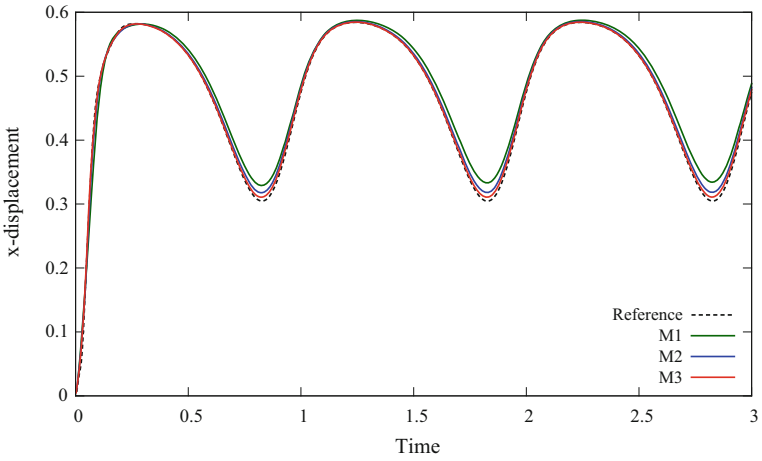


Fig. 15 The x-direction displacement of the tip of the upper beam

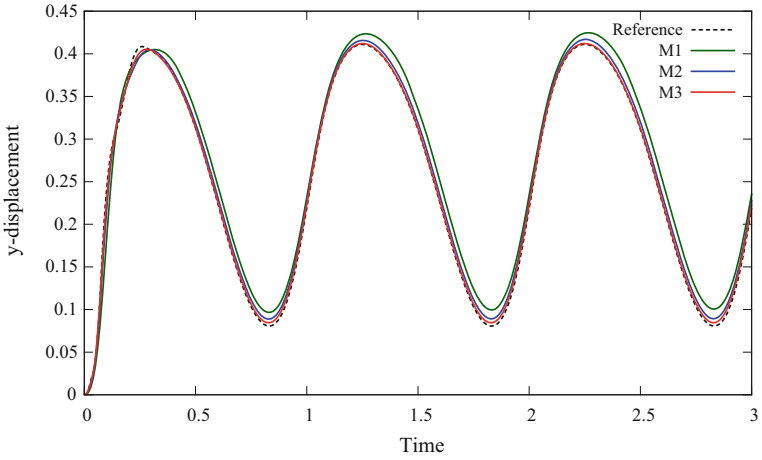


Fig. 16 The y-direction displacement of the tip of the upper beam

6 Application to BHV FSI Analysis

We first review some valve simulations using DAL-based IMGA. Section 6.3 then describes an initial effort toward validating the mathematical model for BHV FSI put forward in Sect. 2.

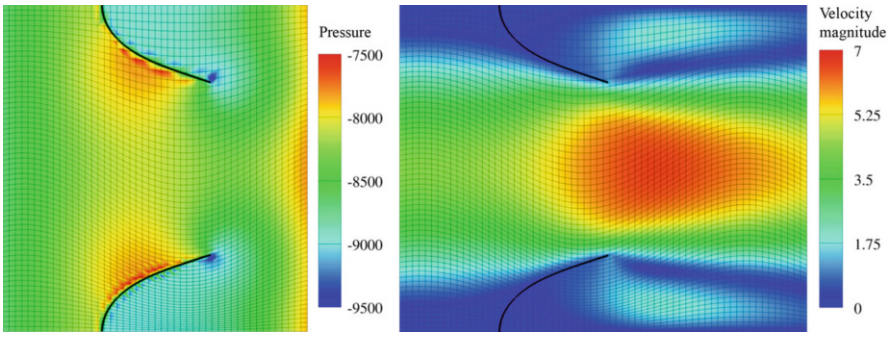


Fig. 17 The pressure field (left) and the velocity magnitude (right) at time $t = 0.5$ on M2

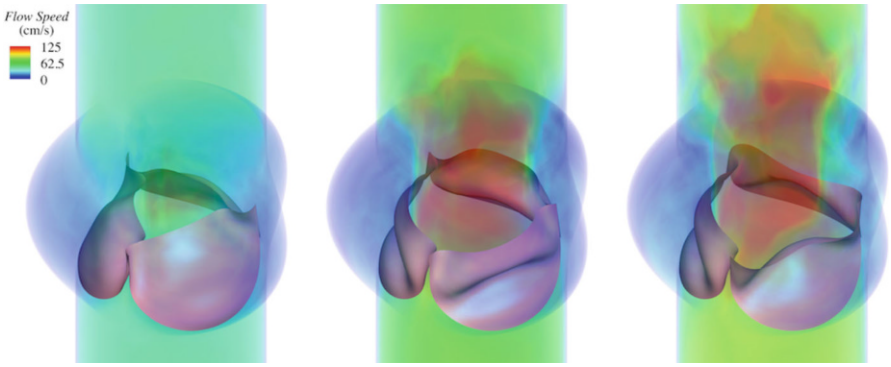


Fig. 18 Snapshots of the valve FSI computation from [77], showing valve deformations and volume renderings of fluid velocity magnitude

6.1 Overview of BHV Simulations

All of the computations reviewed in this section use the VMS discretization of the fluid subproblem described in Sect. 3.1.1 and DAL for fluid–structure coupling. Some of them incorporate phenomena that are beyond the scope of the mathematical problem stated in Sect. 2, such as deforming arteries. However, these BHV simulations illustrate the versatility and practical effectiveness of DAL and IMGA, so we summarize the results while providing citations for additional details.

We introduced the initial variant of DAL in [77], along with the adjustments to VMS and contact penalty needed to effectively simulate a BHV. A crude BHV model immersed in a rigid artery illustrated the effectiveness of the numerics, although the use of an unrealistic pinned boundary condition on the attached edges of the leaflets led to qualitatively incorrect deformations. Further, the rigid artery and resistance outflow boundary condition provided no hydraulic compliance, causing an abnormal flow rate history [77, Figure 28]. Some snapshots of the valve deformations and velocity fields are rendered in Fig. 18.

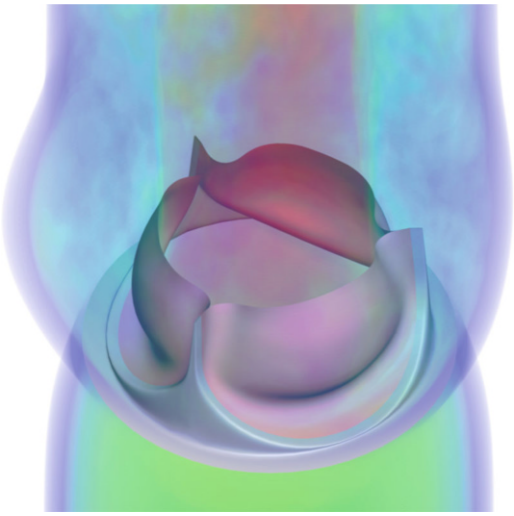


Fig. 19 Snapshot of the valve FSI computation from [132], showing valve deformation and volume rendering of fluid velocity magnitude

The model of [77] was augmented with hydraulic compliance in a follow-up publication [130], by modeling the artery wall as an elastic solid. Unlike the immersed valve, the fluid–artery interface was discretized with a boundary-fitted method, which is a special case of FSITICT [123, 131]. The compliance of the elastic artery led to more realistic flow rates [130, Figure 8].

Hsu et al. [132] realized the potential of IMGA to streamline the design-through-analysis process for BHVs. A parametric design-through-analysis framework was used to generate an analysis-suitable T-spline [133] model of a BHV and IMGA allowed for the BHV design geometry to be directly immersed into a discretization of an artery and lumen. The BHV model incorporated a realistic stent geometry, clamped boundary conditions representative of typical industrial BHVs (cf. patent illustrations in [134]), and a soft tissue constitutive model. A snapshot of the resulting BHV FSI simulation is shown in Fig. 19.

6.2 Div-conforming BHV Simulation

We now look at a BHV simulation using div-conforming B-splines in the fluid subproblem. A capability that is not verified by the div-conforming FSI benchmark testing in Sect. 5.4 is effective simulation of closing heart valves. In principle, div-conforming B-splines should prevent mass loss altogether, but, in practice, for 3D problems, one generally does not solve the discrete algebraic problem exactly, calling this result into question.

6.2.1 Test Problem Definition

A variant of the BHV geometry constructed in [77, Section 5.1] is immersed in a cylindrical fluid domain of radius 1.25 cm and height 3 cm. Rigid extensions are

added to the leaflets, blocking flow passing around the attached boundaries of the leaflets. The fluid subproblem posed on the cylindrical domain has traction boundary conditions on the ends and no-slip and no-penetration conditions on the sides. The bottom of the cylinder is subject to a time-dependent flux condition $\mathbf{h}_1 = P(t)\mathbf{e}_3$, with

$$P(t) = \begin{cases} P_1 & t < T_1 \\ at + b & T_1 \leq t \leq T_2 \\ P_2 & t > T_2 \end{cases} . \tag{57}$$

$P_1 = 2 \times 10^4\,\text{dyn/cm}^2$, $T_1 = 0.05\,\text{s}$, $P_2 = -10^5\,\text{dyn/cm}^2$, $T_2 = 0.1\,\text{s}$, $a = (P_2 - P_1)/(T_2 - T_1)$, and $b = P_1 - aT_1$. The top face is subject to the Neumann condition $\mathbf{h}_1 = \mathbf{0}$. The Neumann boundary stabilization is set to $\gamma = 1$. Properties of the fluid are $\rho_1 = 1\,\text{g/cm}^3$ and $\mu = 4\,\text{cP}$. The valve is modeled as an incompressible neo-Hookean material with shear modulus $\mu_s = 600\,\text{kPa}$ and density $\rho_2 = 1\,\text{g/cm}^3$. The shell thickness is $h_{\text{th}} = 0.04\,\text{cm}$. The attached edges of the leaflets are subject to a clamped boundary condition. The fluid and structure are initially at rest at time $t = 0$. This problem is not intended to be a realistic FSI model of a BHV, but rather to exhibit the similar flow conditions, and demonstrate robustness of div-conforming B-splines.

6.2.2 Discretization

The cylindrical fluid domain is discretized using a B-spline knot space $\widehat{\Omega}_1 = [-1, 1]^2 \times [-1, 2]$. A point $\mathbf{X}$ in this knot space is mapped to the physical domain Ω_1 by

$$\phi_1 = RX_1\sqrt{1 - \frac{1}{2}X_2^2}\ , \quad \phi_2 = RX_2\sqrt{1 - \frac{1}{2}X_1^2}\ , \quad \phi_3 = LX_3\ , \tag{58}$$

with $R = 1.25\,\text{cm}$ and $L = 1\,\text{cm}$. The knot space is evenly subdivided into $42 \times 42 \times 40$ knot spans and div-conforming B-spline velocity and pressure spaces of degree $k' = 1$ are defined on this mesh. The no-penetration constraint on the sides of the cylinder is enforced strongly and the no-slip condition is enforced weakly by velocity penalization, with penalty $C_{\text{no slip}} = 10\,\text{dyn/cm}^2/(\text{cm/s})$. Penalty values are $\tau^B_{\text{NOR}} = 1000\,\text{dyn/cm}^2/(\text{cm/s})$, $\tau^B_{\text{TAN}} = 10\,\text{dyn/cm}^2/(\text{cm/s})$, and $r = 0$. The backward Euler method is used in time with $\Delta t = 5.0 \times 10^{-4}\,\text{s}$.

6.2.3 Results

The valve opening is illustrated in Fig. 20. The closed state is shown in Fig. 21. The flow rate history through the bottom of the cylinder is given in Fig. 22, indicating that the valve blocks flow. These results illustrate the basic soundness of

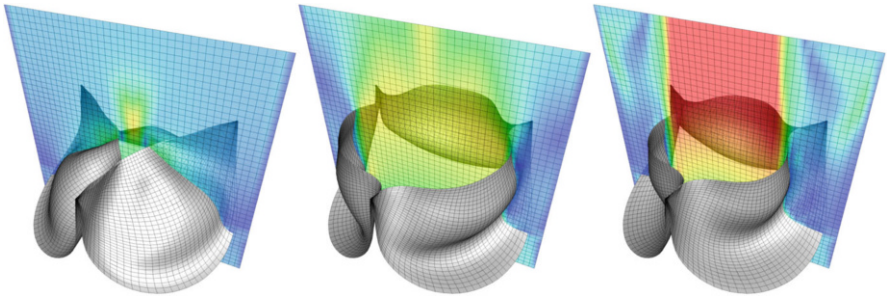


Fig. 20 Velocity magnitude is plotted on a slice, using a color scale ranging from 0 (blue) to $\geq 200\,\text{cm/s}$ (red)

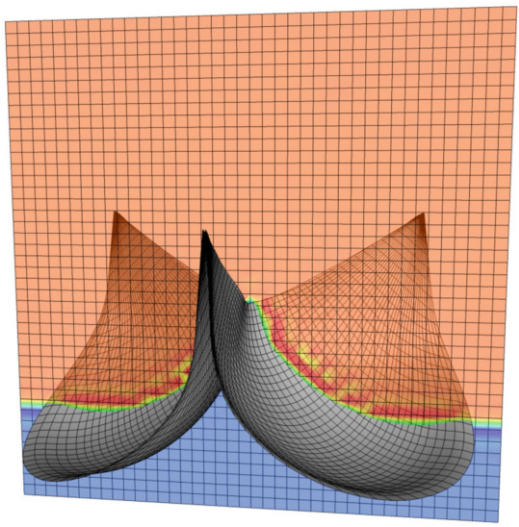


Fig. 21 Pressure is plotted on a slice, using a color scale ranging from $\leq -1.1 \times 10^5\,\text{dyn/cm}^2$ (blue) to $\geq 10^4\,\text{dyn/cm}^2$ (red)

using div-conforming B-splines as a fluid discretization for BHV FSI simulations. We now take a closer look at the mass conservation in the computed solutions. Because we use an iterative solver to approximate the fluid increments in the block iteration, $\nabla \cdot \mathbf{u}_{\text{l}}^h$ is not exactly zero. For the results presented above, we solve for fluid increments with a Krylov method, to a relative tolerance of 10^{-2} for the preconditioned residual. Even with this loose tolerance, there is no disastrous mass loss. We now recompute one step at a time when the valve is closed, under a large pressure jump, with a range of relative tolerances. For this experiment, we use the un-preconditioned residual to measure convergence, so that results generalize more readily to other iterative solvers. The residual is assembled in centimeter–gram–second (CGS) units, without any scaling to compensate for the difference in units between entries of the momentum and continuity equation residuals. The velocity

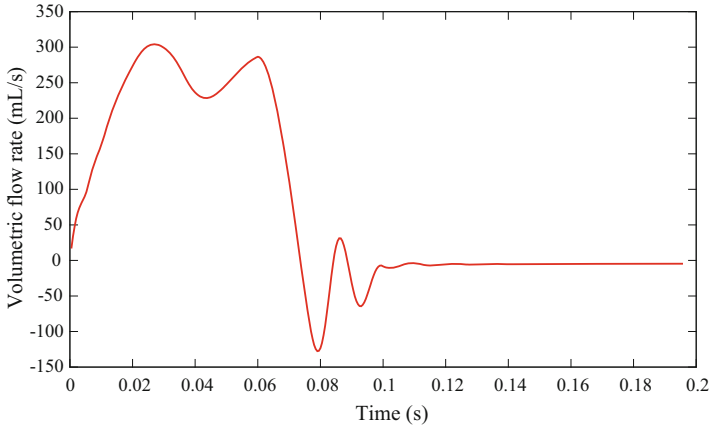


Fig. 22 The volumetric flow rate through the cylinder

Table 1 The effect of relative tolerance in the approximate inversion of A_{f} on mass conservation

Solver tolerance	$\|\nabla \cdot \mathbf{u}_1\|_{L^2(\Omega_1)}$ (CGS units)
10^{-1}	3.9×10^{-5}
10^{-2}	1.2×10^{-5}
10^{-3}	3.0×10^{-7}
10^{-4}	2.0×10^{-8}
10^{-5}	1.2×10^{-9}
10^{-6}	2.4×10^{-10}
10^{-7}	4.3×10^{-11}

divergence L^2 norms of the solutions to this time step are collected in Table 1. As expected, velocity divergence approaches zero as the tolerance decreases.

6.3 Simulating an In Vitro Experiment

This section serves both to further illustrate the application of div-conforming B-splines to realistic problems and to argue that the modeling assumptions from Sect. 2 can represent the dynamics of an artificial heart valve immersed in fluid, by summarizing the validation effort detailed in [86, Section 7].

6.3.1 Description of the Experiment

The validation experiment uses a latex valve in an acrylic tube. We constructed the valve by gluing latex leaflets to an aluminum stent. Leaflet are cut from a flat sheet of latex with thickness 0.054 cm. The valve is shown in Fig. 24. The acrylic tube, illustrated in Fig. 23, has an inner diameter varying between 2 and 3 cm along the

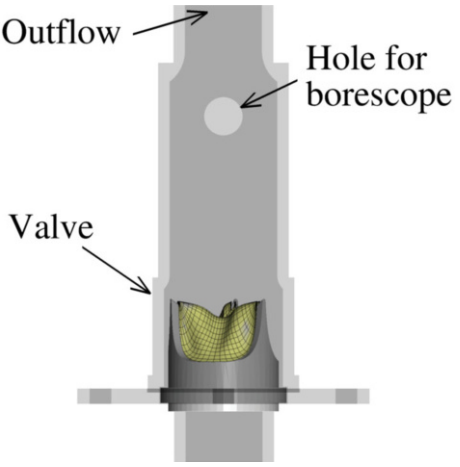


Fig. 23 A to-scale diagram of the tube, showing its relation to the valve and stent

length of the tube, and is roughly the size of a typical human ascending aorta. A hole is included in the side of the tube, for capturing images with a borescope.

Water is pumped through the tube using a flow loop system similar to the bioreactor detailed in [135]. Volumetric flow rate through the tube is measured using an ultrasonic flow meter. We use the IMGA with DAL and div-conforming B-splines to simulate only the segment of tubing containing the artificial aortic valve.

6.3.2 Mathematical Model of the Experiment

This section specifies an instance of the mathematical problem stated in Sect. 2 that models the experiment described in Sect. 6.3.1.

Fluid Subproblem

The mathematical model simplifies the geometry of the region occupied by fluid. Ω_1 is the image of a parametric space $\widehat{\Omega}_1 = (-1, 1)^2 \times (-1, 4.5) \subset \mathbb{R}^3$ under the mapping $\boldsymbol{\phi}$, which is defined by

$$\phi_1 = R(X_3)X_1\sqrt{1 - \frac{1}{2}X_2^2} \ , \quad \phi_2 = R(X_3)X_2\sqrt{1 - \frac{1}{2}X_1^2} \ , \quad \phi_3 = LX_3 \ , \tag{59}$$

where $L = 1\,\text{cm}$ and $R(X_3)$ is defined by

$$R(X_3) = \begin{cases} R_{\text{in}} & X_3 < z_1 \\ R_{\text{out}} & X_3 > z_2 \\ (R_{\text{out}} - R_{\text{in}})\sin^2\left(\frac{\pi(X_3 - z_1)}{2(z_2 - z_1)}\right) + R_{\text{in}} & \text{otherwise} \end{cases} , \tag{60}$$

with $z_1 = -0.45\,\text{cm}$, $z_2 = 0$, $R_{\text{in}} = 1\,\text{cm}$, and $R_{\text{out}} = 1.4025\,\text{cm}$.

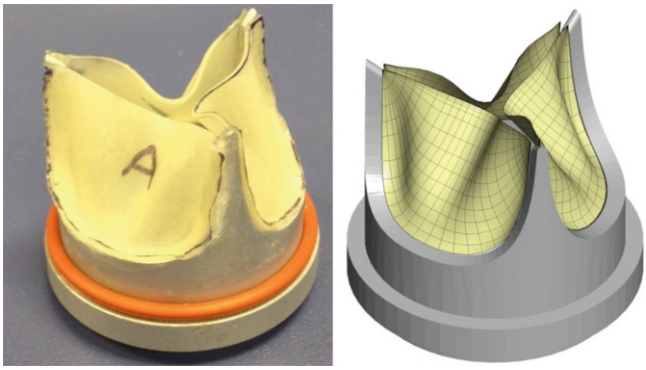


Fig. 24 A visual comparison of the physical valve and its computational model

The lateral sides of Ω_1 are subject to no-slip and no-penetration conditions. The inflow face of the domain is subject to a time-dependent plug flow condition with experimentally measured volumetric flow rate. The outflow is a homogeneous Neumann boundary with $\gamma = 1$. The fluid velocity initial condition is $\mathbf{u}_1^0 \equiv \mathbf{0}$. To model water, the viscosity of the fluid is $\mu = 1$ cP and the density is $\rho_1 = 1.0\,\text{g/cm}^3$.

Structure Subproblem

The latex leaflets are modeled as incompressible neo-Hookean material with shear modulus $\mu_s = 8.7 \times 10^6\,\text{dyn/cm}^2$ (based on uniaxial stretching experiments). The geometry of the stress-free reference configuration Γ_0 is specified by manually selecting B-spline control points to approximate the pattern used to cut the leaflets out of the latex sheet. The leaflets are therefore flat in Γ_0. These leaflets are deformed into a static equilibrium configuration Γ_0', (a discrete approximation of) which is shown in Fig. 24. The boundary corresponding to the attached edge is subject to a strongly enforced clamped boundary condition. In a slight abuse of the notation introduced in Sect. 2, the leaflets are considered to be initially at rest in the deformed configuration Γ_0', rather than the stress-free configuration Γ_0.

6.3.3 Discretization of the Mathematical Model

The fluid parametric domain $\widehat{\Omega}_1$ is split evenly into $64 \times 64 \times 99$ Bézier elements, used to define div-conforming B-spline spaces of degree $k' = 1$. No-slip and inflow Dirichlet boundary conditions are enforced by velocity penalization, with penalty-constants of $C_{\text{no slip}} = 10\,\text{dyn/cm}^2/(\text{cm/s})$ and $C_{\text{inflow}} = 1000\,\text{dyn/cm}^2/(\text{cm/s})$. No-penetration on the lateral sides of the flow domain is enforced strongly. The structure is discretized with a 936-element quadratic B-spline mesh. The equilibrium configuration Γ_0' is approximated by driving a dynamic simulation

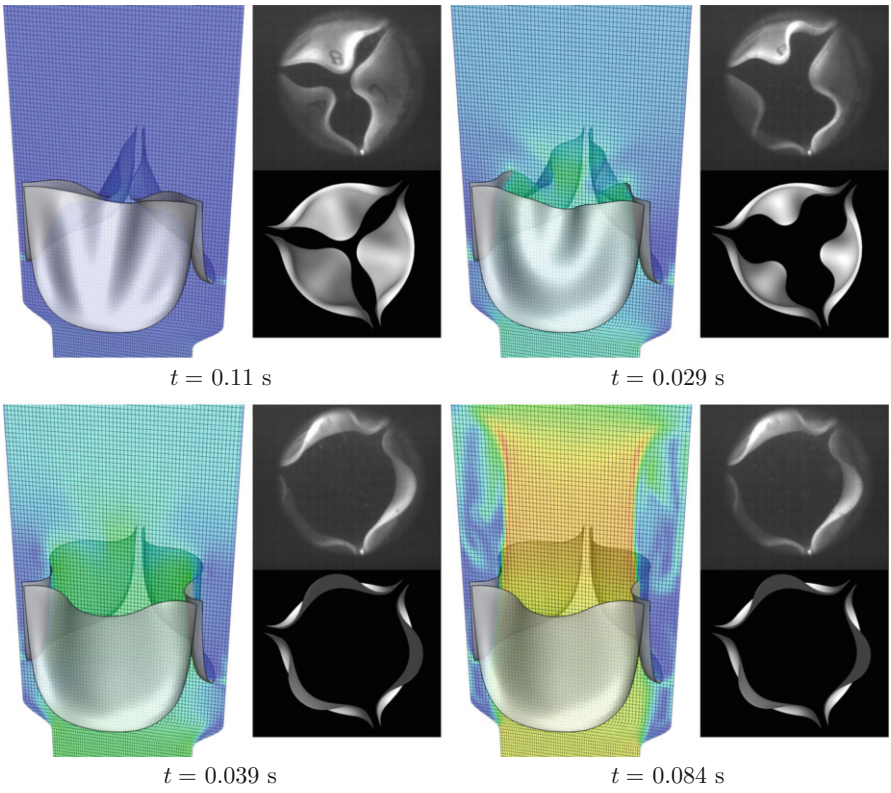


Fig. 25 Several snapshots of the computed solution, compared with experimental images. At each time instant, the computed solution is shown in the left-hand frame and at the bottom of the right-hand frame. The experimental results are shown in the top of the right-hand frame. Colors indicate fluid velocity magnitude on a slice. Color scale: 0 (blue) to $\geq$200 cm/s (red)

with mass damping from Γ_0 to a steady solution. The attached edges of the leaflets are then clamped into this configuration. The FSI penalty parameters are $\tau^B_{\text{NOR}} = 1000\,\text{dyn/cm}^2/(\text{cm/s})$ and $\tau^B_{\text{TAN}} = 10\,\text{dyn/cm}^2/(\text{cm/s})$. The DAL stabilization parameter r is set to zero. Backward Euler time integration is used with $\Delta t = 2.5 \times 10^{-4}$ s.

6.3.4 Comparison of Results

We now compare computational and experimental results. Experimental results are a sequence of images taken through a borescope. Figure 25 compares the computed deformations at several time points with images collected in the experiment. For direct comparison with experimental images, the computed deformations are rendered using perspective, from a vantage point corresponding to the tip of the borescope in the experiment.

The main qualitative difference between these sets of images is in the degree of symmetry of the leaflet deformations during the transition to the fully open state. This difference is expected, given that the initial condition to the computer simulation is symmetrical while the physical valve is not. Asymmetry is mainly due to experimental errors introduced by manually gluing each initially flat leaflet into the stent. The qualitative agreement of results indicates that the modeling assumptions of Sect. 2 are not wildly inappropriate for predicting the deformations of BHV leaflets immersed in physiological flow fields, and may be able to predict quantities of interest related to deformation (such as strain) with practical accuracy. The computed results also agree with qualitative features of artificial valve leaflet deformations observed in other in vitro experiments. The computed solution at time $t = 0.029\,\mathrm{s}$ shows the opening process, as characterized by reversal of leaflet curvature, beginning primarily near the attached edge, as observed by Iyengar et al. [136]. Hsu et al. [132] found that this behavior is *not* captured by simulations using only structural dynamics.

7 Conclusions and Further Work

This chapter reviews the development, verification, and application of a novel numerical method combining IMGA and DAL to simulate thin structures with spline-based geometries immersed in viscous incompressible fluids. We find that this method is sufficiently robust to survive application to FSI analysis of BHVs functioning under physiological conditions.

The method described here is not limited to BHV simulation. We have also applied it to IMGA of the hydraulic arresting gears that help dissipate the kinetic energy of fixed-wing aircraft landing on short runways. Initial results, published in [137], compare favorably with earlier body-fitted simulations of such devices [138]. The flexibility provided by immersogeometric FSI analysis allowed for automated optimization of the device geometry.

Despite its successful application to BHV FSI and other problems, the DAL method outlined here can be improved. The present guidelines for selecting free penalty parameters are based on imprecise dimensional analysis. More precise and rational selection of parameters will likely stem from further numerical analysis of linear model problems, building on the initial work of [86]. Another undesirable aspect of the method presented in this dissertation is the trade-off between conservation and stability parameterized by the stabilization coefficient r (introduced in Sect. 4.1). A possible improvement is to apply the inconsistent stabilization of $r > 0$ only to fine scales of the interface Lagrange multiplier, while retaining strong consistency on coarse scales. Initial work on this was published in [139] and is analyzed in a forthcoming paper [140].

Lastly, the promising initial results of immersogeometric FSI analysis using div-conforming B-spline discretizations of the fluid subproblem indicate that div-conforming B-splines merit further investigation. Casquero et al. [141] have also

recently applied div-conforming B-splines in conjunction with the immersed-boundary numerical approach of [142–144] and efficient solvers from [145]. The ideas of immersogeometric FSI analysis and div-conforming B-spline flow discretizations appear to enjoy a symbiotic connection, in that the strong mass conservation of structure preserving flow discretizations improves the quality of immersogeometric FSI solutions, while the application of div-conforming B-splines to increasingly complicated and realistic problems motivates the development of more powerful implementations.

Acknowledgements The work summarized in this chapter was supported by the National Heart, Lung, and Blood Institute of the National Institutes of Health (NIH/NHLBI) under award number R01HL129077. We thank the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for providing HPC resources that have contributed to the research results reported in this chapter.

References

1. F. J. Schoen and R. J. Levy. Calcification of tissue heart valve substitutes: progress toward understanding and prevention. *Ann. Thorac. Surg.*, 79(3):1072–1080, 2005.
2. P. Pibarot and J. G. Dumesnil. Prosthetic heart valves: selection of the optimal prosthesis and long-term management. *Circulation*, 119(7):1034–1048, 2009.
3. J. S. Soares, K. R. Feaver, W. Zhang, D. Kamensky, A. Aggarwal, and M. S. Sacks. Biomechanical behavior of bioprosthetic heart valve heterograft tissues: Characterization, simulation, and performance. *Cardiovascular Engineering and Technology*, 7(4):309–351, 2016.
4. M. J. Thubrikar, J. D. Deck, J. Aouad, and S. P. Nolan. Role of mechanical stress in calcification of aortic bioprosthetic valves. *J. Thorac. Cardiovasc. Surg.*, 86(1):115–125, Jul 1983.
5. R. F. Siddiqui, J. R. Abraham, and J. Butany. Bioprosthetic heart valves: modes of failure. *Histopathology*, 55:135–144, 2009.
6. W. Sun, A. Abad, and M. S. Sacks. Simulated bioprosthetic heart valve deformation under quasi-static loading. *Journal of Biomechanical Engineering*, 127(6):905–914, 2005.
7. F. Auricchio, M. Conti, A. Ferrara, S. Morganti, and A. Reali. Patient-specific simulation of a stentless aortic valve implant: the impact of fibres on leaflet performance. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(3):277–285, 2014.
8. H. Kim, J. Lu, M. S. Sacks, and K. B. Chandran. Dynamic simulation of bioprosthetic heart valves using a stress resultant shell model. *Annals of Biomedical Engineering*, 36(2):262–275, 2008.
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*, 29:329–349, 1981.
10. J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.
11. J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian–Eulerian methods. In *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 14. John Wiley & Sons, 2004.

12. 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(3):339–351, 1992.
13. 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(3):353–371, 1992.
14. A. A. Johnson and T. E. Tezduyar. Parallel computation of incompressible flows with complex geometries. *International Journal for Numerical Methods in Fluids*, 24:1321–1340, 1997.
15. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Massively parallel finite element computation of 3D flows – mesh update strategies in computation of moving boundaries and interfaces. In A. Ecer, J. Hauser, P. Leca, and J. Periaux, editors, *Parallel Computational Fluid Dynamics – New Trends and Advances*, pages 21–30. Elsevier, 1995.
16. 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*, 145:301–321, 1997.
17. A. A. Johnson and T. E. Tezduyar. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics*, 23:130–143, 1999.
18. K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Space–time interface-tracking with topology change (ST-TC). *Computational Mechanics*, 54:955–971, 2014.
19. K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Space–time fluid mechanics computation of heart valve models. *Computational Mechanics*, 54:973–986, 2014.
20. K. Takizawa, T. E. Tezduyar, T. Terahara, and T. Sasaki. Heart valve flow computation with the integrated space–time VMS, slip interface, topology change and isogeometric discretization methods. *Computers & Fluids*, 158:176–188, 2017.
21. K. Takizawa, T. E. Tezduyar, T. Terahara, and T. Sasaki. Heart valve flow computation with the space–time slip interface topology change (ST-SI-TC) method and isogeometric analysis (IGA). In P. Wriggers and T. Lenarz, editors, *Biomedical Technology: Modeling, Experiments and Simulation*, Lecture Notes in Applied and Computational Mechanics, pages 77–99. Springer International Publishing, 2018.
22. C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.
23. R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
24. F. Sotiropoulos and X. Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, 2014.
25. D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–252:116–150, 2012.
26. T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.
27. 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*, 46:31–41, 2010.
28. J. de Hart. *Fluid–Structure Interaction in the Aortic Heart Valve: a three-dimensional computational analysis*. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2002.
29. J. De Hart, G. W. M. Peters, P. J. G. Schreurs, and F. P. T. Baaijens. A three-dimensional computational analysis of fluid–structure interaction in the aortic valve. *Journal of Biomechanics*, 36:103–112, 2003.

30. J. De Hart, F. P. T. Baaijens, G. W. M. Peters, and P. J. G. Schreurs. A computational fluid–structure interaction analysis of a fiber-reinforced stentless aortic valve. *Journal of Biomechanics*, 36:699–712, 2003.
31. R. van Loon. *A 3D method for modelling the fluid–structure interaction of heart valves*. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2005.
32. R. van Loon, P. D. Anderson, and F. N. van de Vosse. A fluid–structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of Computational Physics*, 217:806–823, 2006.
33. R. van Loon. Towards computational modelling of aortic stenosis. *International Journal for Numerical Methods in Biomedical Engineering*, 26:405–420, 2010.
34. F. P. T. Baaijens. A fictitious domain/mortar element method for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 35(7):743–761, 2001.
35. B. E. Griffith. Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions. *International Journal for Numerical Methods in Biomedical Engineering*, 28(3):317–345, 2012.
36. I. Borazjani. Fluid–structure interaction, immersed boundary-finite element method simulations of bio-prosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 257:103–116, 2013.
37. L. Ge and F. Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics*, 225(2):1782–1809, 2007.
38. I. Borazjani, L. Ge, and F. Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics*, 227(16):7587–7620, 2008.
39. A. Gilmanov, T. B. Le, and F. Sotiropoulos. A numerical approach for simulating fluid structure interaction of flexible thin shells undergoing arbitrarily large deformations in complex domains. *Journal of Computational Physics*, 300:814–843, 2015.
40. A. Gilmanov and F. Sotiropoulos. Comparative hemodynamics in an aorta with bicuspid and trileaflet valves. *Theoretical and Computational Fluid Dynamics*, 30(1):67–85, 2016.
41. LS-DYNA Finite Element Software: Livermore Software Technology Corp. http://www.lstc.com/products/ls-dyna. Accessed 30 April 2016.
42. G. G. Chew, I. C. Howard, and E. A. Patterson. Simulation of damage in a porcine prosthetic heart valve. *Journal of Medical Engineering & Technology*, 23(5):178–189, 1999.
43. C. J. Carmody, G. Burriesci, I. C. Howard, and E. A. Patterson. An approach to the simulation of fluid–structure interaction in the aortic valve. *Journal of Biomechanics*, 39:158–169, 2006.
44. F. Sturla, E. Votta, M. Stevanella, C. A. Conti, and A. Redaelli. Impact of modeling fluid–structure interaction in the computational analysis of aortic root biomechanics. *Medical Engineering and Physics*, 35:1721–1730, 2013.
45. W. Wu, D. Pott, B. Mazza, T. Sironi, E. Dordoni, C. Chiastra, L. Petrini, G. Pennati, G. Dubini, U. Steinseifer, S. Sonntag, M. Kuetting, and F. Migliavacca. Fluid–structure interaction model of a percutaneous aortic valve: Comparison with an in vitro test and feasibility study in a patient-specific case. *Annals of Biomedical Engineering*, 44(2):590–603, 2016.
46. R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.
47. R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J. Res. Develop.*, 11:215–234, 1967.
48. A. E. J. Bogaers, S. Kok, B. D. Reddy, and T. Franz. Quasi-Newton methods for implicit black-box FSI coupling. *Computer Methods in Applied Mechanics and Engineering*, 279(0):113–132, 2014.
49. E. H. van Brummelen. Added mass effects of compressible and incompressible flows in fluid–structure interaction. *Journal of Applied Mechanics*, 76:021206, 2009.
50. C. Michler, H. van Brummelen, and R. de Borst. An investigation of interface-GMRES(R) for fluid–structure interaction problems with flutter and divergence. *Computational Mechanics*, 47(1):17–29, 2011.

51. M. Astorino, J.-F. Gerbeau, O. Pantz, and K.-F. Traoré. Fluid–structure interaction and multi-body contact: Application to aortic valves. *Computer Methods in Applied Mechanics and Engineering*, 198:3603–3612, 2009.
52. K. Cao, M. Bukač, and P. Sucosky. Three-dimensional macro-scale assessment of regional and temporal wall shear stress characteristics on aortic valve leaflets. *Computer Methods in Biomechanics and Biomedical Engineering*, 19(6):603–613, 2016.
53. 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*, 194:4135–4195, 2005.
54. J. A. Evans, Y. Bazilevs, I. Babuška, and T. J. R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198:1726–1741, 2009.
55. A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199(17–20):1143–1152, 2010.
56. A. Buffa, J. Rivas, G. Sangalli, and R. Vásquez. Isogeometric discrete differential forms in three dimensions. *SIAM Journal on Numerical Analysis*, 49(2):814–844, 2011.
57. 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*, 41:371–378, 2008.
58. Y. Bazilevs, V. M. Calo, J. A. Cottrel, 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*, 197:173–201, 2007.
59. J. A. Evans. *Divergence-free B-spline Discretizations for Viscous Incompressible Flows*. Ph.D. thesis, University of Texas at Austin, Austin, Texas, United States, 2011.
60. J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations. *Mathematical Models and Methods in Applied Sciences*, 23(08):1421–1478, 2013.
61. J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
62. J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.
63. N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K.U. Bletzinger, Y. Bazilevs, and T. Rabczuk. Rotation-free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200:3410–3424, 2011.
64. J. Kiendl, M.-C. Hsu, M. C. H. Wu, and A. Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, 2015.
65. S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010.
66. L. De Lorenzis, İ. Temizer, P. Wriggers, and G. Zavarise. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87:1278–1300, 2011.
67. S. Morganti, F. Auricchio, D. J. Benson, F. I. Gambarin, S. Hartmann, T. J. R. Hughes, and A. Reali. Patient-specific isogeometric structural analysis of aortic valve closure. *Computer Methods in Applied Mechanics and Engineering*, 284:508–520, 2015.
68. M. A. Scott. *T-splines as a Design-Through-Analysis Technology*. PhD thesis, The University of Texas at Austin, August 2011.
69. T. W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276–283, 2004.
70. T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCS. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

71. E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249–252:104–115, 2012.
72. M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95:811–846, 2013.
73. M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–731, 2014.
74. D. Schillinger and M. Ruess. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering*, 22(3):391–455, 2015.
75. D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the *p*- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.
76. G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
77. D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
78. M. Hillairet. Lack of collision between solid bodies in a 2D incompressible viscous flow. *Communications in Partial Differential Equations*, 32(9):1345–1371, 2007.
79. Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.
80. M. Esmaily-Moghadam, Y. Bazilevs, T.-Y. Hsia, I. E. Vignon-Clementel, A. L. Marsden, and Modeling of Congenital Hearts Alliance (MOCHA). A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations. *Computational Mechanics*, 48:277–291, 2011.
81. G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, Chichester, 2000.
82. T. M. van Opstal, J. Yan, C. Coley, J. A. Evans, T. Kvamsdal, and Y. Bazilevs. Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 316:859–879, 2017.
83. T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J. B. Quincy. The variational multiscale method–A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.
84. S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 3rd ed.* Springer, 2008.
85. J. A. Evans and T. J. R. Hughes. Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements. *Numerische Mathematik*, 123:259–290, 2013.
86. D. Kamensky, M.-C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.
87. J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.
88. L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.
89. 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*, 32:199–259, 1982.

90. Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.
91. Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.
92. 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*, 199:780–790, 2010.
93. 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*, 229:3402–3414, 2010.
94. M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions. *Computational Mechanics*, 50:499–511, 2012.
95. H. J. C. Barbosa and T. J. R. Hughes. The finite element method with Lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1):109–128, 1991.
96. 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*, 60:371–75, 1993.
97. Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.
98. M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.
99. M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1969.
100. D. Kamensky, J. A. Evans, and M.-C. Hsu. Stability and conservation properties of collocated constraints in immersogeometric fluid-thin structure interaction analysis. *Communications in Computational Physics*, 18:1147–1180, 2015.
101. A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118–125, 1967.
102. D. Goldstein, R. Handler, and L. Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366, 1993.
103. L. B. Wahlbin. Local behavior in finite element methods. In P. G. Ciarlet and J. L. Lions, editors, *Finite Element Methods (Part 1)*, volume 2 of *Handbook of Numerical Analysis*, pages 353–522. North-Holland, 1991.
104. T.-C. Tuan and D. B. Goldstein. Direct numerical simulation of arrays of microjets to manipulate near wall turbulence. Technical Report CAR-96-3, U. T. Austin Center for Aerodynamics Research, 2011.
105. D. B. Goldstein and T.-C. Tuan. Secondary flow induced by riblets. *Journal of Fluid Mechanics*, 363:115–151, 1998.
106. D. B. Goldstein. DNS for new applications of surface textures and MEMS actuators for turbulent boundary layer control - FINAL REPORT. Technical Report AFRL-SR-AR-TR-07-0363, AFSOR, 2006.
107. K. Stephani and D. Goldstein. DNS study of transient disturbance growth and bypass transition due to realistic roughness. In *Proceedings of 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, AIAA Paper 2009-585, Orlando, Florida, 2009.
108. J. S. Strand and D. B. Goldstein. Direct numerical simulations of riblets to constrain the growth of turbulent spots. *Journal of Fluid Mechanics*, 668:267–292, 2011.
109. C. J. Doolittle, S. D. Drews, and D. B. Goldstein. Near-field flow structures about subcritical surface roughness. *Physics of Fluids*, 26:124106, 2014.

110. E. M. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465, 1996.
111. E. M. Saiki and S. Biringen. Spatial numerical simulation of boundary layer transition: effects of a spherical particle. *Journal of Fluid Mechanics*, 345:133–164, 1997.
112. W.-X. Huang, S. J. Shin, and H. J. Sung. Simulation of flexible filaments in a uniform flow by the immersed boundary method. *Journal of Computational Physics*, 226(2):2206–2228, 2007.
113. S. J. Shin, W.-X. Huang, and H. J. Sung. Assessment of regularized delta functions and feedback forcing schemes for an immersed boundary method. *International Journal for Numerical Methods in Fluids*, 58(3):263–286, 2008.
114. W.-X. Huang and H. J. Sung. An immersed boundary method for fluid–flexible structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 198(33–36):2650–2661, 2009.
115. J. Ryu, S. G. Park, B. Kim, and H. J. Sung. Flapping dynamics of an inverted flag in a uniform flow. *Journal of Fluids and Structures*, 57:159–169, 2015.
116. E. Uddin, W.-X. Huang, and H. J. Sung. Actively flapping tandem flexible flags in a viscous flow. *Journal of Fluid Mechanics*, 780:120–142, 10 2015.
117. M. Souli, Y. Sofiane, and L. Olovsson. ALE and fluid/structure interaction in LS-DYNA. In *Proceedings of Emerging Technology in Fluids, Structures, and Fluid–Structure Interactions*. ASME, 2004.
118. M. Souli, N. Capron, and U. Khan. Fluid structure interaction and airbag ALE for out of position. In *Proceedings of the ASME Pressure Vessels and Piping Conference*. AMSE, 2005.
119. M. Souli, J. Wang, I. Do, and C. Hao. ALE and fluid structure interaction in LS-DYNA. In *Proceedings of the 8th International LS-DYNA Users Conference*, 2011.
120. A. Haufe, K. Weimar, and U. Göhner. Advanced airbag simulation using fluid-structure-interaction and the Eluerian method in LS-DYNA. In *Proceedings of the LS-DYNA Anwenderforum*, 2004.
121. A.J. Gil, A. Arranz Carreño, J. Bonet, and O. Hassan. An enhanced immersed structural potential method for fluid–structure interaction. *Journal of Computational Physics*, 250:178–205, 2013.
122. C. Hesch, A. J. Gil, A. Arranz Carreño, and J. Bonet. On continuum immersed strategies for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 247–248:51–64, 2012.
123. T. Wick. Flapping and contact FSI computations with the fluid–solid interface-tracking/interface-capturing technique and mesh adaptivity. *Computational Mechanics*, 53(1):29–43, 2014.
124. C. Kadapa, W. G. Dettmer, and D. Perić. A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-spline grids. *Computer Methods in Applied Mechanics and Engineering*, 301:1–27, 2016.
125. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.
126. 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*, 119:73–94, 1994.
127. K. Stein, T. Tezduyar, and R. Benney. Mesh moving techniques for fluid–structure interactions with large displacements. *Journal of Applied Mechanics*, 70:58–63, 2003.
128. K. Stein, T. E. Tezduyar, and R. Benney. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering*, 193:2019–2032, 2004.
129. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, Chichester, 2013.
130. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54:1055–1071, 2014.

131. 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*, 64:1201–1218, 2010.
132. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. C. H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M. S. Sacks. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics*, 55:1211–1225, 2015.
133. Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.
134. V. L. Huynh, T. Nguyen, H. L. Lam, X. G. Guo, and R. Kafesjian. Cloth-covered stents for tissue heart valves, 2003. US Patent 6,585,766.
135. D. K. Hildebrand. Design and evaluation of a novel pulsatile bioreactor for biologically active heart valves. Master's thesis, University of Pittsburgh, Pittsburgh, United States, 2003.
136. A. K. S. Iyengar, H. Sugimoto, D. B. Smith, and M. S. Sacks. Dynamic in vitro quantification of bioprosthetic heart valve leaflet motion using structured light projection. *Annals of Biomedical Engineering*, 29(11):963–973, 2001.
137. M. C. H. Wu, D. Kamensky, C. Wang, A. J. Herrema, F. Xu, M. S. Pigazzini, A. Verma, A. L. Marsden, Y. Bazilevs, and M.-C. Hsu. Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear. *Computer Methods in Applied Mechanics and Engineering*, 316:668–693, 2017.
138. C. Wang, M. C. H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs. Modeling of a hydraulic arresting gear using fluid-structure interaction and isogeometric analysis. *Computers & Fluids*, 142:3–14, 2017.
139. D. Kamensky, J. A. Evans, M.-C. Hsu, and Y. Bazilevs. Projection-based stabilization of interface Lagrange multipliers in immersogeometric fluid–thin structure interaction analysis, with application to heart valve modeling. *Computers & Mathematics with Applications*, 74:2068–2088, 2017.
140. Y. Yu, D. Kamensky, M.-C. Hsu, X. Y. Lu, Y. Bazilevs, and T. J. R. Hughes. Error estimates for projection-based dynamic augmented Lagrangian boundary condition enforcement, with application to immersogeometric fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 2018. https://doi.org/10.1142/S0218202518500537.
141. H. Casquero, Y. Zhang, C. Bona-Casas, L. Dalcin and H. Gomez. Non-body-fitted fluid–structure interaction: Divergence-conforming B splines, fully-implicit dynamics, and variational formulation. *Journal of Computational Physics*, 2018. https://doi.org/10.1016/j.jcp.2018.07.020.
142. H. Casquero, C. Bona-Casas, and H. Gomez. A NURBS-based immersed methodology for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 284:943–970, 2015.
143. H. Casquero, L. Liu, C. Bona-Casas, Y. Zhang, and H. Gomez. A hybrid variational-collocation immersed method for fluid–structure interaction using unstructured t-splines. *International Journal for Numerical Methods in Engineering*, 105(11):855–880, 2015.
144. H. Casquero, C. Bona-Casas, and H. Gomez. NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow. *Computer Methods in Applied Mechanics and Engineering*, 316:646–667, 2017.
145. A.F. Sarmiento, A.M.A. Côrtes, D.A. Garcia, L. Dalcin, N. Collier, and V.M. Calo. PetIGA-MF: A multi-field high-performance toolbox for structure-preserving B-splines spaces. *Journal of Computational Science*, 18(Supplement C):117–131, 2017.

A Numerical Analysis of Rheology of Capsule Suspensions Using a GPU-Accelerated Boundary Element Method

Yohsuke Imai and Daiki Matsunaga

Abstract Understanding the behavior of capsules in flow and the rheology of capsule suspensions is of fundamental importance for diverse problems in nature and engineering. The particle Reynolds number of capsules is often small, and the flow field is given by the boundary integral formulation of the Stokes equations. The boundary element method (BEM) based on the boundary integral formulation is thus one of the most accurate methods for simulating capsules under Stokes flow regime. A high computational cost of BEM, however, has limited its application to relatively small scale problems. We have developed a graphics process unit (GPU) computing of BEM for capsules and biological cells in Stokes flow. We have investigated rheological properties of capsules, and those of capsule suspensions using the GPU-accelerated BEM. Here, we provide an overview of our recent studies, particularly focusing on the shear viscosity of dense suspensions of capsules in simple shear flow; an overshoot phenomenon of the capsule deformation in oscillating shear flow; and the sedimentation of red blood cells.

1 Introduction

A capsule refers to a liquid droplet enclosed by a thin membrane [1]. Capsules are widely found in natural and engineering products. Red blood cells, for example, consist of a hemoglobin solution enclosed by a lipid-bilayer membrane. Artificial capsules are developed in food and pharmaceutical industries, and are also used for biomedical applications, where a substrate such as flavor and living cells is

Y. Imai (✉)
Graduate School of Engineering Science, Osaka University, Toyonaka, Japan

Graduate School of Engineering, Kobe University, Kobe, Japan
e-mail: yimai@me.es.osaka-u.ac.jp

D. Matsunaga
Graduate School of Engineering Science, Osaka University, Toyonaka, Japan
e-mail: daiki.matsunaga@me.es.osaka-u.ac.jp

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_6

encapsulated. Capsules are often highly deformable, change their shapes in response to fluid forces, and thus may rupture before reaching their targeting positions. The presence of capsules also alters the flow field, and affects the macroscopic rheology of capsule suspensions. Therefore, understanding the dynamics of capsules in flow is important for predicting not only the transport and breakup of capsules, but also the rheological properties of the suspension.

A difficulty in numerical simulations of the capsule dynamics arises from fluid–membrane interaction. The fluid mechanics of the internal and external fluids is coupled with the solid mechanics of the membrane. The kinematic and dynamic boundary conditions must be imposed at interfaces inside and outside of the nearly-zero-thickness membrane. These conditions are often given by no-slip and stress jump conditions on the membrane. Another difficulty arises from hydrodynamic interaction between capsules. Capsules in flow change their positions arbitrarily, and a capsule approaches other capsules within a very small separation distance. We may encounter these difficulties, particularly in mesh resolution for immersed boundary methods or in mesh generation for finite element methods.

Cells and artificial capsules considered here are relatively small. The radius of red blood cells is $a \approx 4\,\mu\mathrm{m}$. Shear rate in microvascular blood flow is $\dot{\gamma} \approx 10^2$–$10^3\ \mathrm{s}^{-1}$. The particle Reynolds number is then estimated as

$$Re = \frac{\rho \dot{\gamma} a^2}{\mu} \approx \frac{\left(10^3\right) \cdot \left(5 \times 10^2\right) \cdot \left(4 \times 10^{-6}\right)^2}{10^{-3}} = 8 \times 10^{-3}, \tag{1}$$

where ρ and μ are the density and viscosity of the external fluid. When $Re \ll 1$, inertial terms can be ignored in the Navier–Stokes equations, and the velocity field is given by the boundary integral formulation of the Stokes equations [2]. For such Stokes flow problems, the boundary element method (BEM) based on the boundary integral formulation can be the most accurate method. Both the no-slip and stress jump conditions are directly imposed in the formulation.

BEM has been used to investigate the dynamics of capsules in Stokes flow. Pioneering works were done by Pozrikidis and colleagues [3, 4], in which BEM was applied to clarify large deformation of a spherical capsule in simple shear flow. Because the membrane tension is linked to the rupture of the capsule, it has also been studied in detail using BEM [5–7]. These studies have been extended to ellipsoidal capsules [8–10] and red blood cells [11, 12]. A drawback of BEM is its heavy computational load. It may take several hours to simulate even a single capsule, and the computational time increases with the square of the computational nodes, or the number of capsules. Therefore, the application of BEM has been limited to relatively small scale problems, for example, a single capsule was often simulated for a short time period. We have developed a graphics processing unit (GPU) computing of BEM for simulating the dynamics of capsules and biological cells in Stokes flow [13]. We have investigated rheological properties of capsules, and those of capsule suspensions using the GPU-accelerated BEM. In this chapter, we provide an overview of our recent studies, particularly focusing on the shear viscosity of dense suspensions of capsules in simple shear flow [14]; an overshoot

phenomenon of the capsule deformation in oscillating shear flow [15]; and the sedimentation of red blood cells [16].

2 Governing Equations and Numerical Methods

2.1 Fluid Mechanics

2.1.1 Boundary Integral Formulation

We first consider flow in a control volume V bounded by closed surfaces S as shown in Fig. 1, where the normal vector n is directed into the control volume [2, 17]. When $Re \ll 1$, inertial terms can be ignored in the Navier–Stokes equations, and we have the Stokes equations:

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

and

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{3}$$

where $\mathbf{v}$ is the velocity vector, and $\boldsymbol{\sigma}$ is the stress tensor. The stress tensor for Newtonian fluids is given by

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mu \left\{ \nabla \otimes \mathbf{v} + (\nabla \otimes \mathbf{v})^T \right\}, \tag{4}$$

where μ is the viscosity of the fluid. The velocity field $\mathbf{v}(\mathbf{x})$ in the control volume is described by the boundary integral formulation of the Stokes equations [2, 17]:

$$\begin{aligned} \mathbf{v}(\mathbf{x}) = &- \frac{1}{8\pi\mu} \int_S \mathbf{J}(\mathbf{x}, \mathbf{y}) \cdot \boldsymbol{\sigma}(\mathbf{y}) \cdot \mathbf{n}(\mathbf{y}) \, dS(\mathbf{y}) \\ &+ \frac{1}{8\pi} \int_S \mathbf{v}(\mathbf{y}) \cdot \mathbf{K}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{n}(\mathbf{y}) \, dS(\mathbf{y}), \end{aligned} \tag{5}$$

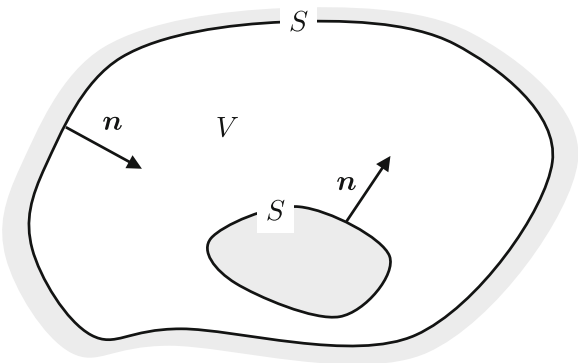


Fig. 1 A control volume V is bounded by closed surfaces S, where the normal vector $\mathbf{n}$ is directed to the control volume

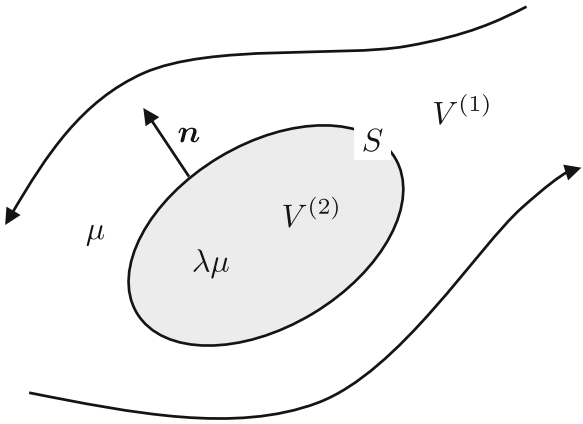


Fig. 2 A capsule with volume $V^{(2)}$ and surface S is suspended in unbounded flow with volume $V^{(1)}$. Viscosities of the external and internal fluids of the capsule are μ and $\lambda\mu$, respectively. The normal vector $\mathbf{n}$ is defined as a unit vector pointing outwards

where $\mathbf{x}$ and $\mathbf{y}$ are the position vectors, and $\mathbf{J}$ and $\mathbf{K}$ are the Green's functions. For infinite unbounded flow, the free-space Green's functions are

$$\mathbf{J}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{I}}{r} + \frac{\mathbf{r}\otimes\mathbf{r}}{r^3}, \tag{6}$$

and

$$\mathbf{K}(\mathbf{x},\mathbf{y}) = -6\frac{\mathbf{r}\otimes\mathbf{r}\otimes\mathbf{r}}{r^5}, \tag{7}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{x}$, and $r = |\mathbf{r}|$. The free-space Green's function $\mathbf{J}$ is also called the Stokeslet or the Oseen–Burgers tensor, and $\mathbf{K}$ is the stress tensor.

We then consider a capsule suspended in an unbounded flow of $\mathbf{v}^{\infty}(\mathbf{x})$ as illustrated in Fig. 2, where the viscosity of the external fluid is μ, and that of the internal fluid is $\lambda\mu$. The velocity field outside the capsule is given by the boundary integral formulation of the external flow $\mathbf{v}^{(1)}(\mathbf{y})$ with associated stress tensor $\boldsymbol{\sigma}^{(1)}(\mathbf{y})$,

$$\begin{aligned}\mathbf{v}(\mathbf{x}) = \mathbf{v}^{\infty}(\mathbf{x}) &- \frac{1}{8\pi\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\sigma}^{(1)}(\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) \\ &+\frac{1}{8\pi}\int_S \mathbf{v}^{(1)}(\mathbf{y})\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}),\end{aligned} \tag{8}$$

where the normal vector $\mathbf{n}$ is defined as a unit vector pointing outwards. The reciprocal identity also gives the boundary integral of the internal flow $\mathbf{v}^{(2)}(\mathbf{y})$ with associated stress tensor $\boldsymbol{\sigma}^{(2)}(\mathbf{y})$ as

$$\begin{aligned}0 = &+\frac{1}{8\pi\lambda\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\sigma}^{(2)}(\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) \\ &-\frac{1}{8\pi}\int_S \mathbf{v}^{(2)}(\mathbf{y})\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}).\end{aligned} \tag{9}$$

Combining these equations with the continuity of velocity $\mathbf{v}^{(1)}(\mathbf{y}) = \mathbf{v}^{(2)}(\mathbf{y})$, we have

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot[\boldsymbol{\sigma}(\mathbf{y})]\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) + \frac{1-\lambda}{8\pi}\int_S \mathbf{v}(\mathbf{y})\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}), \tag{10}$$

where $[\boldsymbol{\sigma}(\mathbf{y})] = \boldsymbol{\sigma}^{(1)}(\mathbf{y}) - \boldsymbol{\sigma}^{(2)}(\mathbf{y})$ is the stress jump at the membrane. The velocity field inside the capsule is also given by

$$\lambda\mathbf{v}(\mathbf{x}) = \lambda\mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot[\boldsymbol{\sigma}(\mathbf{y})]\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) + \frac{1-\lambda}{8\pi}\int_S \mathbf{v}(\mathbf{y})\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}). \tag{11}$$

The velocity of the capsule membrane is obtained from Eq. (10) or Eq. (11) when $\mathbf{x}$ approaches S. The boundary integral formulation then becomes

$$\frac{1+\lambda}{2}\mathbf{v}(\mathbf{x}) = \mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot[\boldsymbol{\sigma}(\mathbf{y})]\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) + \frac{1-\lambda}{8\pi}\int_S{}^{PV} \mathbf{v}(\mathbf{y})\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}), \tag{12}$$

where the superscript PV refers to the principal value integral over $S - \mathbf{x}$. The surface of the capsule is closed, and Eq. (12) can be rewritten by

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\int_S \mathbf{J}(\mathbf{x},\mathbf{y})\cdot[\boldsymbol{\sigma}(\mathbf{y})]\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}) + \frac{1-\lambda}{8\pi}\int_S \{\mathbf{v}(\mathbf{y}) - \mathbf{v}(\mathbf{x})\}\cdot\mathbf{K}(\mathbf{x},\mathbf{y})\cdot\mathbf{n}(\mathbf{y})\,dS(\mathbf{y}). \tag{13}$$

2.1.2 Image System

When the fluid domain is bounded by an infinite plane wall, the semi-infinite Green's functions with the image system [18] are used to satisfy the no-slip condition at the surface of the wall. Consider that the wall is located at y_3^w as shown in Fig. 3. The distance between the wall and a pole $\mathbf{y}$ is denoted by $h = y_3 - y_3^w$, and the image of the pole is $\mathbf{Y} = \mathbf{y} - 2\delta_{i3}h$. The semi-infinite Green's function of a Stokeslet consists of the free-space Green's function $\mathbf{J}^0$ and associated image system $\mathbf{J}^w$ as

$$\mathbf{J}(\mathbf{x},\mathbf{y}) = \mathbf{J}^0(\mathbf{x},\mathbf{y}) + \mathbf{J}^w(\mathbf{x},\mathbf{y}). \tag{14}$$

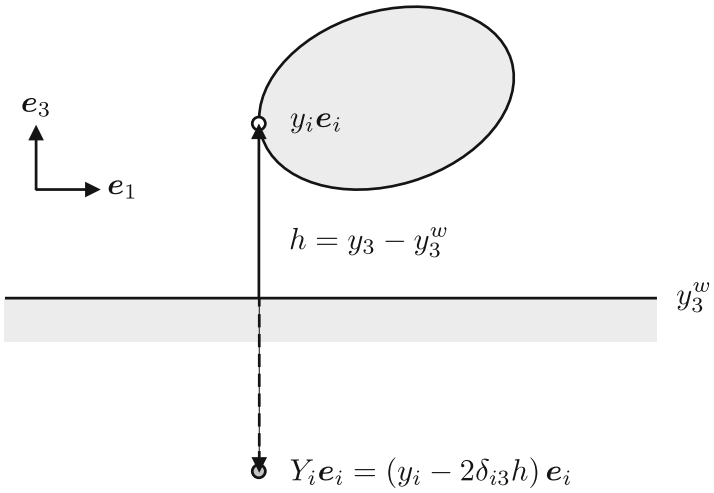


Fig. 3 A schematic of the image system. An infinite plane wall is located in the $\mathbf{e}_1$–$\mathbf{e}_2$ plane. The distance between a pole $\mathbf{y}$ and the wall is h, and the image of the pole is $\mathbf{Y}$

They are written in the Einstein notation as

$$J_{ij}^0(\mathbf{x}, \mathbf{y}) = \frac{\delta_{ij}}{r} + \frac{r_i r_j}{r^3}, \tag{15}$$

and

$$J_{ij}^w(\mathbf{x}, \mathbf{y}) = -J_{ij}^0(\mathbf{x}, \mathbf{Y}) - 2h^2\left(1 - 2\delta_{j3}\right) M_{ij}^D(\mathbf{x}, \mathbf{Y}) + 2h\left(1 - 2\delta_{j3}\right) J_{i3j}^D(\mathbf{x}, \mathbf{Y}), \tag{16}$$

where

$$J_{ij}^0(\mathbf{x}, \mathbf{Y}) = \frac{\delta_{ij}}{R} + \frac{R_i R_j}{R^3}, \tag{17}$$

$$M_{ij}^D(\mathbf{x}, \mathbf{Y}) = -\frac{\delta_{ij}}{R^3} + \frac{3R_i R_j}{R^5}, \tag{18}$$

and

$$J_{i3j}^D(\mathbf{x}, \mathbf{Y}) = \frac{\delta_{i3}R_j - \delta_{3j}R_i - \delta_{ij}R_3}{R^3} + \frac{3R_i R_3 R_j}{R^5}, \tag{19}$$

are the image Stokeslet, the image source dipole, and the image Stokeslet dipole with $\mathbf{R} = \mathbf{Y} - \mathbf{x}$, and $R = |\mathbf{R}|$, respectively.

The free-space Green's function for the stress tensor $\mathbf{K}^0$ is written by a combination of a point source $\mathbf{M}^0$ and Stokeslet dipoles,

$$K_{ijk}^{0}(\mathbf{x},\mathbf{y}) = -2\delta_{ik}M_{j}^{0}(\mathbf{x},\mathbf{y}) - \left\{J_{ijk}^{D}(\mathbf{x},\mathbf{y}) + J_{kji}^{D}(\mathbf{x},\mathbf{y})\right\}$$
$$= -\frac{6r_i r_j r_k}{r^5}, \tag{20}$$

and its image system also consists of an image source and image Stokeslet dipoles [19, 20],

$$K_{ijk}^{w}(\mathbf{x},\mathbf{y}) = -2\delta_{ik}M_{j}^{w}(\mathbf{x},\mathbf{y}) - \left\{J_{ijk}^{D,w}(\mathbf{x},\mathbf{y}) + J_{kji}^{D,w}(\mathbf{x},\mathbf{y})\right\}. \tag{21}$$

The image source and the image Stokeslet dipoles are given by

$$M_{j}^{w}(\mathbf{x},\mathbf{y}) = M_{j}^{0}(\mathbf{x},\mathbf{Y}) - 2hM_{3j}^{D}(\mathbf{x},\mathbf{Y}) + 2J_{j33}^{D}(\mathbf{x},\mathbf{Y}), \tag{22}$$

and

$$\begin{aligned}
&J_{ijk}^{D,w}(\mathbf{x},\mathbf{y}) + J_{kji}^{D,w}(\mathbf{x},\mathbf{y}) \\
&= -(1-2\delta_{k3})J_{ijk}^{D}(\mathbf{x},\mathbf{Y}) - (1-2\delta_{i3})J_{kji}^{D}(\mathbf{x},\mathbf{Y}) \\
&\quad -2\delta_{3k}(1-2\delta_{j3})\left\{2hM_{ij}^{D}(\mathbf{x},\mathbf{Y}) - J_{i3j}^{D}(\mathbf{x},\mathbf{Y})\right\} \\
&\quad -2\delta_{3i}(1-2\delta_{j3})\left\{2hM_{kj}^{D}(\mathbf{x},\mathbf{Y}) - J_{k3j}^{D}(\mathbf{x},\mathbf{Y})\right\} \\
&\quad -2(1-2\delta_{k3})(1-2\delta_{j3})\left\{h^2M_{ijk}^{Q}(\mathbf{x},\mathbf{Y}) + h^2M_{kji}^{Q}(\mathbf{x},\mathbf{Y})\right\} \\
&\quad +2(1-2\delta_{k3})(1-2\delta_{j3})\left\{hJ_{i3jk}^{Q}(\mathbf{x},\mathbf{Y}) + hJ_{k3ji}^{Q}(\mathbf{x},\mathbf{Y})\right\},
\end{aligned} \tag{23}$$

where

$$M_{j}^{0}(\mathbf{x},\mathbf{Y}) = \frac{R_j}{R^3}, \tag{24}$$

$$M_{ij}^{D}(\mathbf{x},\mathbf{Y}) = -\frac{\delta_{ij}}{R^3} + 3\frac{R_i R_j}{R^5}, \tag{25}$$

and

$$M_{ijk}^{Q}(\mathbf{x},\mathbf{Y}) = -3\frac{\delta_{ij}R_k + \delta_{jk}R_i + \delta_{ik}R_j}{R^5} + 15\frac{R_i R_j R_k}{R^7}, \tag{26}$$

are the image source, the image source dipole, and the image source quadrupole, and

$$J_{ijk}^{D}(\mathbf{x},\mathbf{Y}) = \frac{\delta_{ij}R_k - \delta_{jk}R_i - \delta_{ik}R_j}{R^3} + 3\frac{R_i R_j R_k}{R^5}, \tag{27}$$

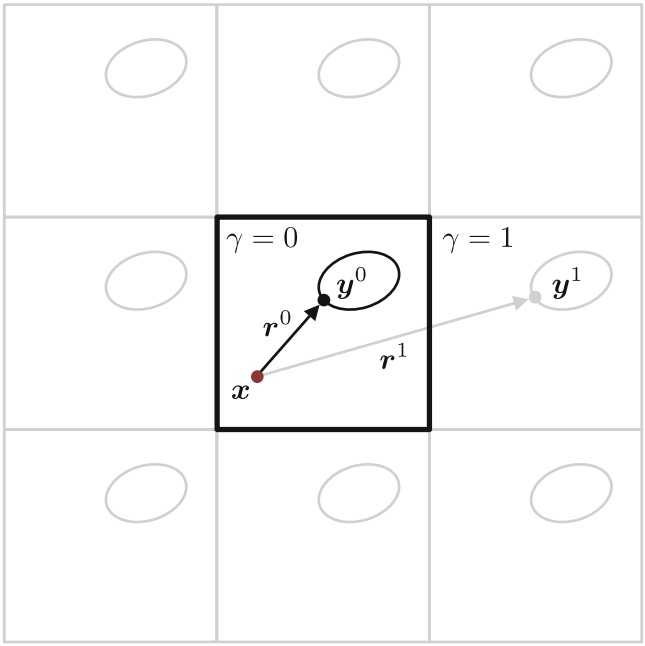


Fig. 4 A schematic of the Eward summation. A capsule is suspended in an original lattice $\gamma = 0$, where γ is the lattice index

and

$$J^{Q}_{ijkl}(\mathbf{x}, \mathbf{Y}) = -3\frac{\delta_{kl}R_iR_j + \delta_{jl}R_iR_k + \delta_{il}R_jR_k}{R^5} + 15\frac{R_iR_jR_kR_l}{R^7}, \tag{28}$$

are the image Stokeslet dipole, and the image Stokeslet quadrupole.

2.1.3 Eward Summation

The Eward summation proposed by Beenakker [21] is used for an infinite periodic domain, where the summation is decomposed into the summation of a real space and that of a reciprocal space. A schematic of a periodic domain is shown in Fig. 4. For simplicity, we consider that a capsule with viscosity ratio $\lambda = 1$ is suspended in the original lattice, $\gamma = 0$, where γ is the lattice index. The boundary integral formulation is then given by

$$\begin{aligned}\mathbf{v}(\mathbf{x}) &= \mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\sum_{\gamma}^{\infty}\int_{S^\gamma}\mathbf{J}(\mathbf{x}, \mathbf{y}^\gamma)\cdot[\boldsymbol{\sigma}(\mathbf{y}^\gamma)]\cdot\mathbf{n}(\mathbf{y}^\gamma)dS(\mathbf{y}^\gamma)\\ &= \mathbf{v}^{\infty}(\mathbf{x}) - \frac{1}{8\pi\mu}\int_{S}\mathbf{J}^{E}(\mathbf{x}, \mathbf{y}^0)\cdot[\boldsymbol{\sigma}(\mathbf{y}^0)]\cdot\mathbf{n}(\mathbf{y}^0)dS(\mathbf{y}^0).\end{aligned} \tag{29}$$

$\mathbf{J}^E$ is the Green's function of the periodic system,

$$J_{ij}^E(\mathbf{x}, \mathbf{y}^0) = \sum_\gamma J_{ij}^{E_I}(\mathbf{r}^\gamma) + \frac{8\pi}{V} \sum_{\lambda \neq 0} J_{ij}^{E_{II}}(\mathbf{k}^\lambda) \tag{30}$$

where

$$J_{ij}^{E_I}(\mathbf{r}) = \frac{\delta_{ij}}{r} E_1 + \frac{r_i r_j}{r^3} E_2, \tag{31}$$

$$J_{ij}^{E_{II}}(\mathbf{k}) = \left(\frac{\delta_{ij}}{k^2} - \frac{k_i k_j}{k^4} \right) \left(1 + \frac{k^2}{4\xi^2} + \frac{k^4}{8\xi^4} \right) \exp\left(-\frac{k^2}{4\xi^2} \right) \cos(\mathbf{k} \cdot \mathbf{r}), \tag{32}$$

and

$$E_1 = \operatorname{erfc}(\xi r) + \frac{\exp(-\xi^2 r^2)}{\sqrt{\pi}} (4\xi^3 r^3 - 6\xi r), \tag{33}$$

$$E_2 = \operatorname{erfc}(\xi r) + \frac{\exp(-\xi^2 r^2)}{\sqrt{\pi}} (2\xi r - 4\xi^3 r^3). \tag{34}$$

2.1.4 Multipole Expansion

For dense suspensions of capsules, the direct integral of the whole capsules becomes a difficult task because of its heavy computational load. We then employ multipole expansion [22]. The Taylor series of the Green's function with respect to $\mathbf{y}$ around an arbitrary position $\mathbf{y}^\alpha$ is given by

$$\mathbf{J}(\mathbf{x}, \mathbf{y}) = \mathbf{J}(\mathbf{x}, \mathbf{y})\big|_{\mathbf{y}=\mathbf{y}^\alpha} + \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{y}^\alpha} \left(\mathbf{y} - \mathbf{y}^\alpha\right) + \cdots. \tag{35}$$

The boundary integral is then rewritten by

$$\begin{aligned}
&\int_S \mathbf{J}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{q}(\mathbf{y}) \, dS(\mathbf{y}) \\
&= \sum_\alpha \int_{S_\alpha} \left\{ \mathbf{J}(\mathbf{x}, \mathbf{y})\big|_{\mathbf{y}=\mathbf{y}^\alpha} + \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{y}^\alpha} \left(\mathbf{y} - \mathbf{y}^\alpha\right) + \cdots \right\} \cdot \mathbf{q}(\mathbf{y}) \, dS^\alpha \\
&= \sum_\alpha \left\{ \mathbf{J}^\alpha \int_{S_\alpha} \mathbf{q}(\mathbf{y}) \, dS^\alpha + \frac{\partial \mathbf{J}^\alpha}{\partial \mathbf{y}} \int_{S^\alpha} \left(\mathbf{y} - \mathbf{y}^\alpha\right) \cdot \mathbf{q}(\mathbf{y}) \, dS^\alpha \cdots \right\},
\end{aligned} \tag{36}$$

where

$$\mathbf{J}^\alpha = \mathbf{J}(\mathbf{x}, \mathbf{y})\big|_{\mathbf{y}=\mathbf{y}^\alpha}, \frac{\partial \mathbf{J}^\alpha}{\partial \mathbf{y}} = \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{y}^\alpha} \left(\mathbf{y} - \mathbf{y}^\alpha\right), \cdots, \tag{37}$$

$\mathbf{q} = [\boldsymbol{\sigma}] \cdot \mathbf{n}$ and α is the pole index. The first term in the brackets of Eq. (36) represents flow due to a point force $\mathbf{F}^\alpha(\mathbf{y}^\alpha)$, given by

$$J_{ij}^{\alpha} \int_{S_\alpha} q_j dS^{\alpha} = J_{ij}^{\alpha} F_j^{\alpha}, \tag{38}$$

where

$$F_j^{\alpha} = \int_{S_\alpha} q_j dS^{\alpha}. \tag{39}$$

The second term is flow due to a point torque $\boldsymbol{\Omega}^{\alpha}(\mathbf{y}^{\alpha})$ and a point stresslet $\mathbf{S}^{\alpha}(\mathbf{y}^{\alpha})$,

$$\frac{\partial J_{ij}^{\alpha}}{\partial y_k} \int_{S^\alpha} \left(y_k - y_k^{\alpha}\right) q_j dS^{\alpha} = R_{ij}^{\alpha} \Omega_j^{\alpha} + L_{ijk}^{\alpha} S_{jk}^{\alpha} \tag{40}$$

where

$$\Omega_j^{\alpha} = \int_{S_\alpha} \varepsilon_{jkl} \left(y_k - y_k^{\alpha}\right) q_l dS^{\alpha}, \tag{41}$$

and

$$S_{jk}^{\alpha} = \frac{1}{2} \int_{S^\alpha} \left\{ \left(y_j - y_j^{\alpha}\right) q_k + \left(y_k - y_k^{\alpha}\right) q_j - \frac{2}{3} \delta_{jk} \left(y_k - y_k^{\alpha}\right) q_l \right\} dS^{\alpha}. \tag{42}$$

$\mathbf{R}^{\alpha}$ and $\mathbf{L}^{\alpha}$ are the Green's functions,

$$R_{ij} = \frac{1}{4} \varepsilon_{ilk} \left\{ J_{ljk}^{D}(\mathbf{x}, \mathbf{y}) - J_{kjl}^{D}(\mathbf{x}, \mathbf{y}) \right\} = \varepsilon_{ijk} \frac{r_k}{r^3}, \tag{43}$$

and

$$L_{ijk} = \frac{1}{2} \left\{ J_{ijk}^{D}(\mathbf{x}, \mathbf{y}) + J_{kji}^{D}(\mathbf{x}, \mathbf{y}) \right\} = -\frac{r_j \delta_{ik}}{r^3} + 3 \frac{r_i r_j r_k}{r^5}, \tag{44}$$

at the pole $\mathbf{y}^{\alpha}$. Equation (36) then reduces to

$$\int_S J_{ij} q_j dS = \sum_{\alpha} \left(J_{ij}^{\alpha} F_j^{\alpha} + R_{ij}^{\alpha} \Omega_j^{\alpha} + L_{ijk}^{\alpha} S_{jk}^{\alpha} \right) + O(r^{-3}). \tag{45}$$

2.2 Membrane Mechanics

2.2.1 Membrane Tension

Consider a material point on the surface of a capsule membrane [1]. Let $\mathbf{X} = X^I \mathbf{e}_I$ and $\mathbf{x} = x^I \mathbf{e}_I$ be the position vectors of the material point in the reference and deformed states of the capsule, respectively, where $\mathbf{e}_I$ is the Cartesian base and $I = 1, 2, 3$. We introduce local curvilinear coordinates with local covariant bases,

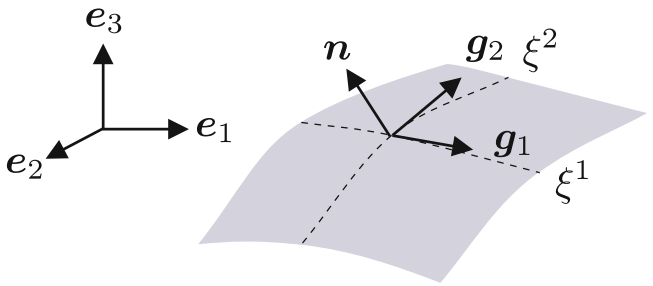


Fig. 5 Local curvilinear coordinates at a deformed state of a capsule membrane with local covariant bases, $\mathbf{g}_1$, $\mathbf{g}_2$, and $\mathbf{n}$. The vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are the Cartesian bases

$$\mathbf{g}_1 = \frac{\partial \mathbf{x}}{\partial \xi^1}, \quad \mathbf{g}_2 = \frac{\partial \mathbf{x}}{\partial \xi^2}, \quad \mathbf{g}_3 = \mathbf{n}, \tag{46}$$

$$\mathbf{G}_1 = \frac{\partial \mathbf{X}}{\partial \xi^1}, \quad \mathbf{G}_2 = \frac{\partial \mathbf{X}}{\partial \xi^2}, \quad \mathbf{G}_3 = \mathbf{N}, \tag{47}$$

where $\mathbf{n}$ and $\mathbf{N}$ are the unit normal vectors in the deformed and reference states (see also Fig. 5). Local contravariant bases are defined as $\mathbf{g}^\alpha \cdot \mathbf{g}_\beta = \delta^\alpha_\beta$, $\mathbf{g}^3 = \mathbf{n}$, $\mathbf{G}^\alpha \cdot \mathbf{G}_\beta = \delta^\alpha_\beta$, and $\mathbf{G}^3 = \mathbf{N}$, where $\alpha = 1, 2$, and $\beta = 1, 2$.

A vector on the membrane $d\mathbf{X} = d\xi^\alpha \mathbf{G}_\alpha$ is transformed into $d\mathbf{x} = d\xi^\alpha \mathbf{g}_\alpha$ as

$$d\mathbf{x} = \mathbf{F}_s \cdot d\mathbf{X}, \tag{48}$$

where $\mathbf{F}_s$ is the surface deformation gradient tensor given by

$$\mathbf{F}_s = \mathbf{g}_\alpha \otimes \mathbf{G}^\alpha. \tag{49}$$

The right Cauchy–Green deformation tensor is then

$$\begin{aligned} \mathbf{C} &= \mathbf{F}_s^T \cdot \mathbf{F}_s \\ &= g_{\alpha\beta}(\mathbf{G}^\alpha \otimes \mathbf{G}^\beta), \end{aligned} \tag{50}$$

where $g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta$ is the covariant metric tensor. The Green–Lagrange strain tensor is given by

$$\begin{aligned} \mathbf{e} &= \frac{1}{2}\{\mathbf{C} - (\mathbf{I} - \mathbf{N} \otimes \mathbf{N})\} \\ &= \frac{1}{2}\left\{g_{\alpha\beta}(\mathbf{G}^\alpha \otimes \mathbf{G}^\beta) - G_{\alpha\beta}(\mathbf{G}^\alpha \otimes \mathbf{G}^\beta)\right\}, \end{aligned} \tag{51}$$

and its invariants are

$$\begin{aligned} I_1 &= 2tr(\mathbf{e}) \\ &= g_{\alpha\beta}G^{\alpha\beta} - 2 \\ &= \lambda_1^2 + \lambda_2^2 - 2, \end{aligned} \tag{52}$$

and

$$\begin{aligned} I_2 &= 2\det(\mathbf{e}) \\ &= |g_{\alpha\beta}||G^{\alpha\beta}| - 1 \\ &= \lambda_1^2\lambda_2^2 - 1 \\ &= J_s^2 - 1, \end{aligned} \tag{53}$$

where λ_1 and λ_2 are the principal extension ratios, and $J_s = \lambda_1\lambda_2 = \det(\mathbf{F}_s)$ is the area dilation ratio. Finally, the Cauchy stress tensor is described as

$$\begin{aligned} \mathbf{T} &= \frac{1}{J_s}\mathbf{F}_s \cdot \frac{\partial w_s}{\partial \mathbf{e}} \cdot \mathbf{F}_s^T \\ &= \frac{2}{J_s}\frac{\partial w_s}{\partial I_1}G^{\alpha\beta}(\mathbf{g}_\alpha \otimes \mathbf{g}_\beta) + 2J_s\frac{\partial w_s}{\partial I_2}g^{\alpha\beta}(\mathbf{g}_\alpha \otimes \mathbf{g}_\beta), \end{aligned} \tag{54}$$

and thus,

$$T^{\alpha\beta}(\mathbf{g}_\alpha \otimes \mathbf{g}_\beta) = \left(\frac{2}{J_s}\frac{\partial w_s}{\partial I_1}G^{\alpha\beta} + 2J_s\frac{\partial w_s}{\partial I_2}g^{\alpha\beta}\right)(\mathbf{g}_\alpha \otimes \mathbf{g}_\beta), \tag{55}$$

where w_s is the strain energy function.

The neo-Hookean constitutive law is widely used to describe the behavior of a thin sheet of an isotropic volume-incompressible rubber-like material. The strain energy function of the neo-Hookean membrane is given by

$$w_s = \frac{G_s}{2}\left(I_1 - 1 + \frac{1}{I_2 + 2}\right), \tag{56}$$

where G_s is the surface shear elastic modulus. Substituting Eq. (56) into Eq. (55), we have

$$T^{\alpha\beta} = G_s\left\{\frac{1}{J_s}G^{\alpha\beta} - \frac{J_s}{(I_2+1)^2}g^{\alpha\beta}\right\}. \tag{57}$$

Skalak et al. [23] proposed a constitutive law to model biological membrane. The strain energy function of Skalak's law is

$$w_s = \frac{G_s}{4}\left(I_1^2 + 2I_1 - 2I_2 + CI_2^2\right), \tag{58}$$

and the Cauchy stress tensor is

$$T^{\alpha\beta} = G_s \left\{ \frac{1}{J_s} (I_1 + 1) G^{\alpha\beta} + J_s (CI_2 - 1) g^{\alpha\beta} \right\}, \tag{59}$$

where C is the area dilation modulus.

2.2.2 Equilibrium Equation

Inertial terms are also ignored for the solid mechanics of the membrane. A weak form of the equilibrium equation is then given by

$$\int_S \hat{\mathbf{u}} \cdot \mathbf{q}_s dS = \int_S \hat{\boldsymbol{\varepsilon}} : \mathbf{T} dS, \tag{60}$$

where $\hat{\mathbf{u}}$ is the virtual displacement, $\hat{\boldsymbol{\varepsilon}}$ is the virtual strain, and $\mathbf{q}_s$ is the load on the membrane.

2.2.3 Bending Stiffness

For bending stiffness of the membrane, Helfrich [24] proposed a bending energy

$$w_b = \frac{E_b}{2} \int_A (2H - c_0) dS, \tag{61}$$

where E_b is the bending modulus, H is the mean curvature of the membrane surface, and c_0 is the reference curvature. The bending energy is converted to a membrane load [25],

$$\mathbf{q}_b = -2E_b \left\{ \Delta_S H + (2H + c_0)(H^2 - K - c_0 H) \right\} \mathbf{n}, \tag{62}$$

where Δ_S is the Laplace–Beltrami operator on the surface, and K is the Gaussian curvature.

2.3 Boundary Conditions

The membrane velocity is given by the kinematic condition as

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}). \tag{63}$$

The load on the membrane $\mathbf{q} = \mathbf{q}_s + \mathbf{q}_b$ is balanced by the stress jump condition as

$$\mathbf{q} = [\boldsymbol{\sigma}] \cdot \mathbf{n}. \tag{64}$$

2.4 Numerical Methods

The boundary element method for fluid mechanics is coupled with the finite element method for membrane mechanics [6]. An unstructured mesh with triangular elements is generated for the capsule membrane. The boundary integral equation (13) is solved by the boundary element method, where the Gaussian quadrature method is used for the integral over the elements. For singular elements, polar coordinates are introduced. The weak form of the equilibrium equation (60) is solved by the finite element method. Here, we use the explicit second-order Runge–Kutta method or the Clank–Nicolson method is used for the time integration.

All the procedures are fully implemented in GPU computing. BEM is a compute-bound process, with an arithmetic intensity [26] $\approx$ 100 FLOP/Byte, which is 100 times larger than the values of other numerical methods for computational fluid dynamics. The performance of the GPU-accelerated BEM thus approaches the ideal performance of GPU. For more details, please refer to [13–16].

3 Shear Viscosity of Capsule Suspensions in Simple Shear Flow

The shear viscosity of particulate suspensions increases with the volume fraction of particles. For a dilute suspension of rigid spheres, the shear viscosity may be given by a linear function of the volume fraction of the particles,

$$\mu_{sp} = \frac{5}{2}\phi, \tag{65}$$

called Einstein's viscosity [27], where μ_{sp} is an increase in viscosity due to the presence of the particles, and ϕ is the volume fraction. When the volume fraction is increased, because of hydrodynamic interaction between the particles, second or higher order terms appear. The shear viscosity then increases nonlinearly with the volume fraction.

If the particles are able to deform, how does the deformability of the particles affect the suspension viscosity? It is known that the shear viscosity exhibits a shear thinning behavior for capsule suspensions [28], and the shear thinning property is pronounced when the volume fraction is increased [29]. We further investigated the rheology of a capsule suspension in simple shear flow using the GPU-accelerated BEM [14]. We found that the effect of high-order terms on the shear viscosity of

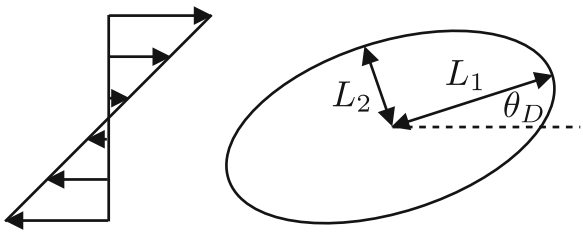


Fig. 6 A schematic of a capsule in simple shear flow. A capsule deforms into an ellipsoidal shape with major and minor radii L_1 and L_2 and an orientation angle θ_D with respect to the flow direction

the capsule suspension is much smaller than that of the rigid-particle suspension. We also discussed how the macroscopic viscosity is related to the deformation and orientation of the capsules in the suspension.

3.1 Deformation and Orientation of Capsules

Consider a spherical capsule with radius a suspended in simple shear flow. The capsule is assumed to be neutrally buoyant, and the internal fluid has the same viscosity as the external fluid ($\lambda = 1$). The neo-Hookean law is used for the constitutive equation of the capsule membrane. The capillary number defined by

$$Ca = \frac{\mu \dot{\gamma} a}{G_s} \tag{66}$$

is the ratio of the viscous force to the elastic force, where μ is the fluid viscosity, $\dot{\gamma}$ is the shear rate, and G_s is the surface shear elastic modulus of the membrane.

In simple shear flow, the capsule deforms into an ellipsoidal shape at an orientation angle θ_D with respect to the flow direction (Fig. 6), and exhibits "tank-tread motion," characterized by membrane rotation. The Taylor parameter is often used to quantify the extent of the capsule deformation:

$$D_{12} = \frac{|L_1 - L_2|}{|L_1 + L_2|}, \tag{67}$$

where L_1 and L_2 are major and minor lengths of the capsule in the shear plane. We use the inertia tensor,

$$\begin{aligned} \mathbf{A} &= \int_V [(\mathbf{x} \cdot \mathbf{x})\, \mathbf{I} - \mathbf{x} \otimes \mathbf{x}]\, dV \\ &= \frac{1}{5} \int_S \mathbf{n} \cdot [(\mathbf{x} \cdot \mathbf{x})\, \mathbf{x} \otimes \mathbf{I} - \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}]\, dS, \end{aligned} \tag{68}$$

to approximate the capsule shape as an ellipsoid.

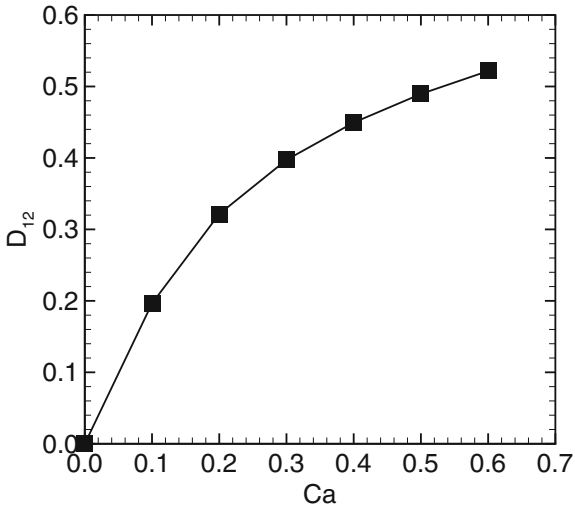


Fig. 7 The Taylor parameter D_{12} for a neo-Hookean capsule in simple shear flow as a function of the capillary number Ca

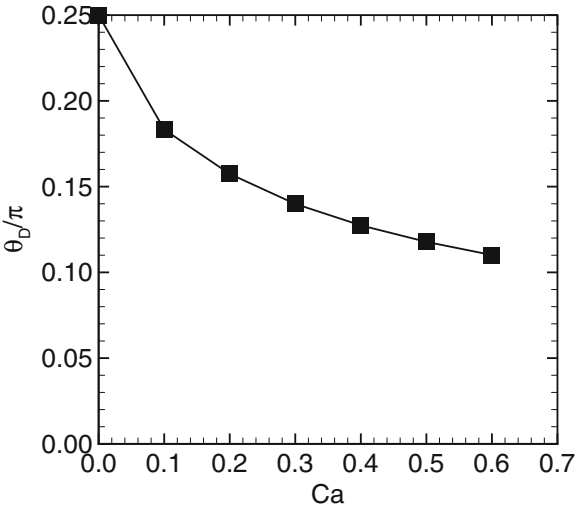


Fig. 8 Orientation angle of a neo-Hookean capsule in simple shear flow with respect to the flow direction

The Taylor parameter for a neo-Hookean capsule is shown as a function of the capillary number in Fig. 7. When the capillary number is increased, the capsule is more elongated, and the Taylor parameter is increased. The orientation angle of the capsule also changes with the capillary number. The orientation angle is lower for higher capillary numbers as shown in Fig. 8.

We then consider multiple capsules. How does the presence of other capsules affect the deformation and orientation of a capsule? To investigate the effect of hydrodynamic interaction between capsules, we simulate dense suspensions of capsules at volume fractions up to $\phi = 0.4$. Capsules are suspended in a triply periodic flow field under simple shear flow. The Eward summation is used to

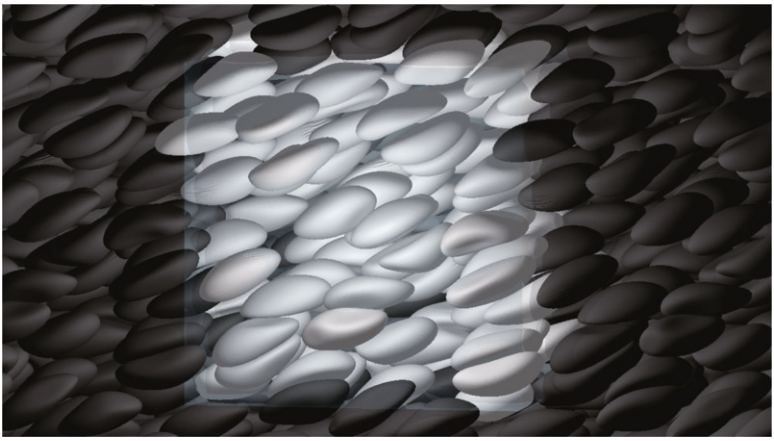


Fig. 9 A snapshot of neo-Hookean capsules in simple shear flow. The volume fraction of the capsules is $\phi = 0.3$ and the capillary number is $Ca = 0.3$. The bright cubic region shows the original lattice, and the dark region is Ewald periodic images. Note that periodic boxes around the original domain are moving with the applied shear

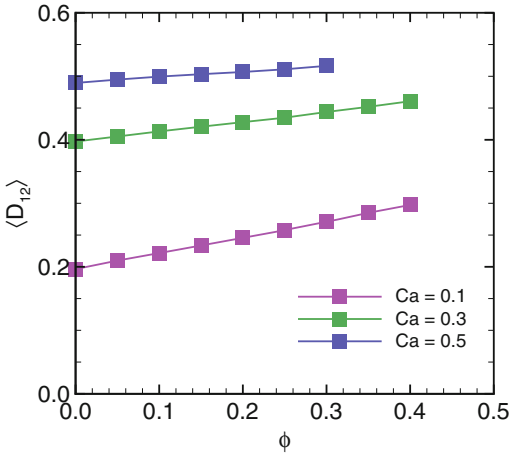


Fig. 10 Ensemble average of the Taylor parameter D_{12} as a function of the volume fraction of capsules ϕ. The deformation is increased as the volume fraction is increased

compute flow in an infinite periodic domain. Multipole expansion is also used for accelerating the computation. Figure 9 is a typical snapshot of dense suspensions of capsules in simple shear flow. The hydrodynamic interaction between the capsules is complex, and each capsule exhibits a different instantaneous deformation from the other capsules. We calculate the ensemble average of the Taylor parameter for the capsules in the suspension. Figure 10 shows that the capsule deformation is increased as the volume fraction is increased. The ensemble average of the orientation angle is also shown in Fig. 11. The orientation angle is decreased as the volume fraction is increased.

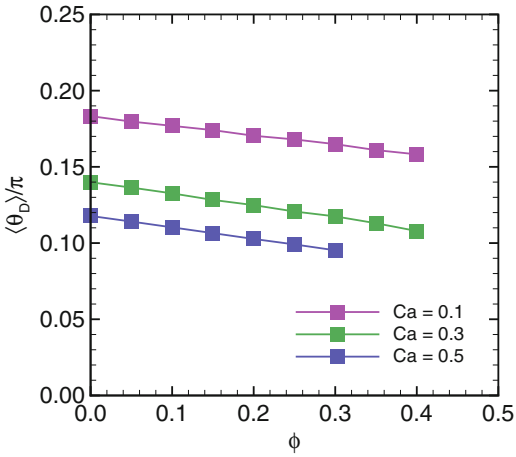


Fig. 11 Orientation angle of capsules with respect to the flow direction. The orientation angle is decreased as the volume fraction is increased

3.2 Suspension Viscosity

The deformation and orientation of capsules change with the volume fraction of capsules. These changes result in a non-Newtonian rheology of the capsule suspensions. We next consider the viscosity of capsule suspensions.

Particle stress is the particle contribution to the total stress of a suspension (see textbooks, for example, [30]). The particle stress tensor is given by Batchelor [31]

$$\boldsymbol{\Sigma}^p = n_c \langle \mathbf{S} \rangle , \tag{69}$$

where n_c is the particle number density. $\langle \mathbf{S} \rangle$ is the ensemble average of the hydrodynamic stresslet of the capsules:

$$\mathbf{S} = \int_S \frac{1}{2} \left(\mathbf{x} \otimes \mathbf{q} + \mathbf{q} \otimes \mathbf{x} \right) dS, \tag{70}$$

where $\mathbf{x}$ is the position vector, and $\mathbf{q}$ is the load on the capsule membrane. The specific viscosity μ_{sp} measures an increase in viscosity due to the particle shear stress,

$$\mu_{sp} = \frac{\Sigma^p_{12}}{\mu \dot{\gamma}}. \tag{71}$$

The specific viscosity of particle suspensions is often described by a polynomial equation of the volume fraction:

$$\mu_{sp} = a_1 \phi + a_2 \phi^2 + a_3 \phi^3 + \cdots. \tag{72}$$

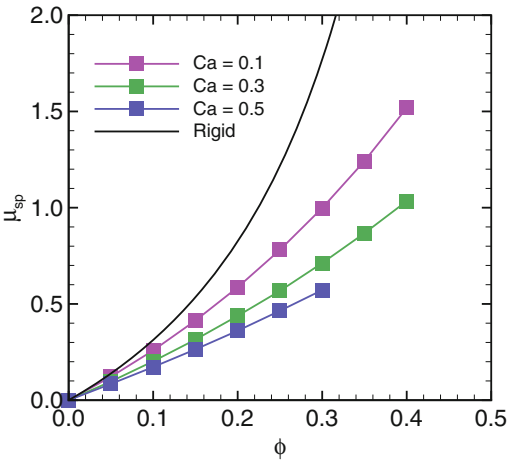


Fig. 12 Specific viscosity of the suspension of neo-Hookean capsules in simple shear flow as a function of the volume fraction of the capsules. The specific viscosity of a rigid-sphere suspension [32] is also shown

Einstein [27] derived the specific viscosity for the dilute limit of rigid-sphere suspensions as Eq. (65). Krierger and Dougherty [32] also proposed an empirical expression:

$$\mu_{sp} = \left(1 - \frac{\phi}{\phi_m}\right)^{-\frac{5}{2}\phi_m} - 1, \tag{73}$$

where ϕ_m is the maximum packing fraction. Using a Taylor series expansion of Eq. (73) around $\phi = 0$, we have

$$a_i = \prod_{n=1}^{i} \left\{ \frac{1}{n} \left(\frac{5}{2} + \frac{n-1}{\phi_m} \right) \right\}, \tag{74}$$

where $\prod$ is the product operator.

We compare the specific viscosity of a neo-Hookean capsule suspension with that of a rigid-sphere suspension in Fig. 12, where the maximum packing fraction is assumed to be $\phi_m = 0.63$ for random close packing [33]. The specific viscosities increase nonlinearly with the volume fractions both for the rigid-sphere and capsule suspensions. The values are lower for higher capillary numbers at a given volume fraction.

We obtain the coefficient for the first-order term, a_1, from the result of the dilute limit, and then estimate the coefficient of the second-order term, a_2, by a least-squares fitting to plots of $\left(\mu_{sp} - a_1\phi\right)$. Note that we set a range of the volume fraction as $0 \leq \phi \leq 0.1$ for the least-squares fitting. Figure 13 shows that both the coefficients a_1 and a_2 become smaller when the capillary number is increased, and a_2 decreases more rapidly than a_1.

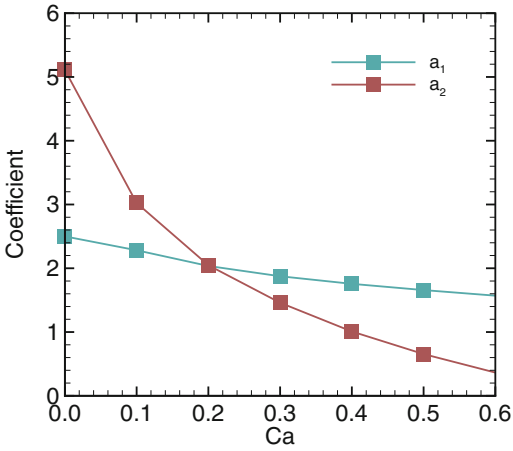


Fig. 13 Coefficients of polynomial equation (72) as a function of the capillary number

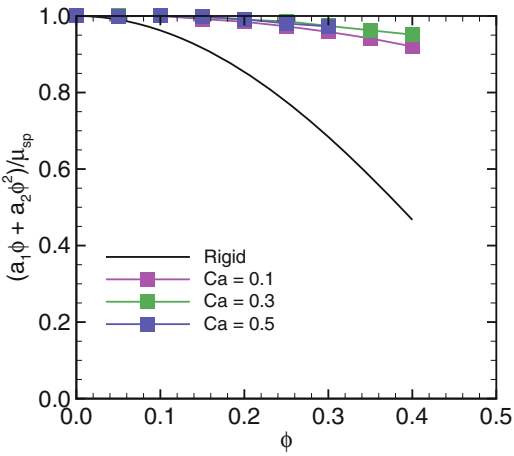


Fig. 14 Ratio of $(a_1\phi + a_2\phi^2)$ to μ_{sp}. The contribution of the first- and second-order terms on the shear viscosity is much larger for a capsule suspension than a rigid-sphere suspension

This result implies that the effect of high-order terms is small for deformable particles. We plot the ratio of $\left(a_1\phi + a_2\phi^2\right)$ to μ_{sp} in Fig. 14. We find that the contribution of the first- and second-order terms is much larger for capsule suspensions than rigid-sphere suspensions. While the first- and second-order terms only account for approximately 50% of μ_{sp} for the rigid-sphere suspension at $\phi = 0.4$, these terms account for 92% even for the capsule suspension with the lowest capillary number examined, $Ca = 0.1$.

3.3 Relationship Between the Deformation and Orientation and the Shear Viscosity

Non-Newtonian behaviors of particle suspensions are often discussed with the microstructure of suspensions [30]. The deformation and orientation of capsules

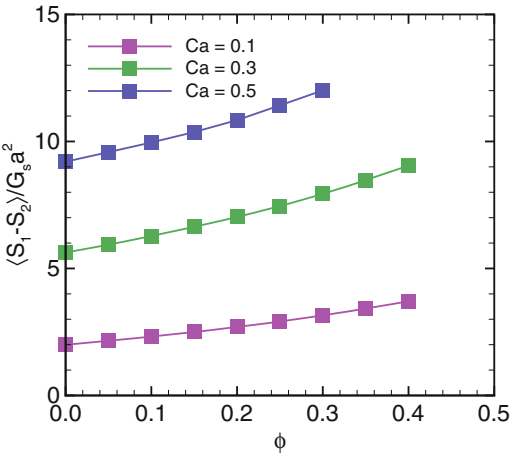


Fig. 15 Ensemble average of the principal stresslet difference of capsules in simple shear flow. The principal stresslet difference is increased when the volume fraction of the capsules is increased

may, however, more directly affect the suspension rheology and also the suspension microstructure. Here, we discuss the relationship between the deformation and orientation of capsules and the suspension rheology.

The specific viscosity is proportional to the shear component of the hydrodynamic stresslet of capsules. The shear component of the stresslet tensor of a capsule may be written as

$$S_{12} = \frac{1}{2}\left(S_1 - S_2\right)\sin 2\theta_S, \tag{75}$$

where S_1 and S_2 are the principal stresslets (the eigenvalues of the stresslet tensor) in the shear plane. The strength of the stresslet is given by the principal stresslet difference $S_1 - S_2$, and its direction is given by the orientation angle θ_S with respect to the flow direction. Equation (75) is valid only for dilute suspensions, but may be applicable to the ensemble average of the stresslets as an approximation,

$$\langle S_{12}\rangle \approx \frac{1}{2}\left\langle S_1 - S_2\right\rangle \sin 2\left\langle \theta_S\right\rangle . \tag{76}$$

When the viscosity ratio $\lambda = 1$, the principal stresslet difference is associated with the deformation of the capsules. When the volume fraction of the capsules is increased, the capsule deformation is increased. The principal stresslet difference is then also increased as shown in Fig. 15. The increase in the deformation thus amplifies the magnitude of the particle stress, resulting in an increase in the specific viscosity. However, the orientation angle of the capsules is also decreased at the same time, i.e. the capsule direction is closer to the flow direction. Figure 16 shows the principal direction of the stresslet. The direction of the stresslet also becomes closer to the flow direction, and the shear component of the stresslet tensor is decreased. Hence, the change in the orientation of capsules suppresses an increase in the shear viscosity.

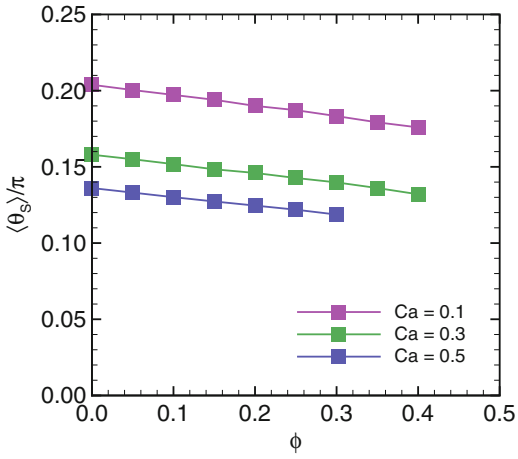


Fig. 16 Ensemble average of the principal direction of the stresslet of capsules, where the angle θ_S is defined as an angle with respect to the flow direction. When the volume fraction of the capsules is increased, the principal direction becomes closer to the flow direction

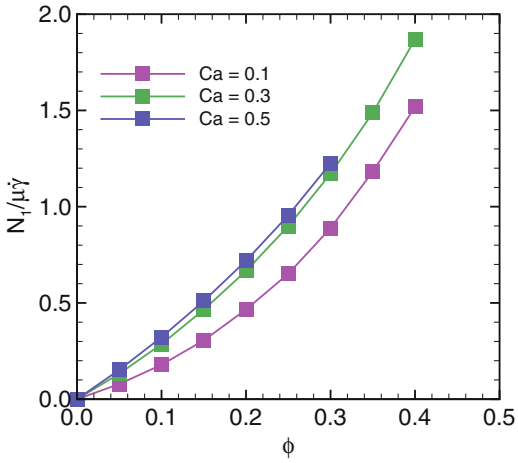


Fig. 17 First normal stress of the suspension of neo-Hookean capsules in simple shear flow. The first normal stress difference is larger for higher capillary numbers

In this relationship, the first normal stress difference,

$$\frac{N_1}{\mu\dot{\gamma}} = \frac{\Sigma^p_{11} - \Sigma^p_{22}}{\mu\dot{\gamma}}, \tag{77}$$

may be larger for higher capillary numbers, because the difference in the normal components of the stresslet is given by

$$\langle S_{11} - S_{22} \rangle \approx \langle S_1 - S_2 \rangle \cos 2 \langle \theta_S \rangle . \tag{78}$$

In Fig. 17, we find that the first normal stress difference is increased as the volume fraction is increased. The first normal stress difference becomes larger for higher capillary numbers.

4 Deformation of Capsules in Oscillating Shear Flow

We next consider the deformation of capsules in oscillating shear flow. While previous studies have clarified the dynamics of capsules in steady flows, only a few studies have investigated the capsule dynamics in unsteady flows [34]. We simulated the deformation of a neo-Hookean capsule in oscillating shear flow and found "overshoot phenomenon" in the deformation of the capsule [15].

Consider a neo-Hookean capsule suspended in oscillating shear flow. The shear rate of the oscillating shear flow is given in a form,

$$\dot{\gamma}\,(t) = \dot{\gamma}_0 \exp j\,(2\pi f t)\,, \tag{79}$$

where t is the time, $\dot{\gamma}_0$ is the amplitude of the shear rate, j is the imaginary unit, and f is the oscillation frequency. The corresponding strain is then

$$\gamma\,(t) = \int_t \dot{\gamma}\,(t)\,dt = \frac{\dot{\gamma}_0}{2\pi f j} \exp j\,(2\pi f t)\,. \tag{80}$$

The strain is inversely proportional to the oscillation frequency f, and thus the strain given to the capsule is small at high oscillation frequencies. Using the amplitude of the shear rate $\dot{\gamma}_0$, the capillary number is defined as

$$Ca = \frac{\mu \dot{\gamma}_0 a}{G_s}, \tag{81}$$

where μ is the viscosity of the external fluid.

For high oscillation frequencies, the capsule deformation may be predicted by the small deformation theory [28, 35]. In the small deformation theory, the Taylor parameter is given by

$$D_{12} = \frac{5}{4\pi j\,(f/\dot{\gamma}_0)\,(2\lambda + 3)} \exp j\,(2\pi f t)\,, \tag{82}$$

and the maximum Taylor parameter is

$$D_{12}^{max} = \frac{5}{4\pi\,(f/\dot{\gamma}_0)\,(2\lambda + 3)}, \tag{83}$$

where λ is the viscosity ratio between the internal and external fluids. The maximum Taylor parameter is inversely proportional to f. We also note that Eq. (83) does not include the elastic properties of the capsule membrane.

4.1 Deformation and Phase

Snapshots of the capsule in oscillating shear flow are shown in Fig. 18 for $Ca = 0.5$ and $\lambda = 5$ at various frequencies. The capsule largely deforms at oscillation

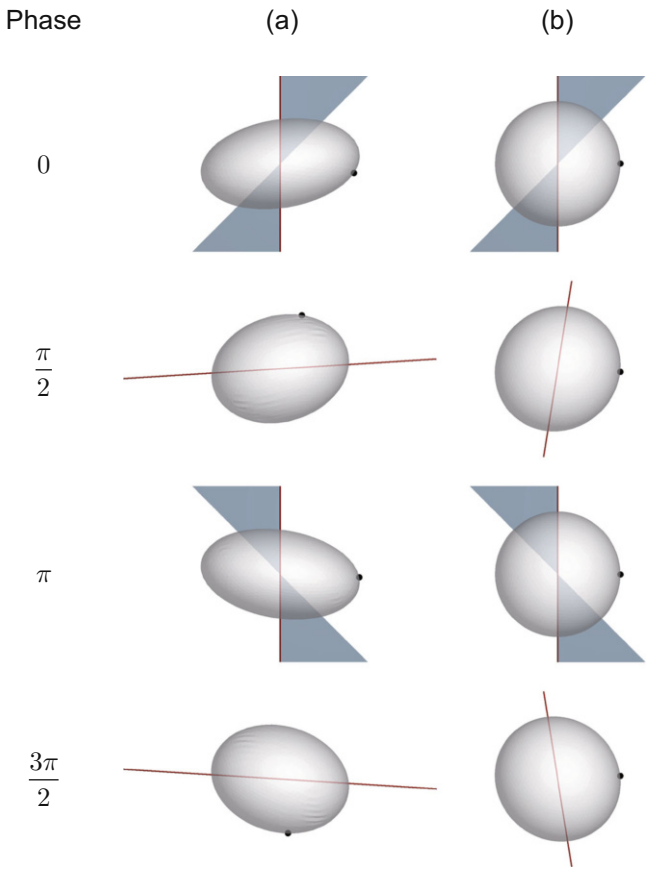


Fig. 18 Snapshots of a neo-Hookean capsule in oscillating shear flow for $Ca = 0.5$ and $\lambda = 5$ at oscillation frequency (**a**) $f/\dot{\gamma}_0 = 10^{-2}$, and (**b**) $f/\dot{\gamma}_0 = 10^{0}$. The red line shows the strain, and the blue triangle is the shear rate

phases 0 and π, where the shear rate is maximum, at a low oscillation frequency $f/\dot{\gamma}_0 = 10^{-2}$. When the oscillation frequency is increased, the capsule deformation is decreased. The deformation peaks at phases approximately $\pi/2$ and $3\pi/2$, where the strain is maximum.

We plot the maximum Taylor parameter with the small deformation theory, Eq. (83), in Fig. 19 as a function of the oscillation frequency. At low frequencies, the maximum Taylor parameter is larger for higher capillary numbers, but weakly depends on the oscillation frequency. When the oscillation frequency is increased, the maximum Taylor parameter converges to values predicted by the small deformation theory. The capsule with a larger capillary number shows a faster convergence, but finally the maximum Taylor parameter becomes independent of the capillary number at high oscillation frequencies. The capsule deformation at high frequencies is inversely proportional to the oscillation frequency, and is only determined by viscosity ratio.

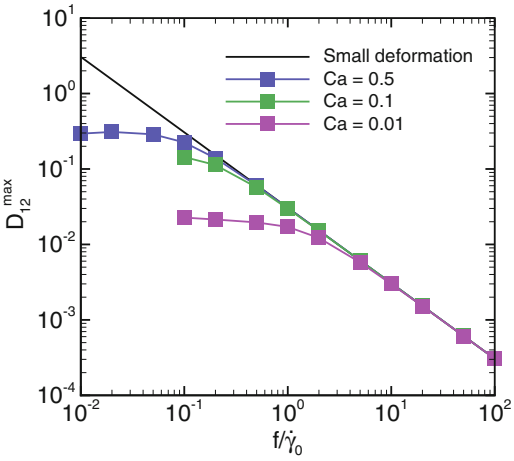


Fig. 19 Maximum Taylor parameter of a neo-Hookean capsule in oscillating shear flow as a function of the non-dimensional oscillation frequency $f/\dot{\gamma}_0$ for $\lambda = 5$. The black line shows values predicted by the small deformation theory, Eq. (83)

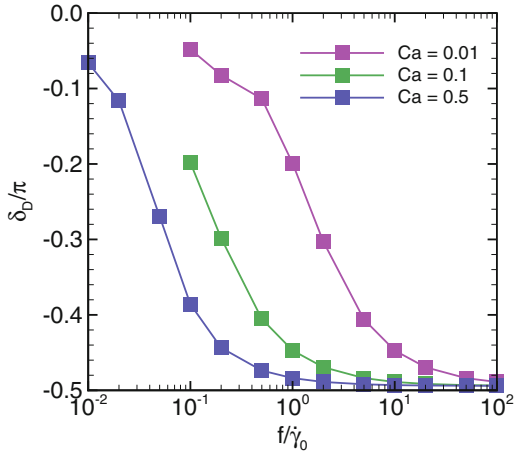


Fig. 20 Phase of a neo-Hookean capsule in oscillating shear flow as a function of the non-dimensional oscillation frequency for $\lambda = 5$

The phase of the capsule deformation is presented in Fig. 20. At low frequencies, the capsule deformation is determined by the equilibrium between the fluid and membrane forces, and the deformation phase follows the phase of the shear rate ($\delta_D = 0$). However, at high frequencies, the capsule deformation is limited by the strain, and the phase of the deformation shifts toward the phase of the strain ($\delta_D = \pi/2$).

4.2 Overshoot Phenomenon

We now focus more on the capsule deformation at low frequencies. We compare the Taylor parameter of the capsule in oscillating shear flow with that in steady

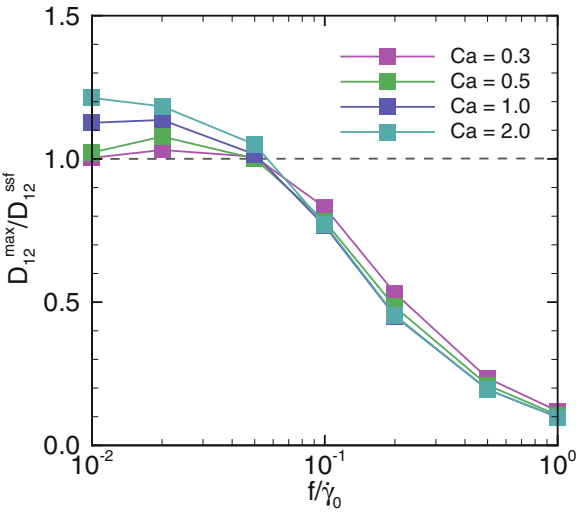


Fig. 21 Maximum Taylor parameter in oscillating shear flow D_{12}^{max} normalized by the Taylor parameter in steady shear flow D_{12}^{ssf} for $\lambda = 5$. The capsule in oscillating shear flow undergoes larger deformation than that in steady shear flow at low frequencies. The dashed line shows $D_{12}^{max}/D_{12}^{ssf} = 1$

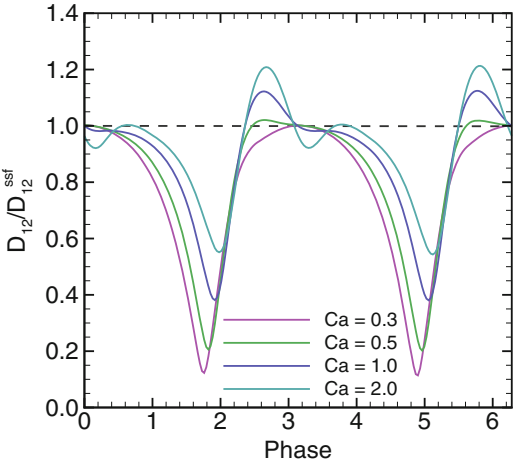


Fig. 22 Time history of the Taylor parameter of a neo-Hookean capsule in oscillating shear flow, where $\lambda = 5$ and $f/\dot{\gamma}_0 = 10^{-2}$. The Taylor parameter D_{12} is normalized by that in steady shear flow D_{12}^{ssf}. The dashed line shows $D_{12}/D_{12}^{ssf} = 1$

shear flow, and find "overshoot phenomenon." Figure 21 shows the maximum Taylor parameter in oscillating shear flow normalized by the Taylor parameter in steady shear flow, $D_{12}^{max}/D_{12}^{ssf}$ for $\lambda = 5$ and various capillary numbers. This result indicates that the capsule undergoes larger deformation in oscillating shear flow than that in steady shear flow at low frequencies.

Figure 22 shows the time history of the Taylor parameter at $f/\dot{\gamma} = 10^{-2}$ for $\lambda = 5$. The Taylor parameter has a single peak in a half cycle and the maximum value is nearly the same as D_{12}^{ssf} at a low capillary number $Ca = 0.3$. When the capillary number is increased, however, the maximum value becomes slightly larger than that in steady shear flow. The Taylor parameter peaks before the shear rate

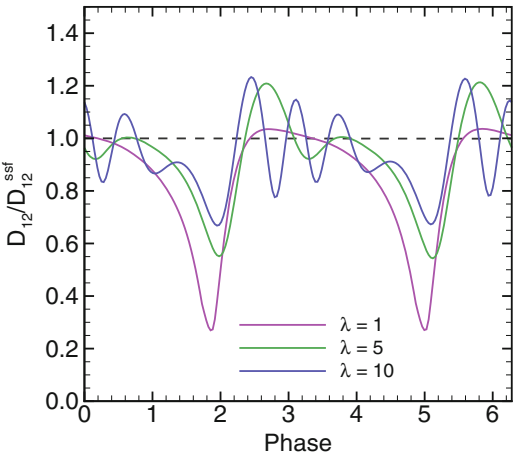


Fig. 23 Time history of the Taylor parameter of a neo-Hookean capsule in oscillating shear flow, where $Ca = 2.0$ and $f/\dot{\gamma}_0 = 10^{-2}$. The Taylor parameter D_{12} is normalized by that in steady shear flow D_{12}^{ssf}

reaches the maximum value, and at $Ca = 2.0$, the second peak also appears. When the viscosity ratio is increased, the number of the deformation peaks is increased as shown in Fig. 23.

4.3 Mechanism of Overshoot Phenomenon

What is the mechanism underlying the overshoot phenomenon? The overshoot phenomenon also occurs during transient deformation in start-up shear flow [4]. Here, we consider start-up shear flow as the simplest case to discuss the mechanism of the overshoot phenomenon.

The start-up shear flow consists of an extensional flow $E_{12}^{\infty} = E_{21}^{\infty} = \dot{\gamma}/2$ and a rotational flow $\Omega_{12}^{\infty} = -\Omega_{21}^{\infty} = \dot{\gamma}/2$. When the start-up shear flow is applied to a capsule, the capsule is elongated into the direction $\theta = \pi/4$ by the extensional flow. Meanwhile, the orientation of the capsule is changed from $\theta = \pi/4$ to a lower angle by the rotational flow. Finally, the capsule reaches equilibrium deformation and orientation, and exhibits a steady tank-tread motion. Effective extension rate at an orientation angle θ may be given by

$$\dot{\epsilon}\,(\theta) = \frac{\dot{\gamma}}{2}\sin 2\theta. \tag{84}$$

and thus the effective extension rate is the largest at $\theta = \pi/4$ and becomes smaller as decreasing θ.

We plot the time history of the Taylor parameter as a function of the effective extensional rate in Fig. 24 for $Ca = 1.0$ and three values of λ. Because the orientation angle decreases from $\theta = \pi/4$ during a transient process, the capsule first undergoes the largest deformation, and then D_{12} decreases until the capsule

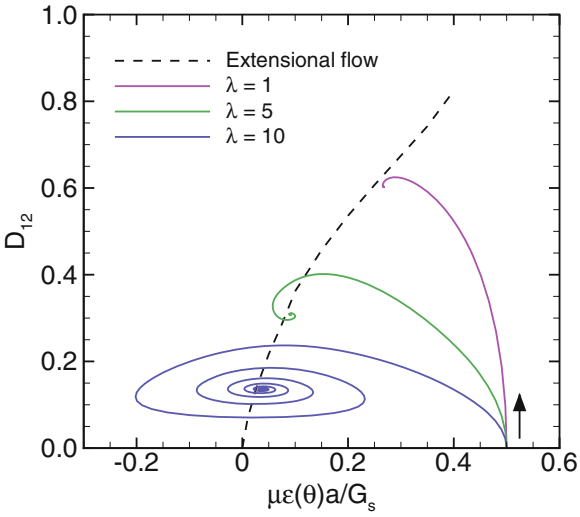


Fig. 24 Time history of the Taylor parameter in start-up shear flow as a function of the effective extension rate $\mu\epsilon\,(\theta)\,a/G_s$ for $Ca = 1.0$. The dashed line represents the Taylor parameter in steady extensional flow. The arrow shows the direction of the time

reaching an equilibrium state. At the equilibrium state, the effective extensional flow is counterbalanced with the flow generated by the capsule membrane. Thus, the equilibrium deformation in the start-up shear flow is nearly the same as that in steady extensional flow. For a large value of λ, this process is not a monotonic process. The effective extensional flow may be canceled by the flow due to the membrane even at a different orientation angle from the final one, but the resultant flow still changes the capsule to a different shape. The capsule therefore exhibits swinging-like motion, repeating overshoot and undershoot until when the capsule reaches the final equilibrium state.

Does a capsule undergo a larger overshoot when it has a smaller orientation angle at its equilibrium state? Note that the equilibrium deformation and orientation in start-up shear flow is identical to those in steady shear flow. Figure 25 is a contour plot of the orientation angle of the capsule in simple shear flow for various values of the capillary number and viscosity ratio. The orientation angle becomes lower for higher capillary numbers and larger viscosity ratios. We also show the maximum Taylor parameter of the capsule in start-up shear flow as a contour plot for Ca and λ in Fig. 26. We find that a large overshoot well corresponds to a low orientation angle. More than 10% overshoot, for example, are expected when the orientation angle is $\theta^{ssf} < 0.05\pi$ for the neo-Hookean capsule.

When the peak shear rate $\dot{\gamma}_0$ is fixed, the largest strain is given to the capsule by start-up shear flow among oscillating shear flows. The overshoot deformation presented in Fig. 26 thus also quantifies the upper limit of the capsule deformation in oscillating shear flows.

We have shown that there are two frequency ranges in terms of the capsule deformation. At low frequencies, the capsule deformation weakly depends on the frequency and the overshoot phenomenon likely occurs. At high frequencies, the capsule deformation decreases on the order of f^{-1} and follows the small

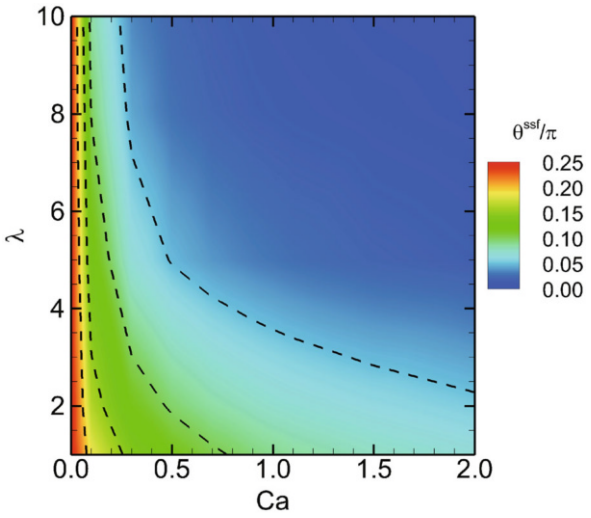


Fig. 25 Contour plot of the orientation angle of a neo-Hookean capsule in simple shear flow θ^{ssf} as functions of the capillary number Ca and viscosity ratio λ

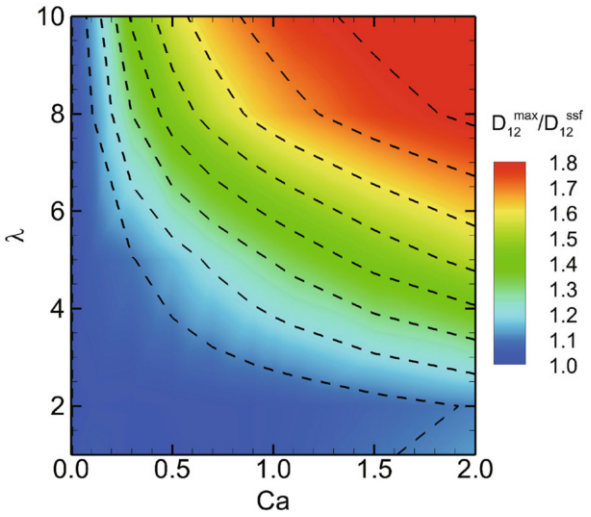


Fig. 26 Contour plot of the maximum Taylor parameter of a neo-Hookean capsule in start-up shear flow D_{12}^{max} as functions of the capillary number Ca and viscosity ratio λ. The value is normalized by the Taylor parameter in simple shear flow D_{12}^{ssf}

deformation theory. Finally, we roughly estimate the threshold frequency for the overshoot phenomenon. Substituting D_{12}^{ssf} into the left-hand side of Eq. (83), we have the threshold frequency as

$$\left(\frac{f}{\dot{\gamma}_0}\right)_{th} = \frac{5}{4\pi D_{12}^{ssf}\,(2\lambda+3)}. \tag{85}$$

This equation does not include membrane properties, and may be applicable to other deformable particles, such as bubbles and drops.

5 Sedimentation of Red Blood Cells

We have also applied the GPU-accelerated BEM to a numerical analysis of the sedimentation of red blood cells [16]. Because the density of red blood cells is slightly larger than that of plasma, red blood cells settle under gravity. The settling velocity called erythrocyte sedimentation rate has been used for medical practice [36].

A previous study reported that the number of red blood cells vertically oriented was 2.5 times greater than those horizontally oriented [37]. Hoffman and Inoue [38] also reported that under high centrifugal forces, red blood cells reoriented vertically and deformed into a bag-like shape. To understand this phenomenon more detail, we simulated the sedimentation of a red blood cell at various conditions.

A density difference between the internal and external fluids $\Delta\rho$ is given to a red blood cell suspended in an infinite unbounded domain. The gravitational force is added into the load on the membrane as

$$\mathbf{q}_g(\mathbf{y}) = \Delta\rho\{\mathbf{y}\cdot\mathbf{g}\}\mathbf{n}(\mathbf{y}), \tag{86}$$

where $\mathbf{y}$ is the position vector, $\mathbf{g}$ is the gravity, and $\mathbf{n}$ is the normal vector pointing outwards. The Skalak constitutive law, Eq. (58) is used for the red blood cell membrane. The initial shape of the red blood cell is obtained by imposing bending stiffness to a stretch-free oblate capsule, where the spontaneous curvature $c_0 = 0$. The aspect ratio of the oblate capsule is set to be 3.9 so that its reduced volume is similar to a real value ∼0.637 [39]. The characteristic length of this problem is defined by the radius of a sphere that has the same volume as the red blood cell. The Bond number is then given by

$$Bo = \frac{L^2\Delta\rho g}{G_s}, \tag{87}$$

where L is the characteristic length. Here, we also consider bending stiffness in the non-dimensional form,

$$Be = \frac{E_b}{G_s L^2}. \tag{88}$$

While the Bond number varies from $Bo = 2.5\times10^{-3}$ to $Bo = 2.5\times10^{-1}$, the bending stiffness is fixed to $Be = 1.0\times10^{-2}$ in this chapter.

5.1 Sedimentation Under Standard Gravity

First, we consider a single red blood cell settling in plasma under standard gravity. The corresponding Bond number is $Bo = 2.5\times10^{-3}$. We introduce local Cartesian

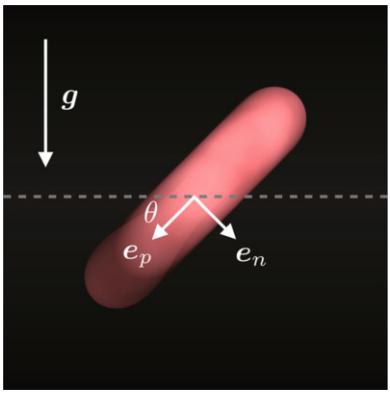


Fig. 27 Sedimentation of a single red blood cell under gravity **g**. The red blood cell settles and drifts at an orientation angle θ. Vectors $\mathbf{e}_p$ and $\mathbf{e}_n$ are local Cartesian bases

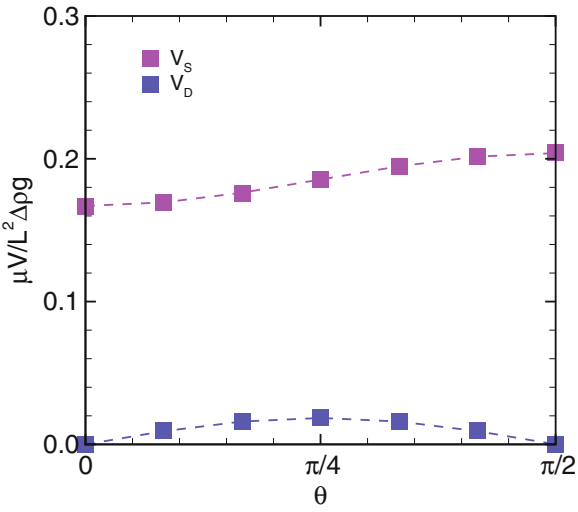


Fig. 28 Settling and drifting velocities of a red blood cell for $Bo = 2.5 \times 10^{-3}$ as a function of the orientation angle of the cell. V_S and V_D are the settling and drifting velocities, respectively. Dashed lines are obtained from Eqs. (89) and (90)

coordinates as shown in Fig. 27. The orientation angle of the red blood cell θ is defined by the angle between the horizontal plane and the vector $\mathbf{e}_p$, and $\theta = 0$ and $\theta = \pi/2$ represent horizontal and vertical orientations, respectively. The red blood cell is initially placed at an orientation angle $\theta = \theta_0$, and then undergoes sedimentation. While the red blood cell settles in the gravity direction, it also drifts in the horizontal direction. For this Bond number, the angular velocity of the red blood cell is small, and the cell orientation angle remains approximately $\theta = \theta_0$. The cell velocity also remains nearly constant after a transient state in the simulation.

The settling and drifting velocities change slightly with the orientation angle of the red blood cell as shown in Fig. 28, where $V_S(\theta)$ and $V_D(\theta)$ refer to the settling and drifting velocities at an orientation angle θ, respectively. The settling velocity is slightly increased when the orientation angle is increased. The drifting velocity is zero at orientation angles $\theta = 0$ and $\theta = \pi/2$, and peaks at $\theta_0 = \pi/4$.

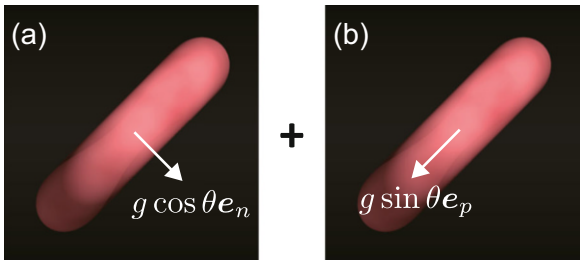


Fig. 29 A schematic of subproblems. (**a**) A force of $g\cos\theta\mathbf{e}_n$ is applied to the cell. (**b**) Another force of $g\sin\theta\mathbf{e}_p$ is applied to the cell

This is a Stokes flow problem. We decompose the problem into two subproblems: a force of $g\cos\theta\mathbf{e}_n$ is applied to the cell (Fig. 29a); another force of $g\sin\theta\mathbf{e}_p$ is applied to the cell (Fig. 29b). The cell velocities of these subproblems are $V^n\mathbf{e}_n = V_S(0)\cos\theta\ \mathbf{e}_n$, and $V^p\mathbf{e}_p = V_S(\pi/2)\sin\theta\ \mathbf{e}_p$. The settling and drifting velocities are then given by the superposition of these velocities:

$$\begin{aligned} V_S(\theta) &= V^n\cos\theta + V^p\sin\theta \\ &= V_S(0)\cos^2\theta + V_S(\pi/2)\sin^2\theta, \end{aligned} \tag{89}$$

and

$$\begin{aligned} V_D(\theta) &= -V^n\sin\theta + V^p\cos\theta \\ &= \{V_S(\pi/2) - V_S(0)\}\sin\theta\cos\theta. \end{aligned} \tag{90}$$

In the numerical results, $V_S(0) \approx 1.67\times10^{-1}L^2\Delta\rho g/\mu$ and $V_S(\pi/2) \approx 2.04\times10^{-1}L^2\Delta\rho g/\mu$. We substitute these values into the right-hand side of Eqs. (89) and (90), and compare these equations with the numerical results in Fig. 28. The equations well describe the settling and drifting velocities of the red blood cell at arbitrary orientation angles. When red blood cells settle in a suspension at random orientation angles, the mean velocity of the cells is estimated as 1.67 μm/s. This value is consistent with an experimental result, 1.69 μm/s [37].

5.2 *Effect of Cell Deformation*

We then consider effects of cell deformation. Figure 30 shows snapshots of the sedimentation of the red blood cell for $Bo = 2.5\times10^{-3}$ and $Bo = 2.5\times10^{-1}$, where $\lambda = 1$. The red blood cell deforms into a bag-like shape for $Bo = 2.5\times10^{-1}$ as observed in previous experiments [38]. This is likely due to gravitational hydrostatic pressure. A difference in the pressure between the upper and lower parts of the cell drives the cell to deform asymmetry in the gravitational direction. The orientation

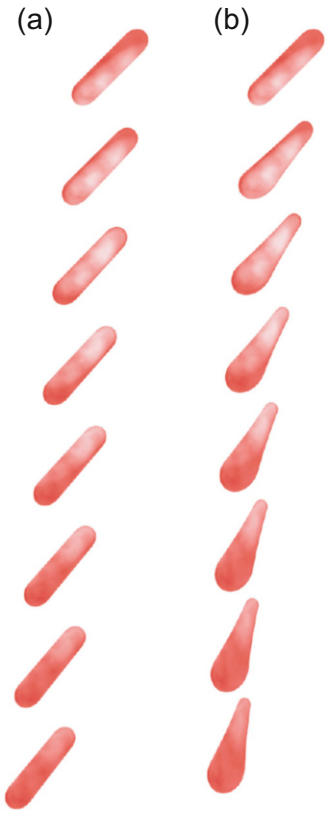


Fig. 30 Snapshots of the sedimentation of a red blood cell for (**a**) $Bo = 2.5 \times 10^{-3}$ and (**b**) $Bo = 2.5 \times 10^{-1}$. While the shape and orientation are nearly constant for $Bo = 2.5 \times 10^{-3}$, a bag-like shape and an orientation toward $\theta = \pi/2$ are observed for $Bo = 2.5 \times 10^{-1}$

angle of the red blood cell gradually increases for $Bo = 2.5 \times 10^{-1}$, and converges to $\theta = \pi/2$ after a long time period. Such a vertical orientation of red blood cells was also observed in previous experiments when a large gravitational acceleration was applied [38].

The deformation of the red blood cell has a small effect on the settling and drifting velocities of the cell. Figure 31 shows the time histories of the settling and drifting velocities for $Bo = 2.5 \times 10^{-1}$ as a function of the instantaneous orientation angle, where the sedimentation starts at various orientation angles. The non-dimensional settling and drifting velocities for $Bo = 2.5 \times 10^{-1}$ converge to those for $Bo = 2.5 \times 10^{-3}$. We also examined different viscosity ratio, and the non-dimensional velocities also converged to the same velocities.

We also quantify the angular velocity of the cell, $\omega = d\theta/dt$. Time histories of the angular velocities are shown in Fig. 32 as a function of the instantaneous orientation angle. Even when the red blood cell departures at different initial orientation angle, the angular velocities converge to the same velocity. We also find

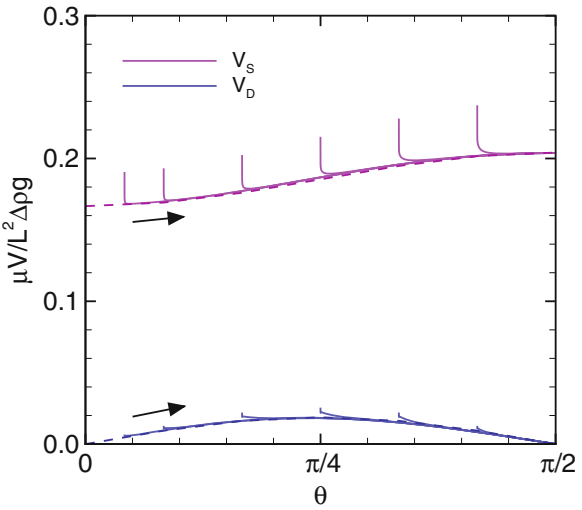


Fig. 31 Time histories of settling and drifting velocities for $Bo = 2.5 \times 10^{-1}$ as a function of the instantaneous orientation angle, where the sedimentation starts at various orientation angles. The dashed lines represent the velocities for $Bo = 2.5 \times 10^{-3}$. The arrows indicate the direction of the time

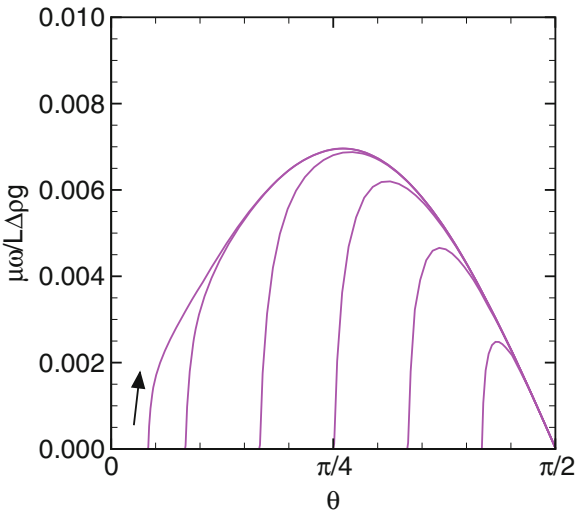


Fig. 32 Time histories of angular velocities for $Bo = 2.5 \times 10^{-1}$ as a function of the instantaneous orientation angle, where the sedimentation starts at various orientation angles. The arrow indicates the direction of the time

that the converged angular velocity becomes maximum around $\theta = \pi/4$ and may be written as

$$\omega \sim \omega_{\max} \sin 2\theta, \tag{91}$$

where $\omega_{\max}$ is the maximum angular velocity.

The maximum angular velocities are compared among different Bond numbers and viscosity ratios in Fig. 33. The maximum angular velocity increases linearly with the Bond number, in particular, for low Bond numbers. While the maximum angular velocity is lower for a larger viscosity ratio at high Bond numbers, it does not depend on the viscosity ratio at low Bond numbers.

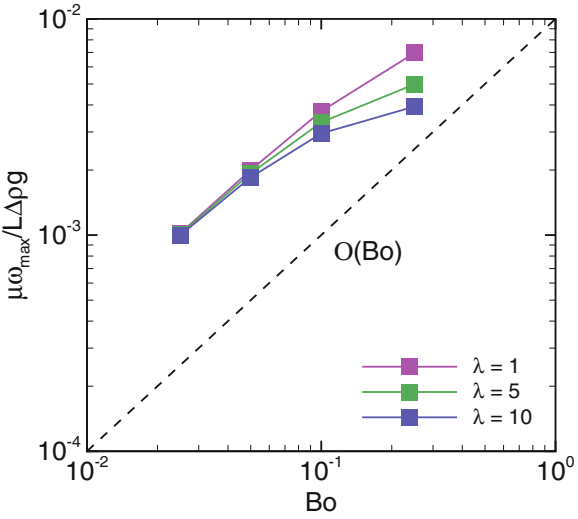


Fig. 33 Maximum angular velocity as a function of the Bond number. The dashed line represents a linear function of Bo

5.3 Asymmetric Deformation and Reorientation

We have shown that the red blood cell changes its orientation into the vertical direction during the sedimentation. The reorientation of the red blood cell is caused by the asymmetric deformation of the red blood cell. The extent of the shape asymmetry is quantified by the asymmetry parameter [40]:

$$\begin{aligned}\frac{\boldsymbol{\chi}}{L} &= \frac{\mathbf{X}_A - \mathbf{X}_V}{L} \\ &= \frac{1}{S}\int_S \frac{\mathbf{x}}{L} dS - \frac{1}{V}\int_V \frac{\mathbf{x}}{L} dV \\ &= \frac{1}{S}\int_S \frac{\mathbf{x} - \mathbf{X}_V}{L} dS, \end{aligned} \tag{92}$$

where $\mathbf{X}_A$ refers to the area centroid of the cell membrane, and $\mathbf{X}_V$ is the volume centroid of the cell. When the red blood cell settles at the vertical orientation $\theta = \pi/2$, the norm of the asymmetry parameter $\chi/L = |\boldsymbol{\chi}|/L$ is proportional to the Bond number as shown in Fig. 34.

The shape asymmetry of the red blood cell changes with the instantaneous orientation angle θ during the sedimentation. The shape asymmetry in the orientation direction $\mathbf{e}_p$ is calculated by $\chi_p = \boldsymbol{\chi} \cdot \mathbf{e}_p$. Figure 35 shows time histories of the asymmetry parameter as a function of the instantaneous orientation angle for $Bo = 2.5 \times 10^{-1}$. The shape asymmetry changes with a $\sin\theta$-like function and peaks at $\theta = \pi/2$.

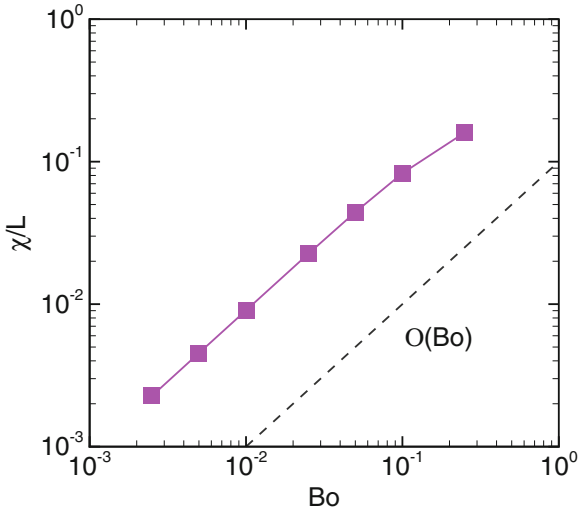


Fig. 34 Norm of the asymmetry parameter, $\chi/L = |\boldsymbol{\chi}|/L$, at the vertical orientation $\theta = \pi/2$ as a function of the Bond number

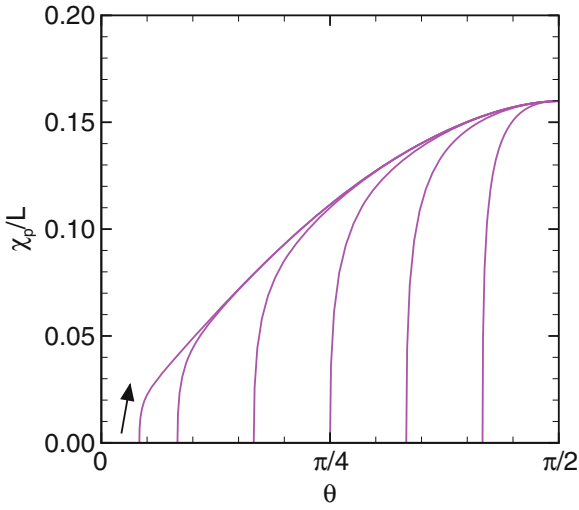


Fig. 35 Time histories of the asymmetry parameter χ_p/L for $Bo = 2.5 \times 10^{-1}$ as a function of the instantaneous orientation angle. The arrow indicates the direction of the time

We now discuss the relationship between the shape asymmetry and the angular velocity. We consider an asymmetric dumbbell consisting of two different-sized spheres connected by a rigid rod. The asymmetric dumbbell also exhibits reorientation because of a difference between the volume centroid and the hydrodynamic center. The angular velocity of the dumbbell is then given by Mogami et al. [41]

$$\omega(\theta) = \frac{\Delta\rho g d}{\mu} R\cos\theta, \tag{93}$$

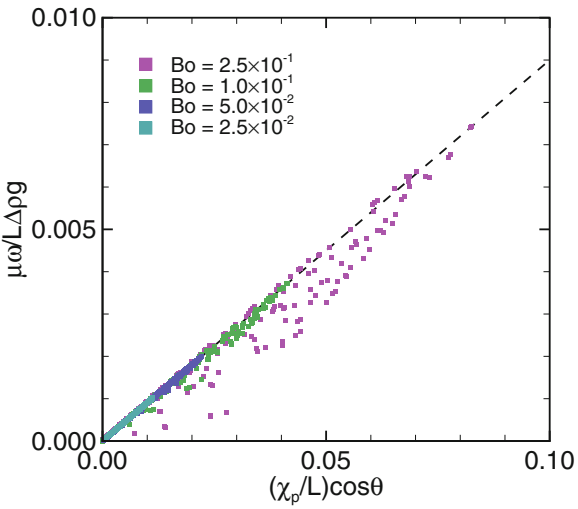


Fig. 36 Relationship between the asymmetry parameter and the angular velocity. The dashed line is Eq. (94) with $R_m = 0.09$

where d is the distance between the volume centroid and the hydrodynamic center, and R is the shape coefficient. Here, we use the asymmetric parameter χ_p instead of the distance d, and we have the non-dimensional angular velocity as

$$\frac{\mu\omega}{a\Delta\rho g} = \frac{\chi_p}{L} R_m \cos\theta, \tag{94}$$

where R_m is the modified shape coefficient. In Fig. 36, we plot the non-dimensional angular velocities at various orientation angles as a function of the parameter $\left(\chi_p/L\right)\cos\theta$. We find that the relationship between the shape asymmetry and the angular velocity is well described by Eq. (94). The modified shape coefficient is then estimated as $R_m \sim 0.09$. Recalling that the asymmetry parameter χ_p/L is proportional to $Bo\sin\theta$ (Figs. 34 and 35), we have also the following relationship:

$$\frac{\mu\omega}{L\Delta\rho g} \propto Bo\sin 2\theta. \tag{95}$$

This is consistent with Eq. (91).

6 Conclusions

While BEM is one of the most accurate methods for simulating capsules in Stokes flow, its heavy computational load has been a major issue, even when only a few capsules are simulated. We have shown that the GPU-accelerated BEM is a solution of this issue. While we have focused on the shear viscosity of capsule suspensions, the capsule deformation in oscillating shear flow, and the sedimentation of red

blood cells in this chapter, we have also applied the GPU-accelerated BEM to other problems, including the behavior of capsules near the wall [20, 40], the cytoadhesion of red blood cells infected by Plasmodium falciparum malaria [42], and the swimming of microorganisms [43, 44].

GPU computing is also effective for memory-bound processes. We have developed a GPU-accelerated lattice Boltzmann method, and have applied it to physiological flow problems, including microcirculation [45–49], pulmonary air flow [50, 51], and gastric flow [52, 53]. If you are interested in the GPU-accelerated LBM and its biomedical applications, please refer to these papers.

Acknowledgements This work was supported by JSPS KAKENHI Grant numbers 24680048, 25000008, 26107703, and 14J04025.

References

1. D. Barthès-Biesel, Annu. Rev. Fluid Mech. **48**, 25 (2016)
2. C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow* (Cambridge University Press, 1992)
3. C. Pozrikidis, J. Fluid Mech. **297**, 123 (1995)
4. S. Ramanujan, C. Pozrikidis, J. Fluid Mech. **361**, 117 (1998)
5. E. Lac, D. Barthès-Biesel, N.A. Pelekasis, J. Tsamopoulos, J. Fluid Mech. **516**, 303 (2004)
6. J. Walter, A.V. Salsac, D. Barthès-Biesel, P. Le Tallec, Int. J. Numer. Meth. Eng. **83**, 829 (2010)
7. É. Foessel, J. Walter, A.V. Salsac, Barthès-Biesel, J. Fluid Mech. **672**, 477 (2011)
8. J. Walter, A.V. Salsac, D. Barthès-Biesel, J. Fluid Mech. **676**, 318 (2011)
9. C. Dupont, A.V. Salsac, D. Barthès-Biesel, J. Fluid Mech. **721**, 180 (2013)
10. C. Dupont, F. Delahaye, D. Barthès-Biesel, A.V. Salsac, J. Fluid Mech. **791**, 738 (2016)
11. T. Omori, Y. Imai, T. Yamaguchi, T. Ishikawa, Phys. Rev. Lett. **108**, 138102 (2012)
12. T. Omori, D. Ishikawa, T. Barthès-Biesel, A.V. Salsac, Y. Imai, T. Yamaguchi, Phys. Rev. E **86**, 056321 (2012)
13. D. Matsunaga, Y. Imai, T. Omori, T. Ishikawa, T. Yamaguchi, J. Biomech. Sci. Eng. **9**, 00039 (2014)
14. D. Matsunaga, Y. Imai, T. Yamaguchi, T. Ishikawa, J. Fluid. Mech. **786**, 110 (2016)
15. D. Matsunaga, Y. Imai, T. Yamaguchi, T. Ishikawa, J. Fluid. Mech. **762**, 288 (2015)
16. D. Matsunaga, Y. Imai, C. Wagner, T. Ishikawa, J. Fluid. Mech. **806**, 102 (2016)
17. D. Barthès-Biesel, *Microhydrodynamics and Complex Fluids* (CRC Press, 2012)
18. J.R. Blake, A.T. Chwang, J. Eng. Math. **8**, 23 (1974)
19. S.E. Spagnolie, E. Lauga, J. Fluid Mech. **700**, 105147 (2012)
20. S. Nix, Y. Imai, T. Ishikawa, J. Biomech. **49**, 2249 (2016)
21. C.W.J. Beenakker, J. Chem. Phys. **85**, 1581 (1986)
22. A.Z. Zinchenko, R.H. Davis, J. Comput. Phys. **207**, 695 (2005)
23. R. Skalak, A. Tozeren, R. Zarda, S. Chien, Biophys. J. **13**, 245 (1973)
24. W. Helfrich, Z. Naturforsch. C **28**, 693 (1973)
25. O.Y. Zhong-can, W. Helfrich, Phys. Rev. A **39**, 5280 (1989)
26. T. Shimokawabe, T. Aoki, C. Muroi, J. Ishida, K. Kawano, T. Endo, A. Nukada, N. Maruyama, S. Matsuoka, in *Proc. 2010 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis* (2010), pp. 1–11
27. A. Einstein, Ann. Physik **19**, 289 (1906)
28. D. Barthès-Biesel, J.M. Rallison, J. Fluid Mech. **113**, 251 (1981)
29. J.R. Clausen, D.A. Reasor, C.K. Aidun, J. Fluid Mech. **685**, 202 (2011)

30. E. Guazzelli, F.J. Morris, *A Physical Introduction to Suspension Dynamics* (Cambridge University Press, 2012)
31. G.K. Batchelor, J. Fluid Mech. **41**, 545 (1970)
32. I.M. Krieger, T.J. Dougherty, T. Soc. Rheol. **3**, 137 (1959)
33. G.Y. Onoda, E.G. Liniger, Phys. Rev. Lett. **64**, 2727 (1990)
34. M. Zhao, P. Bagchi, Phys. Fluids **23**, 111901 (2011)
35. D. Barthès-Biesel, H. Sgaier, J. Fluid Mech. **160**, 119 (1985)
36. S. Baskrut, B. Neu, H.J. Meiselman, *Red Blood Cell Aggregation* (CRC Press, 2012)
37. A.C. Groom, J.C. Anderson, J. Cell. Physiol. **79**, 127 (1972)
38. J.F. Hoffman, S. Inoué, Proc. Natl. Acad. Sci. U. S. A. **103**, 2971 (2006)
39. E. Evans, Y.C. Fung, Microvasc. Res. **4**, 335 (1972)
40. S. Nix, Y. Imai, D. Matsunaga, T. Yamaguchi, T. Ishikawa, Phys. Rev. E **90**, 043009 (2014)
41. Y. Mogami, J. Ishii, A. Baba, Biol. Bull. **201**, 26 (2001)
42. S. Ishida, Y. Imai, Y. Ichikawa, S. Nix, D. Matsunaga, T. Omori, T. Ishikawa, Sci. Technol. Adv. Mater. **17**, 454 (2016)
43. K. Kyoya, D. Matsunaga, Y. Imai, T. Omori, T. Ishikawa, Phys. Rev. E **92**, 063027 (2015)
44. T. Ishikawa, T. Tanaka, Y. Imai, T. Omori, D. Matsunaga, Proc. Roy. Soc. Lond. A **472**, 20150604 (2016)
45. N. Takeishi, Y. Imai, K. Nakaaki, T. Yamaguchi, T. Ishikawa, Physiol. Rep. **2**, e12037 (2014)
46. N. Takeishi, Y. Imai, T. Yamaguchi, T. Ishikawa, Phys. Rev. E **92**, 063011 (2015)
47. N. Takeishi, Y. Imai, S. Ishida, T. Omori, R.D. Kamm, T. Ishikawa, Am. J. Physiol. Heart Circ. Physiol. **311**, H395 (2016)
48. N. Takeishi, Y. Imai, Sci. Rep. **7**, 5381 (2017)
49. S. Ishida, A. Ami, Y. Imai, Biophys. J. **113**, 1163 (2017)
50. T. Miki, X. Wang, T. Aoki, Y. Imai, T. Ishikawa, K. Takase, T. Yamaguchi, Comput. Meth. Biomech. Biomed. Eng. **15**, 771 (2012)
51. Y. Imai, T. Miki, T. Ishikawa, T. Aoki, T. Yamaguchi, J. Biomech. **45**, 1809 (2012)
52. R. Berry, T. Miyagawa, N. Paskaranandavadivel, P. Du, T.R. Angeli, M.L. Trew, J.A. Windsor, Y. Imai, G. O'Grady, L.K. Cheng, Am. J. Physiol. Gastrointest. Liver Physiol. **311**, G895 (2016)
53. T. Miyagawa, Y. Imai, S. Ishida, T. Ishikawa, Am. J. Physiol. Gastrointest. Liver Physiol. **311**, G1114 (2016)

Recent Advances in ALE-VMS and ST-VMS Computational Aerodynamic and FSI Analysis of Wind Turbines

Artem Korobenko, Yuri Bazilevs, Kenji Takizawa, and Tayfun E. Tezduyar

Abstract We describe the recent advances made by our teams in ALE-VMS and ST-VMS computational aerodynamic and fluid–structure interaction (FSI) analysis of wind turbines. The ALE-VMS method is the variational multiscale version of the Arbitrary Lagrangian–Eulerian method. The VMS components are from the residual-based VMS method. The ST-VMS method is the VMS version of the Deforming-Spatial-Domain/Stabilized Space–Time method. The ALE-VMS and ST-VMS serve as the core methods in the computations. They are complemented by special methods that include the ALE-VMS versions for stratified flows, sliding interfaces and weak enforcement of Dirichlet boundary conditions, ST Slip Interface (ST-SI) method, NURBS-based isogeometric analysis, ST/NURBS Mesh Update Method (STNMUM), Kirchhoff–Love shell modeling of wind-turbine structures, and full FSI coupling. The VMS feature of the ALE-VMS and ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow, and the moving-mesh feature of the ALE and ST frameworks enables high-resolution computation near the rotor surface. The ST framework, in a general context, provides higher-order accuracy. The ALE-VMS version for sliding interfaces and the ST-SI enable moving-mesh computation of the spinning rotor. The mesh covering the rotor spins with it, and the sliding interface or the SI between the spinning mesh and the rest of the mesh accurately connects the

A. Korobenko (✉)
Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, AB, Canada
e-mail: artem.korobenko@ucalgary.ca

Y. Bazilevs
School of Engineering, Brown University, Providence, RI, USA

K. Takizawa
Department of Modern Mechanical Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan

T. E. Tezduyar
Mechanical Engineering, Rice University, Houston, TX, USA

Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_7

two sides of the solution. The ST-SI also enables prescribing the fluid velocity at the turbine rotor surface as weakly-enforced Dirichlet boundary condition. The STNMUM enables exact representation of the mesh rotation. The analysis cases reported include both the horizontal-axis and vertical-axis wind turbines, stratified and unstratified flows, standalone wind turbines, wind turbines with tower or support columns, aerodynamic interaction between two wind turbines, and the FSI between the aerodynamics and structural dynamics of wind turbines. Comparisons with experimental data are also included where applicable. The reported cases demonstrate the effectiveness of the ALE-VMS and ST-VMS computational analysis in wind-turbine aerodynamics and FSI.

1 Introduction

1.1 Role of Wind Turbines in Renewable-Energy Generation

Electricity generating wind turbines now present one of the biggest renewable-energy-based power production facilities on the planet. Latest estimates from the International Renewable Energy Agency [1] show that onshore wind is already at grid parity, i.e., it is able to generate power at the so-called levelized cost of electricity (LCOE) that is less than or equal to the price of purchasing power from the electric grid. According to recent reports of the Energy Information Administration of the U.S. Department of Energy [2] and the annual energy report of the European Commission [3], LCOE for onshore wind falls within the range of \$0.04–\$0.10 per kWh, making it cost-competitive with conventional power sources such as coal, natural gas, and nuclear energy (\$0.06–\$0.12 per kWh). Moreover, wind energy is also one of the fastest growing power production sectors: During 2000–2012, the installed capacity for wind power increased by 266 GW, while that for solar power went up by 100 GW and nuclear power by only 9 GW [4]. In addition, as noted in [1], wind energy technology has the largest remaining cost reduction potential. As a result, countries around the world are putting substantial effort into the development of wind energy technologies. The ambitious wind energy goals put pressure on the wind energy industry research and development to significantly enhance the current wind generation capabilities in a short period of time and further decrease the associated costs. This calls for transformative concepts and designs (e.g., floating offshore wind turbines) that must be created and analyzed with high-precision methods and tools. These include complex-geometry, 3D, time-dependent, multi-physics predictive simulation methods and software that will play an increasingly important role as wind energy technology continues to develop.

1.2 Computational Analysis of Wind Turbines

To obtain high-fidelity predictive simulation results for wind turbines, 3D modeling is essential. However, simulation of wind turbines at full scale engenders a number of challenges: the flow is fully turbulent, requiring highly accurate methods and increased grid resolution. The presence of fluid boundary layers, where turbulence is created, complicates the situation further. Wind-turbine blades are long and slender structures, with complex distribution of material properties, for which the numerical approach must have good efficiency and approximation power, and avoid locking. Wind-turbine blades are laminated composite structures that require advanced multiscale modeling approaches to accurately predict the onset and progression of damage and to estimate their remaining useful fatigue life. Wind-turbine simulations involve moving and stationary components, and the fluid–structure coupling must be accurate, efficient, and robust to preclude divergence of the computations.

In recent years, several attempts were made to address some of the above challenges and to raise the fidelity and predictability levels of wind-turbine simulations. Standalone aerodynamics simulations of wind-turbine configurations in 3D were reported in [5–10], while standalone structural analyses of rotor blades of complex geometry and material composition, but under assumed wind-load conditions or wind-load conditions coming from separate aerodynamic computations, were reported in [11–16].

While the above contributions present important advances in the aerodynamics and structural modeling of wind turbines, the authors of the present paper feel that fluid–structure interaction (FSI) simulations at full scale are essential for accurate modeling of wind turbines. Indeed, in a recent work [17], it was shown that FSI modeling and simulation of wind turbines is important for accurately predicting their mechanical behavior at full scale. This is because the motion and deformation of the wind-turbine blades depend on the wind speed and air flow, and the air flow patterns depend on the motion and deformation of the blades. In order to simulate the coupled problem, the equations governing the air flow and the blade motions and deformations need to be solved simultaneously, with proper kinematic and dynamic conditions coupling the two physical systems. Without that the modeling cannot be realistic.

Additional modeling challenges stem from realistic scenarios of wind turbines arranged in arrays, and operating in complex turbulent atmospheric boundary layer (ABL) flows with a wide range of energy-containing scales and in different atmospheric stability regimes. Wind turbines positioned downstream operate in the wakes generated by upstream turbines, and have been observed to generate less power compared to the upstream turbines. In addition, downstream turbines experience higher variations in aerodynamic loads, which tend to shorten their

fatigue life leading to premature blade failure [18]. Depending on the atmospheric stability regime, spacing between turbines, the underlying surface topology, turbulence intensity, and wind direction and speed, the power-generation deficit for the downstream turbines may be as high as 40%, as was reported in [19, 20].

We feel that, to address the FSI modeling challenges described, it is important to focus on the development and application of numerical techniques that are general, accurate, robust, and efficient for the targeted class of problems. Such techniques are summarized in what follows and are described in greater detail in the body of this article.

1.3 Isogeometric Analysis

Isogeometric Analysis (IGA), first introduced in [21] and further expanded on in [22–33], is adopted as the geometry modeling and simulation framework for wind turbines in some of the examples presented in this paper. We use IGA based on NURBS (non-uniform rational B-splines), which are more efficient than standard finite elements for representing complex, smooth geometries, such as wind-turbine blades. IGA was successfully employed for computation of turbulent flows [34–39], nonlinear structures [15, 40–44], and FSI [45–48], and, in most cases, gave a clear advantage over standard low-order finite elements in terms of solution per-degree-of-freedom accuracy. This is in part attributable to the higher-order smoothness of the basis functions employed. In addition, flows about rotating components are naturally handled in an IGA framework because all conic sections, and in particular, circular and cylindrical shapes, are represented exactly [49].

The wind-turbine structures are governed by the isogeometric rotation-free shell formulation with the aid of the bending-strip method [15]. The method is appropriate for thin-shell structures comprised of multiple C^1- or higher-order continuous surface patches that are joined or merged with continuity no greater than C^0. The Kirchhoff–Love shell theory, which relies on higher-order continuity of the basis functions, is employed in the patch interior as in [44]. Although NURBS-based IGA is used in this work, other discretizations that allow for local mesh refinement, such as T-splines [28, 29] or subdivision surfaces [50–52], may also be employed.

In addition, an isogeometric representation of the analysis-suitable geometry can be used in generating meshes for finite element computation with the ALE-VMS and ST-VMS methods.

1.4 ALE-VMS

The ALE-VMS method [8, 47] is the variational multiscale version of the Arbitrary Lagrangian–Eulerian method [53]. The VMS components are from the residual-based VMS (RBVMS) method given in [34, 39, 54, 55]. The ALE-VMS is a moving-mesh method, suitable for computation of flows with moving boundaries and interfaces (MBI), including FSI. Moving the fluid mechanics mesh to track a fluid–solid interface enables mesh-resolution control near the interface and, consequently, high-resolution representation of the boundary layer. The ALE-VMS originated from the RBVMS formulation of incompressible turbulent flows proposed in [34] for nonmoving meshes, and may be thought of as an extension of the RBVMS method to moving meshes. As such, it was presented for the first time in [47] in the context of FSI. Although ALE-VMS gave reasonably good results for several important turbulent flows, it was evident in [34, 37] that to obtain accurate results for wall-bounded turbulent flows the method required relatively fine resolution of the boundary layers. This fact makes ALE-VMS a somewhat costly technology for full-scale wall-bounded turbulent flows at high Reynolds numbers, which are characteristic of the present application. For this reason, weakly enforced essential boundary condition formulation was introduced in [56], which significantly improved the performance of the ALE-VMS formulation in the presence of unresolved boundary layers [35, 36, 39]. The weak boundary condition formulation may be thought of as an extension of Nitsche's method [57] to the Navier–Stokes equations of incompressible flows. Another interpretation of the weak boundary condition formulation is that it is a discontinuous Galerkin method (see, e.g., [58]), where the continuity of the basis functions is enforced everywhere in the domain interior, but not at the domain boundary. Recently, the ALE-VMS and weakly enforced essential boundary condition formulations were extended to the modeling of stratified turbulent flows in [59–61].

The ALE-VMS and RBVMS have been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include wind-turbine aerodynamics and FSI [8, 17, 62–67], more specifically, vertical-axis wind turbines (VAWTs) [68, 69], floating wind turbines [70], wind turbines in atmospheric boundary layers [71], and fatigue damage in wind-turbine blades [72], patient-specific cardiovascular fluid mechanics and FSI [46, 47, 73–77], biomedical-device FSI [78–83], ship hydrodynamics with free-surface flow and fluid–object interaction [84, 85], hydrodynamics and FSI of a hydraulic arresting gear [86, 87], hydrodynamics of tidal-stream turbines with free-surface flow [88], and bioinspired FSI for marine propulsion [89, 90].

1.5 ST-VMS and ST-SUPS

The Space–Time Variational Multiscale (ST-VMS) method [91–93] is the VMS version of the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [94–96]. The DSD/SST was introduced for computation of flows with MBI, including FSI. In MBI computations the DSD/SST functions as a moving-mesh method. Some of the earliest FSI computations with the DSD/SST formulation were reported in [97] for vortex-induced vibrations of a cylinder and in [98] for flow-induced vibrations of a flexible, cantilevered pipe (1D structure with 3D flow). The DSD/SST formulation has been used extensively in 3D computations of parachute FSI, starting with the 3D computations reported in [99], and evolving within a decade to computations with more sophisticated core and special methods [96, 100–102]. Because the stabilization components of the DSD/SST are the Streamline-Upwind/Petrov-Galerkin (SUPG) [103] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [94] stabilizations, the method is also called "ST-SUPS." The VMS components of the ST-VMS are from the RBVMS. The ST-VMS has two more stabilization terms beyond those the ST-SUPS has, and these additional terms give the method better turbulence modeling features. Conversely, we can see the ST-SUPS as a reduced version of the ST-VMS. The ST-SUPS and ST-VMS, because of the higher-order accuracy of the ST framework (see [91, 92]), are desirable also in computations that do not involve MBI. The thermo-fluid ST-VMS for the coupled incompressible-flow and thermal-transport equations was introduced in [93]. It was derived by expanding the incompressible-flow ST-VMS [91, 92] to include the thermal-transport equation and the coupling between the flow and thermal-transport equations.

The ST-SUPS and ST-VMS have also been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include spacecraft parachute analysis for the main parachutes [104–108], cover-separation parachutes [109] and the drogue parachutes [110–112], wind-turbine aerodynamics for horizontal-axis wind-turbine (HAWT) rotors [8, 9, 105, 113], full HAWTs [66, 114–116] and VAWTs [117], flapping-wing aerodynamics for an actual locust [105, 118–120], bioinspired MAVs [115, 116, 121, 122] and wing-clapping [123, 124], blood flow analysis of cerebral aneurysms [115, 125], stent-blocked aneurysms [125–127], aortas [128, 129] and heart valves [116, 123, 129–132], spacecraft aerodynamics [109, 133], thermo-fluid analysis of ground vehicles and their tires [93], thermo-fluid analysis of disk brakes [134], flow-driven string dynamics in turbomachinery [135], flow analysis of turbocharger turbines [136–138], flow around tires with road contact and deformation [139, 140], ram-air parachutes [141], and compressible-flow parachute aerodynamics [142].

In this article, in the ST computational flow analysis of a VAWT, the ST framework provides higher-order accuracy in a general context. The VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow. The moving-mesh feature of the ST framework enables high-resolution computation near the turbine rotor surface.

1.6 ST-SI

The ST Slip Interface (ST-SI) method [117, 134] was introduced in [117], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS when we have spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers. The SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the flow field. The starting point in the development of the ST-SI was the ALE-VMS version for sliding interfaces [49, 143]. In the ST-SI, interface terms similar to those in the ALE-VMS version are added to the ST-VMS formulation to account for the compatibility conditions for the velocity and stress. An ST-SI version where the SI is between fluid and solid domains with weakly enforced Dirichlet boundary conditions for the fluid was also presented in [117]. The SI in this case is a "fluid–solid SI" rather than a standard "fluid–fluid SI." The thermo-fluid ST-SI, introduced in [134] for the coupled incompressible-flow and thermal-transport equations, retain the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been successfully applied to aerodynamic analysis of VAWTs [117], thermo-fluid analysis of disk brakes [134], flow-driven string dynamics in turbomachinery [135], flow analysis of turbocharger turbines [136–138], flow around tires with road contact and deformation [139, 140], aerodynamic analysis of ram-air parachutes [141], and heart valve flow analysis [129, 131, 132].

In another version of the ST-SI presented in [117], the SI is between a thin porous structure and the fluid on its two sides. This enables dealing with the fabric porosity in a fashion consistent with how the standard fluid–fluid SIs are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid SIs. Furthermore, this version enables handling thin structures that have T-junctions. This method has been successfully used in incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [141]. The compressible-flow ST-SI methods were introduced in [142], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [142]. These, together with the compressible-flow ST SUPG method [144], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled successful ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [142]. The computations were in the context of finite element discretization.

In this article, in the ST computational flow analysis of the VAWT, the ST-SI enables moving-mesh computation of the spinning turbine rotor. The mesh covering the rotor spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. It also enables, in one of the computations, prescribing the fluid velocity at the rotor surface as weakly enforced Dirichlet boundary condition.

1.7 ST-IGA and STNMUM

The ST-IGA [91, 118, 136] was introduced in [91]. It is the integration of the ST framework with isogeometric discretization. First computations with the ST-VMS and ST-IGA were reported in [91] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. The stability and accuracy analysis given [91] for the advection equation showed that using higher-order basis functions in time would be essential in getting full benefit out of using higher-order basis functions in space.

In the early stages of the ST-IGA, the emphasis was on IGA basis functions in time. As pointed out in [91, 92] and demonstrated in [118, 119, 121], higher-order NURBS basis functions in time provide a more accurate representation of the motion of the solid surfaces and a mesh motion consistent with that. They also provide more efficiency in temporal representation of the motion and deformation of the volume meshes, and better efficiency in remeshing. That is how the ST/NURBS Mesh Update Method (STNMUM) [114, 118, 119, 121] was introduced and demonstrated in [118, 119, 121]. The name "STNMUM" was given in [114].

The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS basis functions in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly, and a "secondary mapping" [91, 92, 105, 118] enables also specifying a constant angular velocity for invariant speeds along the paths.

The ST framework and NURBS in time also enable, with the "ST-C" method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [93, 134, 135, 145].

The STNMUM and desirable features of the ST-IGA with IGA basis functions in time have been demonstrated in many 3D computations. The classes of problems solved are flapping-wing aerodynamics for an actual locust [105, 118–120], bioinspired MAVs [115, 116, 121, 122] and wing-clapping [123, 124], separation aerodynamics of spacecraft [109], aerodynamics of HAWTs [66, 114–116] and VAWTs [117], thermo-fluid analysis of ground vehicles and their tires [93], thermo-fluid analysis of disk brakes [134], flow-driven string dynamics in turbomachinery [135], and flow analysis of turbocharger turbines [136–138].

The ST-IGA with IGA basis functions in space provides more accurate representation of the geometry and increased accuracy in the flow solution. Because it accomplishes that with less number of control points, and consequently with larger effective element sizes, it enables using larger time-step sizes while keeping the Courant number at a desirable level for good accuracy. It has been utilized in ST computational flow analysis of turbocharger turbines [136–138], ram-air parachutes [141], tires with road contact and deformation [140], and heart valves [129, 131, 132].

In this article, in the ST computational flow analysis of the VAWT, the STNMUM enables exact representation of the mesh rotation.

1.8 Computations Reported

The analysis cases reported include both the HAWTs and VAWTs, stratified and unstratified flows, standalone wind turbines, wind turbines with tower or support columns, aerodynamic interaction between two wind turbines, and the FSI between the aerodynamics and structural dynamics of wind turbines.

1.9 Outline of the Remaining Sections

In Sects. 2 and 3, we describe the methods for the aerodynamics parts of this work, namely, the ALE-VMS, ST-VMS, ST-SUPS, weak enforcement of Dirichlet boundary conditions, sliding-interface formulation, and the ST-SI with fluid–fluid SI and fluid–solid SI. For ALE-VMS, the methods are presented in the context of stratified incompressible flows used to model ABLs. Additional VMS modeling terms and stabilization parameters, which arise due to coupling of the Navier–Stokes momentum and temperature equations, are also presented. We also describe, for the ST-VMS computations, a method with NURBS basis functions in time for exact temporal representation of the rotor motion with constant angular velocity. In Sect. 4, we describe our approach to structural modeling of wind turbines, which is based on the Kirchhoff–Love thin-shell theory and the bending-strip method (see [15, 17, 44]). We add the discussion of rotation-free bending-stabilized cables, which are employed in conjunction with Kirchhoff–Love shells to produce full-turbine structural models for use in FSI simulations. In Sect. 5, we describe the core FSI coupling and mesh moving techniques and a recently proposed mesh moving technique for sliding-interface computations that can accommodate complex turbine motions [67]. Although developed and applied in the context of wind-turbine FSI, the proposed mesh moving method is applicable in any situation that involves rotating objects whose center of rotation also moves in space (e.g., turning maneuver of a submarine with a spinning propeller). In Sect. 6, we provide a description of the wind-turbine rotor geometry modeling and summarize the aerodynamics computations for the 5 MW HAWT rotor in a stably stratified ABL flow. The results presented are from reference [59] and include results from ALE-VMS finite element computations. In Sect. 7, we present simulations of a three-blade, high-solidity VAWT with rated power of 3.5 kW from [68]. One- and two-turbine simulations are shown to illustrate the methods. No stratification is employed in the VAWT simulations. In Sect. 8, we describe, from [117], finite-element-based ST-VMS computational flow analysis of a VAWT with four support columns at the periphery. We conduct test computations with 2D and 3D models. In Sect. 9 we present single-wind-turbine FSI simulations from [67, 69] to highlight the use of the mesh moving technique that accommodates sliding interfaces in relative motion. In Sect. 10, we present simulations of two back-to-back 5 MW wind turbines operating in an ABL in an effort to assess the effect of the upstream turbine on the wind loading and power production characteristics of the downstream turbine. The simulations are

from [61], and present an amalgamation of the computational techniques described in this article. In addition, a multi-domain method, originally proposed in [146], is adopted to carry out the simulations. In Sect. 11, we provide our concluding remarks.

2 ALE-VMS Formulation of the Navier–Stokes Equations of Incompressible Stratified Flows

2.1 Continuous Problem

Let $\Omega_t \in \mathbb{R}^{n_{\text{sd}}}$, $d = 2, 3$, be the spatial domain of the aerodynamics problem with boundary Γ_t at time $t \in (0, T)$. The subscript t indicates that the fluid mechanics spatial domain is time-dependent. The Navier–Stokes equations of incompressible stratified flows in the ALE frame may be written on Ω_t and $\forall t \in (0, T)$ as

$$\rho \left(\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u} - \hat{\mathbf{u}} \right) \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma} - \mathbf{b} = \mathbf{0}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\left. \frac{\partial \phi}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u} - \hat{\mathbf{u}} \right) \cdot \nabla \phi - \nabla \cdot \nu_\phi \nabla \phi - f = 0, \tag{3}$$

where ρ, $\mathbf{u}$, and $\mathbf{f}$ are the density, velocity, and the external force, respectively, and the stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}). \tag{4}$$

Here p is the pressure, $\mathbf{I}$ is the identity tensor, μ is the dynamic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor given by

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right). \tag{5}$$

In Eq. (3), ϕ is the fluid potential temperature and ν_ϕ is the diffusivity. The temperature field ϕ has the decomposition

$$\phi(\mathbf{x}, t) = \bar{\phi}(x_3) + \phi'(\mathbf{x}, t), \tag{6}$$

where $\bar{\phi}(x_3)$ is a background field varying only in the x_3-direction (i.e., vertical direction), and $\phi'(\mathbf{x}, t)$ is a fluctuating field with full space and time dependence. The Boussinesq forcing term denoted by $\mathbf{b}$ in Eq. (1) takes on the form

$$\mathbf{b} = \rho g \frac{\phi'}{\phi_0} \mathbf{e}_3, \tag{7}$$

where ϕ_0 is the reference temperature assumed constant in the Boussinesq approximation, g is the gravitational-acceleration magnitude, and $\mathbf{e}_3$ is the Cartesian basis vector pointing in the vertical direction.

The structure of the coupled system given by Eqs. (1)–(3) is as follows: The temperature field ϕ enters the Navier–Stokes momentum equation through the Boussinesq force, while the fluid velocity $\mathbf{u}$ appears in the convective term in Eq. (3).

Remark 1 In Eqs. (1) and (3), the notation $\left|_{\hat{x}}\right.$ implies that the time derivative is taken with respect to a fixed referential-domain spatial coordinates $\hat{\mathbf{x}}$, and $\hat{\mathbf{u}}$ is the velocity of the fluid domain Ω_t. The spatial gradients are taken with respect to the spatial coordinates $\mathbf{x}$ of the current configuration.

Remark 2 In ABL simulations, the Earth rotation effects may be important. For this, the Coriolis force is added to the Navier–Stokes momentum balance equation given by Eq. (1) as

$$\mathbf{f} = f_c \epsilon_{ij3} u_j \mathbf{e}_i, \tag{8}$$

where f_c is the Coriolis parameter and ϵ_{ijk}'s are the Cartesian components of the alternator tensor.

Remark 3 Equation (3) can also be used for problems with density stratification. In this case, ϕ is associated with the fluid density, and the form of the Boussinesq forcing term takes into account the density fluctuation and ratio of the inertial to gravity forces. See [59–61] for details.

In the interest of generality, the theory sections of this article cover the full stratified-flow case. The regular, unstratified-flow formulation may be recovered by neglecting Eq. (3), and removing the Boussinesq forcing term from Eq. (1). The computations presented in this article make use of both stratified- and unstratified-flow models.

2.2 ALE-VMS

The ALE-VMS formulation of stratified incompressible flows is given as follows: find $\mathbf{u}^h \in \mathcal{S}_u^h$, $p^h \in \mathcal{S}_p^h$, and $\phi^h \in \mathcal{S}_\phi^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$, $\forall q^h \in \mathcal{V}_p^h$, and $\eta^h \in \mathcal{V}_\phi^h$:

$$\int_{\Omega_t} \mathbf{w}^h \cdot \rho \left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega$$
$$- \int_{\Omega_t} \mathbf{w}^h \cdot \mathbf{b}^h \, \mathrm{d}\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathrm{d}\Omega$$

$$
\begin{aligned}
&- \int_{(\Gamma_t)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \, \mathrm{d}\Gamma + \int_{\Omega_t} q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}\Omega \\
&+ \int_{\Omega_t} \eta^h \left(\left. \frac{\partial \phi^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \nabla \phi^h - f^h \right) \mathrm{d}\Omega \\
&- \int_{\Omega_t} \nabla \eta^h \cdot \nu_\phi \nabla \phi^h \, \mathrm{d}\Omega - \int_{(\Gamma_t)_{\mathrm{h}}} \eta^h h^h \, \mathrm{d}\Gamma \\
&+ \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{SUPS}} \left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \nabla \mathbf{w}^h + \frac{\nabla q^h}{\rho} \right) \cdot \mathbf{r}_{\mathrm{M}} \left(\mathbf{u}^h, p^h \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega_t^e} \rho \nu_{\mathrm{LSIC}} \nabla \cdot \mathbf{w}^h r_{\mathrm{C}}(\mathbf{u}^h, p^h) \, \mathrm{d}\Omega \\
&- \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{SUPS}} \mathbf{w}^h \cdot \left(\mathbf{r}_{\mathrm{M}} \left(\mathbf{u}^h, p^h \right) \cdot \nabla \mathbf{u}^h \right) \mathrm{d}\Omega \\
&- \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega_t^e} \frac{\nabla \mathbf{w}^h}{\rho} : \left(\tau_{\mathrm{SUPS}} \mathbf{r}_{\mathrm{M}} \left(\mathbf{u}^h, p^h \right) \right) \otimes \left(\tau_{\mathrm{SUPS}} \mathbf{r}_{\mathrm{M}} \left(\mathbf{u}^h, p^h \right) \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{SUPG}} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \nabla \eta^h r_{\mathrm{T}} \left(\mathbf{u}^h, \phi^h \right) \mathrm{d}\Omega = 0. \qquad (9)
\end{aligned}
$$

Here Ω_t is divided into n_{el} spatial finite element subdomains denoted by Ω_t^e. The finite dimensional trial function spaces $\mathcal{S}_u^h$ for the velocity, $\mathcal{S}_p^h$ for the pressure, and $\mathcal{S}_\phi^h$ for the temperature, as well as the corresponding test function spaces $\mathcal{V}_u^h$, $\mathcal{V}_p^h$, and $\mathcal{V}_\phi^h$ are assumed to be of equal order. In Eq. (9), $\mathbf{h}$ and h are the natural boundary conditions for the Navier–Stokes momentum and temperature equations, respectively, and $(\Gamma_t)_{\mathrm{h}}$ is used to denote a part of the boundary where we specify these conditions, $\hat{\mathbf{u}}^h$ is the mesh velocity, and $\mathbf{r}_{\mathrm{M}}$, r_{C}, and r_{T} are the residuals of the momentum, continuity, and temperature equations, given as

$$
\begin{aligned}
\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) &= \rho \left(\left. \frac{\partial \mathbf{u}^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \\
&\quad - \nabla \cdot \boldsymbol{\sigma} \left(\mathbf{u}^h, p^h \right), \qquad (10) \\
r_{\mathrm{C}}(\mathbf{u}^h, p^h) &= \nabla \cdot \mathbf{u}^h, \qquad (11) \\
r_{\mathrm{T}}(\mathbf{u}^h, \phi^h) &= \left. \frac{\partial \phi^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \nabla \phi^h - \nabla \cdot \nu_\phi \nabla \phi^h - f^h. \qquad (12)
\end{aligned}
$$

Also in Eq. (9), τ_{SUPS}, ν_{LSIC}, and τ_{SUPG} are the stabilization parameters defined in [47] as

$$\tau_{\text{SUPS}} = \left(\frac{4}{\Delta t^2} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{G} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) + C_I \nu^2 \mathbf{G} : \mathbf{G} \right)^{-1/2}, \tag{13}$$

$$\nu_{\text{LSIC}} = \left(\text{tr}\mathbf{G}\, \tau_{\text{SUPS}} \right)^{-1}, \tag{14}$$

$$\tau_{\text{SUPG}} = \left(\frac{4}{\Delta t^2} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{G} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) + C_I \nu_\phi^2 \mathbf{G} : \mathbf{G} \right)^{-1/2}, \tag{15}$$

where

$$\text{tr}\mathbf{G} = \sum_{i=1}^{d} G_{ii} \tag{16}$$

is the trace of the element metric tensor $\mathbf{G}$, Δt is the time-step size, and C_I is a positive constant, independent of the mesh size, derived from an appropriate element-wise inverse estimate (see, e.g., [147–149]).

Remark 4 The stabilization parameters in the above equations originate from stabilized finite element methods for fluid dynamics (see, e.g., [95, 103, 150–153]). The notation "SUPS," introduced in [91], indicates that there is a single stabilization parameter for the SUPG and PSPG stabilizations, instead of two separate parameters. The notation "LSIC," introduced in [153], denotes the stabilization based on least-squares on the incompressibility constraint. The stabilization parameters were designed and studied extensively in the context of stabilized finite element formulations of linear model problems of direct relevance to fluid mechanics. These model problems include advection–diffusion and Stokes equations. The design of τ_{SUPS}, ν_{LSIC}, and τ_{SUPG} is such that optimal convergence with respect to the mesh size and polynomial order of discretization is attained for these cases. Furthermore, enhanced stability for advection-dominated flows and the ability to conveniently employ the same basis functions for velocity and pressure variables for incompressible flows are some of the attractive outcomes of this method. More recently, the stabilization parameters were derived in the context of the variational multiscale methods [54, 154] and were interpreted as the appropriate averages of the small-scale Green's function, a key mathematical object in the theory of VMS methods (see [155] for an elaboration).

Remark 5 The ALE-VMS formulation is a moving-mesh extension of the RBVMS turbulence modeling technique proposed for non-moving meshes in [34]. It was also presented in [47] for moving meshes in the context of FSI. Recently, a VMS version of the DSD/SST formulation, which is called both DSD/SST-VMST and ST-VMS, was introduced in [91, 92] for computations with moving meshes (see next section).

2.2.1 Additional VMS Modeling Terms and Stabilization Parameters

In the VMS framework, coupling between Navier–Stokes and temperature equations brought about by the Boussinesq approximation gives rise to additional modeling terms. In particular, it can be shown that the x_3-component of the linear-momentum equation and incompressibility constraint are coupled with the residual of the advection-diffusion equation, and the following terms are added to the left-hand side of Eq. (9):

$$+\sum_{e=1}^{n_{\text{el}}}\int_{\Omega_t^e}\left(\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\nabla w_3^h+\frac{1}{\rho}\frac{\partial q^h}{\partial x_3}\right)\bar{\tau}r_{\text{T}}\left(\mathbf{u}^h,\phi^h\right)\,\mathrm{d}\Omega. \tag{17}$$

The stabilization parameter $\bar{\tau}$ may be obtained following the developments in stabilized methods for advective–diffusive systems presented in [156–158], which gives the following expression for $\bar{\tau}$:

$$\bar{\tau}=-\frac{a_2}{a_1\sqrt{a_3}+a_3\sqrt{a_1}}, \tag{18}$$

where a_i's are given by

$$\begin{aligned}
a_1&=\frac{4}{\Delta t^2}+\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{G}\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)+C_I\nu^2\mathbf{G}:\mathbf{G},\\
a_2&=\frac{4}{\Delta t}\frac{\rho g}{\phi_0},\\
a_3&=\frac{4}{\Delta t^2}+\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{G}\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)+C_I\nu_\phi^2\mathbf{G}:\mathbf{G}.
\end{aligned} \tag{19}$$

Although the numerical examples presented in this article do not make use of these additional terms, a recent study of stratified turbulent flows in [60] showed that these additional VMS modeling terms can appreciably improve the performance of ALE-VMS methods for this class of problems.

2.3 *Weakly Enforced Essential Boundary Conditions*

In this section we state the formulation of the weakly enforced essential boundary conditions. This was first proposed in [56] for the advection–diffusion equation and Navier–Stokes equations of incompressible flows in an effort to improve the accuracy of stabilized and multiscale formulations in the presence of unresolved boundary layers. In [35, 36, 39], the method for the weakly enforced boundary condition was further refined and studied in a set of challenging wall-bounded turbulent flows. Here, we present the method in the context of ABL flows.

To account for the weak enforcement of the essential boundary conditions, we remove them from the trial ($\mathcal{S}_u^h$ and $\mathcal{S}_\phi^h$) and test ($\mathcal{V}_u^h$ and $\mathcal{V}_\phi^h$) function sets, and add the following terms to the left-hand side of Eq. (9):

$$
\begin{aligned}
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\mathbf{w}^h\cdot\boldsymbol{\sigma}\left(\mathbf{u}^h,\,p^h\right)\mathbf{n}\,\mathrm{d}\Gamma\\
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\left(2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}^h\right)\mathbf{n}+q^h\mathbf{n}\right)\cdot\left(\mathbf{u}^h-\mathbf{g}^h\right)\,\mathrm{d}\Gamma\\
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}^-}\mathbf{w}^h\cdot\rho\left(\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{n}\right)\left(\mathbf{u}^h-\mathbf{g}^h\right)\,\mathrm{d}\Gamma\\
&+\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\tau_{\text{TAN}}^B\left(\mathbf{w}^h-\left(\mathbf{w}^h\cdot\mathbf{n}\right)\mathbf{n}\right)\cdot\\
&\left(\left(\mathbf{u}^h-\mathbf{g}^h\right)-\left(\left(\mathbf{u}^h-\mathbf{g}^h\right)\cdot\mathbf{n}\right)\mathbf{n}\right)\,\mathrm{d}\Gamma\\
&+\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\tau_{\text{NOR}}^B\left(\mathbf{w}^h\cdot\mathbf{n}\right)\left(\left(\mathbf{u}^h-\mathbf{g}^h\right)\cdot\mathbf{n}\right)\,\mathrm{d}\Gamma\\
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\eta^h\nu_\phi\nabla\phi^h\cdot\mathbf{n}\,\mathrm{d}\Gamma\\
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\nu_\phi\nabla\eta^h\cdot\mathbf{n}\left(\phi^h-g^h\right)\,\mathrm{d}\Gamma\\
&-\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}^-}\eta^h\left(\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{n}\right)\left(\phi^h-g^h\right)\,\mathrm{d}\Gamma\\
&+\sum_{b=1}^{n_{\text{eb}}}\int_{\Gamma_t^b\bigcap(\Gamma_t)_{\text{g}}}\tau_\phi^B\eta^h\left(\phi^h-g^h\right)\,\mathrm{d}\Gamma
\end{aligned}
\tag{20}
$$

Here $(\Gamma_t)_{\text{g}}$ is the part of the boundary where the velocity $\mathbf{g}$ and temperature g are prescribed, and $\mathbf{n}$ is the unit outward normal vector. The boundary $(\Gamma_t)_{\text{g}}$ is decomposed into n_{eb} surface elements denoted by Γ_t^b, and $(\Gamma_t)_{\text{g}}^-$ is defined as the "inflow" part of $(\Gamma_t)_{\text{g}}$, i.e.,

$$
(\Gamma_t)_{\text{g}}^- = \left\{\mathbf{x}\;\middle|\;\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{n}<0,\;\forall\mathbf{x}\subset(\Gamma_t)_{\text{g}}\right\}. \tag{21}
$$

If $(\Gamma_t)_{\text{g}}$ coincides with the moving wall (rigid or flexible), then $\mathbf{g}$ is the prescribed wall velocity.

The terms in the first and sixth integral in Eq. (20) are the so-called consistency terms. It is necessary to ensure that the discrete formulation is identically satisfied by the exact solution of the Navier–Stokes and temperature equations, which, in turn, has implications on the accuracy of the discrete formulation. Also note that these terms cancel with the contributions coming from the integration-by-parts of the stress and thermal diffusion terms in Eq. (9), thus correctly removing the traction and thermal flux boundary conditions from the essential boundary. The terms in the second and seventh integral are the so-called adjoint consistency terms. Their role is less intuitive. These terms ensure that the analytical solution of the adjoint equations, when introduced in place of the linear momentum, continuity, and temperature equation test functions, also satisfies the discrete formulation. Adjoint consistency is linked to optimal convergence of the discrete solution in lower-order norms (see, e.g., [58]). The remaining terms are penalty-like, in that they penalize the deviation of the discrete solution from its prescribed value at the boundary. The terms in the third and eighth integral are also penalty-like, and lead to better satisfaction of the inflow boundary conditions.

The penalty terms are necessary to ensure the stability (or coercivity) of the discrete formulation, which may be lost due to the introduction of the consistency and adjoint consistency terms. The weak boundary condition formulation is numerically stable if

$$\tau^B_{\text{TAN}} = \tau^B_{\text{NOR}} = \frac{C^B_I \mu}{h_n}, \tag{22}$$

and

$$\tau^B_{\phi} = \frac{C^B_I \nu_{\phi}}{h_n}, \tag{23}$$

where h_n is the wall-normal element size, and C^B_I is a sufficiently large positive constant computed from an appropriate element-level inverse estimate (see, e.g., [147–149]). The constant C^B_I depends on the space dimension d, the element type (tetrahedron, hexahedron, etc.), and the polynomial order of the finite element approximation. For a linear tetrahedron, it is sufficient to take $4.0 \leq C^B_I \leq 8.0$ to obtain a stable discrete solution. The wall-normal element size may be computed from the element metric tensor:

$$h_n = (\mathbf{n} \cdot \mathbf{G}\mathbf{n})^{1/2}. \tag{24}$$

Remark 6 Rather than setting the no-slip boundary conditions exactly, the weak boundary condition formulation gives the no-slip solution only in the limit as $h_n \to 0$. As a result, coarse discretizations do not need to struggle to resolve boundary layers; the flow simply slips on the solid boundary. Because of this added flexibility, the weak boundary condition enforcement tends to produce more accurate results on meshes that are too coarse to capture the boundary layer solution. However, as

the mesh is refined to capture the boundary layer, the weak and strong boundary-condition solutions converge to the same result (see [35]).

Remark 7 Although the weak boundary condition formulation is also stable for very large values of C_I^B, we do not favor that. Large values of C_I^B place a heavy penalization on the no-slip condition, and the above-mentioned flexibility of the method is lost together with the associated accuracy benefits. We favor using a C_I^B that is just large enough to guarantee stability of the discrete formulation.

Remark 8 In reference [35], a connection was identified between the weakly enforced boundary conditions and wall functions. The latter are commonly employed in conjunction with RANS formulations of turbulent flows (see, e.g., [159, 160]). In the case of wall function formulation, a no-slip boundary condition is replaced with a tangential traction boundary condition, where the traction direction is given by that of the local slip velocity, and the traction magnitude is computed by invoking the "law-of-the-wall." This is an empirical relationship between the flow speed and the normal distance to the wall, both appropriately normalized (see, e.g., [159]). The penalty parameter τ_{TAN}^B may be defined as

$$\tau_{\text{TAN}}^B = \frac{\rho u^{*2}}{\|\mathbf{u}_{\text{TAN}}^h\|}, \tag{25}$$

where $\mathbf{u}_{\text{TAN}}^h = \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) - \left(\left(\mathbf{u}^h - \mathbf{g}^h\right) \cdot \mathbf{n}\right)\mathbf{n}\right)$ is the tangential slip velocity, and u^* is the so-called friction velocity, which, among other factors, depends on the magnitude of the slip velocity, and is computed from the law-of-the-wall formula by nonlinear iterations. It was shown in [35], however, that when the boundary layer mesh is fine enough, τ_{TAN}^B from Eq. (25) is independent of the local flow solution, and reverts to the definition given by Eq. (22). This fact is remarkable in that Eq. (22) is purely based on considerations of numerical stability, while Eq. (25) derives from the physics of wall-bounded turbulent flows. In earlier work it was observed that both the "numerics-based" and "physics-based" definitions of the penalty parameter τ_{TAN}^B give very similar results [35].

Remark 9 Recently, in [161], a new near-wall model formulation was proposed in the framework of weakly enforced no-slip conditions that is better aligned with traditional near-wall modeling approaches than its predecessors. The new model gives more accurate results for the mean flow and velocity fluctuations than its older versions, while exhibiting better numerical stability than traditional near-wall modeling techniques.

2.4 Sliding-Interface Formulation

In order to simulate the full wind turbine configuration and investigate the rotor–tower interaction, we consider an approach that makes use of a moving subdomain,

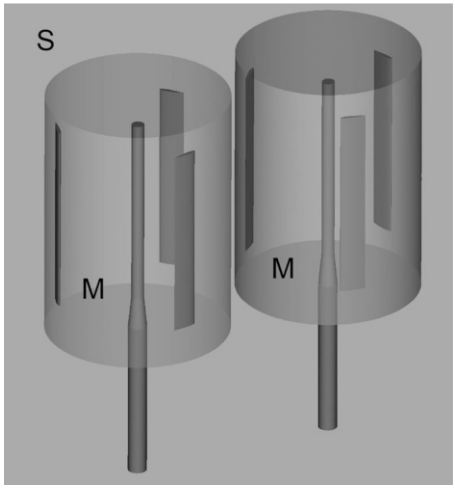


Fig. 1 Example of computational domain decomposition in the case of two VAWTs. The two cylindrical subdomains (labeled "*M*") spin with the rotors, while the remaining subdomain (labeled "*S*") is stationary

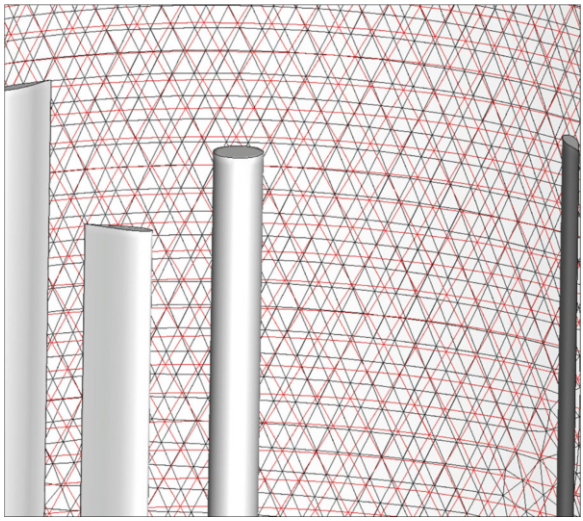


Fig. 2 Nonmatching meshes at the sliding interface between the stationary (black) and moving (red) subdomains

which encloses the entire wind turbine rotor, and a stationary subdomain that contains the rest of the wind turbine (see Fig. 1). The two domains are in relative motion and share one or several sliding cylindrical interfaces. The meshes on each side of an interface are nonmatching because of the relative motion (see Fig. 2). As a result, a numerical procedure is needed to impose the continuity of the kinematics and tractions at the stationary and rotating subdomain interfaces. Such a procedure was developed in [49] in the context of IGA for computing flows about rotating components. The advantage of IGA for rotating-component flows is that the cylindrical sliding interfaces are represented exactly and no geometry errors are incurred. In the case of standard finite elements employed here, the geometric

compatibility is only approximate. We present the sliding-interface formulation in the context of ABL flows in what follows.

Let the subscripts S and M denote the quantities pertaining to the fluid mechanics problem on the stationary and moving subdomains, respectively. The subdomain that encloses the rotor rotates with it, and the interior of the rotating subdomain is allowed to deflect to accommodate the motion of the blades. However, the motion of the outer boundary of the rotor subdomain is restricted to a rigid rotation to maintain geometric compatibility with the stationary subdomain. To enforce the compatibility of the fluid kinematics and tractions, as well as the compatibility of temperatures and heat fluxes, across the sliding interface, we add the following terms to the ALE-VMS formulation given by Eq. (9), which is now assumed to govern both the stationary and moving subdomains:

$$
\begin{aligned}
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{S}}^h - \mathbf{w}_{\mathrm{M}}^h\right) \cdot \frac{1}{2} \left(\boldsymbol{\sigma}_{\mathrm{S}} \mathbf{n}_{\mathrm{S}} - \boldsymbol{\sigma}_{\mathrm{M}} \mathbf{n}_{\mathrm{M}}\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \frac{1}{2} \left(\delta\boldsymbol{\sigma}_{\mathrm{S}} \mathbf{n}_{\mathrm{S}} - \delta\boldsymbol{\sigma}_{\mathrm{M}} \mathbf{n}_{\mathrm{M}}\right) \cdot \left(\mathbf{u}_{\mathrm{S}}^h - \mathbf{u}_{\mathrm{M}}^h\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \mathbf{w}_{\mathrm{S}}^h \cdot \rho \left\{ \left(\mathbf{u}_{\mathrm{S}}^h - \hat{\mathbf{u}}_{\mathrm{S}}^h\right) \cdot \mathbf{n}_{\mathrm{S}} \right\}_{-} \left(\mathbf{u}_{\mathrm{S}}^h - \mathbf{u}_{\mathrm{M}}^h\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \mathbf{w}_{\mathrm{M}}^h \cdot \rho \left\{ \left(\mathbf{u}_{\mathrm{M}}^h - \hat{\mathbf{u}}_{\mathrm{M}}^h\right) \cdot \mathbf{n}_{\mathrm{M}} \right\}_{-} \left(\mathbf{u}_{\mathrm{M}}^h - \mathbf{u}_{\mathrm{S}}^h\right) \mathrm{d}\Gamma \\
&+\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \frac{C_I^B \mu}{h_n} \left(\mathbf{w}_{\mathrm{S}}^h - \mathbf{w}_{\mathrm{M}}^h\right) \cdot \left(\mathbf{u}_{\mathrm{S}}^h - \mathbf{u}_{\mathrm{M}}^h\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \left(\eta_{\mathrm{S}}^h - \eta_{\mathrm{M}}^h\right) \cdot \frac{1}{2} \left(\nu_\phi \nabla\phi_{\mathrm{S}}^h \mathbf{n}_{\mathrm{S}} - \nu_\phi \nabla\phi_{\mathrm{M}}^h \mathbf{n}_{\mathrm{M}}\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \frac{1}{2} \left(\nu_\phi \nabla\eta_{\mathrm{S}}^h \mathbf{n}_{\mathrm{S}} - \nu_\phi \nabla\eta_{\mathrm{M}}^h \mathbf{n}_{\mathrm{M}}\right) \cdot \left(\phi_{\mathrm{S}}^h - \phi_{\mathrm{M}}^h\right) \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \eta_{\mathrm{S}}^h \cdot \left\{ \left(\mathbf{u}_{\mathrm{S}}^h - \hat{\mathbf{u}}_{\mathrm{S}}^h\right) \cdot \mathbf{n}_{\mathrm{S}} \right\}_{-} \left(\phi_{\mathrm{S}}^h - \phi_{\mathrm{M}}^h\right) \mathrm{d}\Gamma
\end{aligned}
$$

$$
\begin{aligned}
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \eta_{\mathrm{M}}^h \cdot \left\{ \left(\mathbf{u}_{\mathrm{M}}^h - \hat{\mathbf{u}}_{\mathrm{M}}^h \right) \cdot \mathbf{n}_{\mathrm{M}} \right\}_- \left(\phi_{\mathrm{M}}^h - \phi_{\mathrm{S}}^h \right) \mathrm{d}\Gamma \\
&+\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma_t^b \bigcap (\Gamma_t)_{\mathrm{SI}}} \frac{C_I^B \nu_\phi}{h_n} \left(\eta_{\mathrm{S}}^h - \eta_{\mathrm{M}}^h \right) \cdot \left(\phi_{\mathrm{S}}^h - \phi_{\mathrm{M}}^h \right) \mathrm{d}\Gamma
\end{aligned} \tag{26}
$$

where $\delta\boldsymbol{\sigma}$ is given by

$$
\delta\boldsymbol{\sigma}\left(\mathbf{w}, q\right) \mathbf{n} = 2\mu\boldsymbol{\varepsilon}(\mathbf{w})\mathbf{n} + q\mathbf{n}, \tag{27}
$$

$(\Gamma_t)_{\mathrm{SI}}$ is the sliding interface, and $\{\mathcal{A}\}_-$ denotes the negative part of $\mathcal{A}$, that is, $\{\mathcal{A}\}_- = \mathcal{A}$ if $\mathcal{A} < 0$ and $\{\mathcal{A}\}_- = 0$ if $\mathcal{A} \geq 0$. The structure of the terms on the sliding interface is similar to that of the weak enforcement of essential boundary conditions. Note that, in the current application, $\hat{\mathbf{u}}_{\mathrm{S}}^h = \mathbf{0}$, because the subdomain S is stationary. However, as will be shown in the later sections of this paper, the formulation is able to handle situations where both subdomains are in motion.

Remark 10 Nonmatching interface discretizations in the FSI and sliding-interface problems necessitate the use of interpolation or projection of kinematic and traction data between the nonmatching surface meshes (see, e.g., [63, 91, 92], where [92] is more comprehensive than [91]). A computational procedure, which can simultaneously handle the data transfer for IGA and finite element discretizations, was proposed in [63]. The procedure also includes a robust approach in identifying "closest points" for arbitrary shaped surfaces. While such interface projections are rather straightforward for weakly coupled FSI algorithms, they require special techniques [92, 96, 104] for strongly coupled, direct, and quasi-direct methods [92, 96, 100, 104, 162] that are monolithic-like (i.e., become monolithic for matching discretizations).

3 ST-VMS, ST-SUPS, and ST-SI

3.1 ST-VMS and ST-SUPS

We include from [117, 132, 139] the ST-VMS:

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q \\
&\quad + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q - \int_{(P_n)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P
\end{aligned}
$$

$$
\begin{aligned}
&+ \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{SUPS}}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h \right) + \nabla q^h \right] \cdot \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \nu_{\mathrm{LSIC}} \nabla \cdot \mathbf{w}^h \rho r_{\mathrm{C}}(\mathbf{u}^h) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \tau_{\mathrm{SUPS}} \mathbf{w}^h \cdot \left(\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \cdot \nabla \mathbf{u}^h \right) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{{\tau_{\mathrm{SUPS}}}^2}{\rho} \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \cdot \left(\nabla \mathbf{w}^h \right) \cdot \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&= 0, \qquad (28)
\end{aligned}
$$

where

$$
\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \qquad (29)
$$

$$
r_{\mathrm{C}}(\mathbf{u}^h) = \nabla \cdot \mathbf{u}^h \qquad (30)
$$

are the residuals of the momentum equation and incompressibility constraint. The symbol Q_n represents the ST slice between time levels n and $n+1$ (see Fig. 3), $(P_n)_{\mathrm{h}}$ is the part of the lateral boundary of that slice associated with the traction boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and n_{el} is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level. See [93, 95, 96, 114, 117] for the definitions used here for the stabilization parameters τ_{SUPS} and ν_{LSIC}. For more ways of calculating the stabilization parameters in computation of flow problems, see [38, 163–184]. These include some recent stabilization parameters and element length definitions [184] that target isogeometric discretization but are also applicable to finite element discretization.

Remark 11 The ST-SUPS can be obtained from the ST-VMS by dropping the eighth and ninth integrations.

Remark 12 One of the main differences between the ALE and ST forms of the VMS method is that the ST form retains the fine-scale time derivative term $\left.\frac{\partial \mathbf{u}'}{\partial t}\right|_{\boldsymbol{\xi}}$. Dropping this term is called the "quasi-static" assumption (see [8] for the

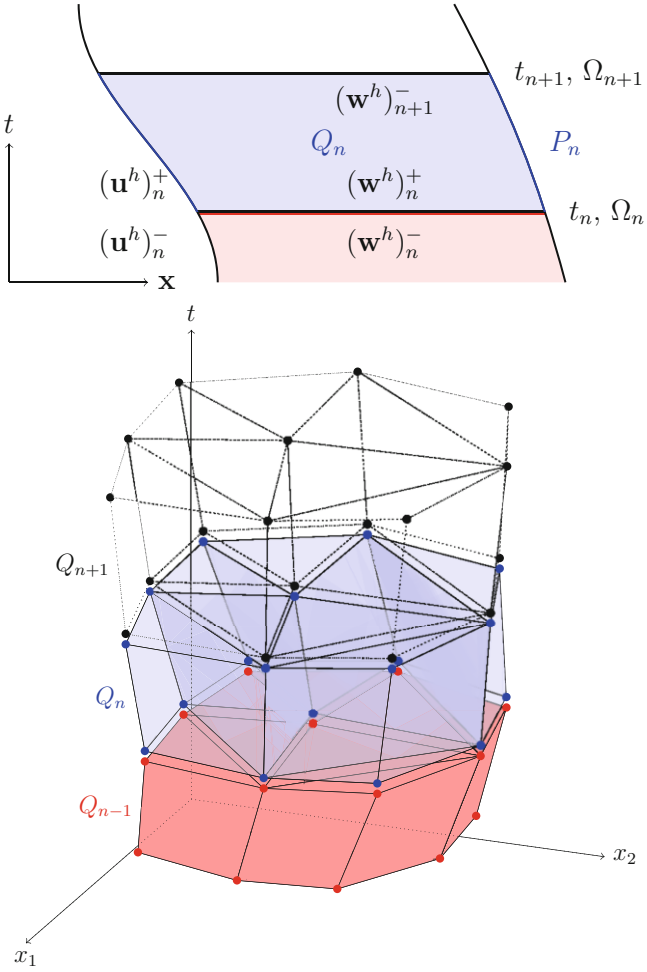


Fig. 3 ST slab in an abstract representation (top) and in a 2D context (bottom)

terminology). This is the same as the "WTSE" option in the DSD/SST formulation (see Remark 2 of [96]). We believe that this makes a significant difference, especially when the polynomial orders in space or time are higher (see [91]).

Remark 13 The thermo-fluid ST-VMS for the coupled incompressible-flow and thermal-transport equations can be found in [93, 134]. It was derived by expanding Eq. (28) to include the thermal-transport equation and the coupling between the flow and thermal-transport equations.

3.2 Rotation Representation with Constant Angular Velocity

This subsection, which is related to the ST-VMS computations, is from [114]. We use quadratic NURBS functions, as described in [91, 92, 105, 118], to represent a circular arc. We discretize time and position as follows:

$$t = \sum_{\alpha=1}^{n_{\text{ent}}} T^{\alpha}(\Theta_t(\theta)) t^{\alpha}, \tag{31}$$

$$\mathbf{x} = \sum_{\alpha=1}^{n_{\text{ent}}} T^{\alpha}(\Theta_x(\theta)) \mathbf{x}^{\alpha}. \tag{32}$$

Here n_{ent} is the number of temporal element nodes, T^{α} is the basis function, $\Theta_t(\theta)$ and $\Theta_x(\theta)$ are the secondary mappings for time and position, and t^{α} and $\mathbf{x}^{\alpha}$ are the time and position values corresponding to the basis function T^{α}. The basis functions could be finite element or NURBS basis functions. For the circular arc, $n_{\text{ent}} = 3$ and they are quadratic NURBS. The secondary mapping concept above was introduced in [91], and the velocity can be expressed as follows:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{\mathrm{d}T^{\alpha}}{\mathrm{d}\Theta_x} \frac{\mathrm{d}\Theta_x}{\mathrm{d}\theta} \mathbf{x}^{\alpha} \right) \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{\mathrm{d}T^{\alpha}}{\mathrm{d}\Theta_t} \frac{\mathrm{d}\Theta_t}{\mathrm{d}\theta} t^{\alpha} \right)^{-1}, \tag{33}$$

leading to

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{\mathrm{d}T^{\alpha}}{\mathrm{d}\Theta_x} \mathbf{x}^{\alpha} \right) \left(\sum_{\alpha=1}^{n_{\text{ent}}} \frac{\mathrm{d}T^{\alpha}}{\mathrm{d}\Theta_t} t^{\alpha} \right)^{-1} \left(\frac{\mathrm{d}\Theta_x}{\mathrm{d}\theta} \frac{\mathrm{d}\theta}{\mathrm{d}\Theta_t} \right). \tag{34}$$

Thus, the speed along the path can be specified only by modifying the secondary mapping. For a circular arc, two methods were introduced in [92, 118] and also described in [105]; one is modifying the secondary mapping for position and the other one is modifying both such that $\frac{\mathrm{d}t}{\mathrm{d}\theta}$ is constant. We note that, in theory, the secondary mapping selections do not make any difference as long as the relationship $\frac{\mathrm{d}\Theta_x}{\mathrm{d}\Theta_t}$ is the same.

In our implementation, to keep the process general, we search for the parametric coordinate θ by using an iterative solution method [92, 105, 118]. We use the latter set of the secondary mappings, having constant $\frac{\mathrm{d}t}{\mathrm{d}\theta}$.

Remark 14 When we use a secondary mapping for discretization of unknowns, the selection of the mappings affects the numerical integration accuracy in the physical domain.

3.3 ST-SI

We include from [117, 132, 139] the ST-SI.

3.3.1 Two-Side Formulation (Fluid–Fluid SI)

Labels "Side A" and "Side B" represent the two sides of the SI. We add boundary terms to Eq. (28). The boundary terms are first added separately for the two sides, using test functions $\mathbf{w}_{\mathrm{A}}^h$ and q_{A}^h and $\mathbf{w}_{\mathrm{B}}^h$ and q_{B}^h. Putting them together, the complete set of terms added becomes

$$
\begin{aligned}
&- \int_{(P_n)_{\mathrm{SI}}} \left(q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} - q_{\mathrm{A}}^h \mathbf{n}_{\mathrm{A}} \right) \cdot \frac{1}{2} \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2} \left(\left(\mathcal{F}_{\mathrm{B}}^h - \left| \mathcal{F}_{\mathrm{B}}^h \right| \right) \mathbf{u}_{\mathrm{B}}^h \right. \\
&\left. - \left(\mathcal{F}_{\mathrm{B}}^h - \left| \mathcal{F}_{\mathrm{B}}^h \right| \right) \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{A}}^h \cdot \frac{1}{2} \left(\left(\mathcal{F}_{\mathrm{A}}^h - \left| \mathcal{F}_{\mathrm{A}}^h \right| \right) \mathbf{u}_{\mathrm{A}}^h \right. \\
&\left. - \left(\mathcal{F}_{\mathrm{A}}^h - \left| \mathcal{F}_{\mathrm{A}}^h \right| \right) \mathbf{u}_{\mathrm{B}}^h \right) \mathrm{d}P \\
&+ \int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{n}_{\mathrm{B}} \cdot \mathbf{w}_{\mathrm{B}}^h + \mathbf{n}_{\mathrm{A}} \cdot \mathbf{w}_{\mathrm{A}}^h \right) \frac{1}{2} \left(p_{\mathrm{B}}^h + p_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h \right) \cdot \left(\hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu \left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^h) + \boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^h) \right) \right) \mathrm{d}P \\
&- \gamma \int_{(P_n)_{\mathrm{SI}}} \hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu \left(\boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{B}}^h \right) + \boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{A}}^h \right) \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&+ \int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P, \qquad (35)
\end{aligned}
$$

where

$$\mathcal{F}_{\mathrm{B}}^h = \mathbf{n}_{\mathrm{B}} \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \hat{\mathbf{u}}_{\mathrm{B}}^h \right), \qquad (36)$$

$$\mathcal{F}_{\mathrm{A}}^h = \mathbf{n}_{\mathrm{A}} \cdot \left(\mathbf{u}_{\mathrm{A}}^h - \hat{\mathbf{u}}_{\mathrm{A}}^h \right), \qquad (37)$$

$$h = \min(h_{\mathrm{B}}, h_{\mathrm{A}}), \qquad (38)$$

$$h_{\mathrm{B}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{B}} \cdot \nabla N_a^{\alpha} \right| \right)^{-1} \text{(for Side B)}, \qquad (39)$$

$$h_{\mathrm{A}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{A}} \cdot \nabla N_a^{\alpha} \right| \right)^{-1} \text{(for Side A)}, \qquad (40)$$

$$\hat{\mathbf{n}}_{\mathrm{B}} = \frac{\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}}{\| \mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}} \|}. \qquad (41)$$

Here, $(P_n)_{\text{SI}}$ is the SI in the ST domain, $\hat{\mathbf{u}}$ is the mesh velocity, n_{ens} and n_{ent} are the number of spatial and temporal element nodes, N_a^α is the basis function associated with spatial and temporal nodes a and α, $\gamma = 1$, and C is a nondimensional constant. For our element length definition, we set $C = 1$.

Remark 15 A number of remarks were provided in [117] to explain the added terms and to comment on related interpretations. We refer the reader interested in such details to [117].

3.3.2 One-Side Formulation (Fluid–Solid SI)

Sometimes we prefer to specify on solid surfaces weakly imposed Dirichlet conditions for the fluid [56, 185]. In such cases we use the ST-SI version where the SI is between the fluid and solid domains. This version is obtained (see [117]) by starting with the terms added for Side B and replacing the Side A velocity with the velocity $\mathbf{g}^h$ coming from the solid domain. Then the terms added to Eq. (28) to represent the weakly imposed Dirichlet conditions become

$$
\begin{aligned}
&- \int_{(P_n)_{\text{SI}}} q_{\text{B}}^h \mathbf{n}_{\text{B}} \cdot \mathbf{u}_{\text{B}}^h \, \mathrm{d}P - \int_{(P_n)_{\text{SI}}} \rho \mathbf{w}_{\text{B}}^h \cdot \mathcal{F}_{\text{B}}^h \mathbf{u}_{\text{B}}^h \, \mathrm{d}P \\
&+ \int_{(P_n)_{\text{SI}}} q_{\text{B}}^h \mathbf{n}_{\text{B}} \cdot \mathbf{g}^h \, \mathrm{d}P \\
&+ \int_{(P_n)_{\text{SI}}} \rho \mathbf{w}_{\text{B}}^h \cdot \frac{1}{2} \left(\left(\mathcal{F}_{\text{B}}^h + \left| \mathcal{F}_{\text{B}}^h \right| \right) \mathbf{u}_{\text{B}}^h \right. \\
&\left. + \left(\mathcal{F}_{\text{B}}^h - \left| \mathcal{F}_{\text{B}}^h \right| \right) \mathbf{g}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\text{SI}}} \mathbf{w}_{\text{B}}^h \cdot \left(\mathbf{n}_{\text{B}} \cdot \boldsymbol{\sigma}_{\text{B}}^h \right) \mathrm{d}P \\
&- \gamma \int_{(P_n)_{\text{SI}}} \mathbf{n}_{\text{B}} \cdot 2\mu \boldsymbol{\varepsilon} \left(\mathbf{w}_{\text{B}}^h \right) \cdot \left(\mathbf{u}_{\text{B}}^h - \mathbf{g}^h \right) \mathrm{d}P \\
&+ \int_{(P_n)_{\text{SI}}} \frac{\mu C}{h_{\text{B}}} \mathbf{w}_{\text{B}}^h \cdot \left(\mathbf{u}_{\text{B}}^h - \mathbf{g}^h \right) \mathrm{d}P. \qquad (42)
\end{aligned}
$$

Remark 16 New versions of the fluid–fluid SI and fluid–solid SI were introduced in [139, 186], including those with new element length definitions [186]. Some of the new element length definitions were introduced in [186], and some were straightforward extensions of the one in [184]. The option γ was introduced in [139].

Remark 17 The thermo-fluid ST-SI for the coupled incompressible-flow and thermal-transport equations can be found in [134].

4 Wind-Turbine Isogeometric Structural Modeling

The main structural components of wind turbines are modeled using a combination of the displacement-based Kirchhoff–Love shell [17, 44] and bending-stabilized cable [187] formulations. The latter may be deployed in the regime of Euler–Bernoulli beams as shown in [187]. The shell formulation is used to represent the wind-turbine rotor, nacelle and tower, while the beam/cable formulation is used to describe the main shaft, struts in the VAWT design, and mooring cables in offshore wind applications (see Fig. 4 and references [67, 69, 70] for some examples). Both are discretized using IGA [21, 26] based on NURBS [188]. This approach gives a good combination of structural-mechanics accuracy due to the higher-order and higher-continuity representation of the geometry and solution, and efficiency due to the lack of rotational degrees of freedom in the formulation.

The variational formulation of structural mechanics is obtained from the principle of virtual work (see, e.g., [189]), and is stated as: find the structural velocity $\mathbf{u}_2^h \in S_y^h$, such that $\forall \mathbf{w}_2^h \in V_y^h$:

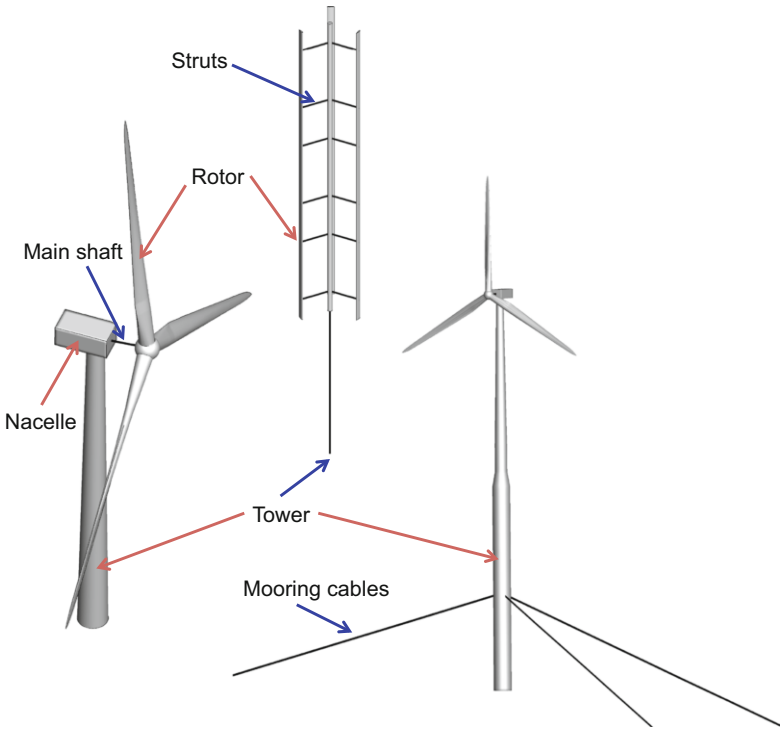


Fig. 4 NURBS-based IGA structural model of a HAWT, VAWT, and floating HAWT

$$\int_{\Omega_t^s} \mathbf{w}_2^h \cdot \rho_2 \left(\frac{\mathrm{d}\mathbf{u}_2^h}{\mathrm{d}t} - \mathbf{f}_2 \right) \mathrm{d}\Omega + \int_{\Omega_0^s} (\delta \mathbf{E} \,:\, \mathbf{S}) \,\mathrm{d}\Omega$$
$$- \int_{(\Gamma_t^s)_{2h}} \mathbf{w}_2^h \cdot \mathbf{h_2} \,\mathrm{d}\Gamma = \mathbf{0}, \tag{43}$$

where ρ_2 is the structural density, $\mathbf{f}_2$ is the body force per unit mass, $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor, and $\delta\mathbf{E}$ is the variation of the Green–Lagrange strain measure. Ω_0^s and Ω_t^s denote the structural mechanics domain in the reference and current configurations, respectively, and $(\Gamma_t^s)_{2h} \in \Gamma_t^s$ denotes a part of the boundary where the traction $\mathbf{h_2}$ is specified.

4.1 Kirchhoff–Love Shell Formulation

With assumptions on the kinematics, the 3D solid formulation may be reduced to that posed on the shell midsurface. We use Γ_0^s and Γ_t^s to denote the shell midsurface reference and deformed configurations, respectively. It assumed in the Kirchhoff–Love theory that the shell director remains normal to its midsurface during the deformation, which implies that the transverse shear strains are zero. Normal through-thickness strains are also neglected. As a result, only in-plane stress and strain measures are considered. As a strain measure we adopt the Green–Lagrange strain tensor, defined in the local coordinate system as

$$\overline{\mathbf{E}} = \overline{\boldsymbol{\varepsilon}} + \xi_3 \overline{\boldsymbol{\kappa}}, \tag{44}$$

where $\overline{\boldsymbol{\varepsilon}}$ and $\overline{\boldsymbol{\kappa}}$ are the membrane-strain and curvature-change tensors, and ξ_3 is the through-thickness coordinate. (See [44, 190] for more details.)

We assume the St. Venant–Kirchhoff constitutive law, and write the following stress–strain relationship in the local coordinate system:

$$\overline{\mathbf{S}} = \overline{\mathbf{C}}\,\overline{\mathbf{E}}, \tag{45}$$

where $\overline{\mathbf{S}}$ is a vector of components of the second Piola–Kirchhoff stress tensor in the local coordinate system, and $\overline{\mathbf{C}}$ is a constitutive material matrix, which is symmetric.

To model a laminated composite blade structure, the Classical Laminated-Plate Theory (CLPT) [191] is employed. The shell thickness is denoted by t_{th}, the thickness of the k^{th} ply by t_k, and its centroid by $\overline{z}_k$ (see Fig. 5). The extensional, coupling, and bending stiffnesses, denoted by $\mathbf{K}_{\text{exte}}$, $\mathbf{K}_{\text{coup}}$, and $\mathbf{K}_{\text{bend}}$, respectively, may be computed for *any* layup as

$$\mathbf{K}_{\text{exte}} = \int_{t_{th}} \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k t_k, \tag{46}$$

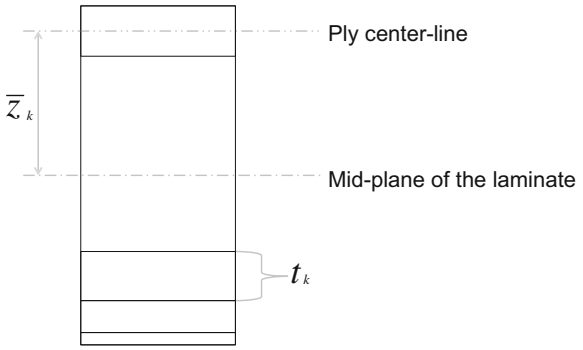


Fig. 5 Composite layup with nonuniform and nonsymmetric distribution of the lamina

$$\mathbf{K}_{\text{coup}} = \int_{t_{th}} \xi_3 \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k t_k \overline{z_k}, \tag{47}$$

$$\mathbf{K}_{\text{bend}} = \int_{t_{th}} {\xi_3}^2 \overline{\mathbf{C}}\, d\xi_3 = \sum_{k=1}^{n} \overline{\mathbf{C}}_k \left(t_k \overline{z_k^2} + \frac{t_k^3}{12} \right). \tag{48}$$

Here, $\overline{\mathbf{C}}$ is a constitutive material matrix given by

$$\overline{\mathbf{C}}_k = \mathbf{T}^T(\phi_k)\, \overline{\mathbf{C}}_{\text{ort}}\, \mathbf{T}(\phi_k), \tag{49}$$

$$\mathbf{T}(\phi) = \begin{bmatrix} \cos^2\phi & \sin^2\phi & \sin\phi\cos\phi \\ \sin^2\phi & \cos^2\phi & -\sin\phi\cos\phi \\ -2\sin\phi\cos\phi & 2\sin\phi\cos\phi & \cos^2\phi - \sin^2\phi \end{bmatrix}, \tag{50}$$

where ϕ denotes the fiber orientation angle in each ply and $\mathbf{C}_{\text{ort}}$ is the constitutive matrix for an orthotropic material written with respect to the principal material axes (see [15] for more details).

With the above definitions the complete variational formulation of the rotation-free Kirchhoff–Love shell is stated as follows: find the velocity of the shell midsurface $\mathbf{u}_2^h \in \mathcal{S}_y^h$, such that $\forall \mathbf{w}_2^h \in \mathcal{V}_y^h$,

$$\begin{aligned} &\int_{\Gamma_t^s} \mathbf{w}_2^h \cdot \overline{\rho_2} t_{th} \left(\frac{\mathrm{d}\mathbf{u}_2^h}{\mathrm{d}t} - \mathbf{f}_2^h \right) \mathrm{d}\Gamma \\ &+ \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{K}_{\text{exte}} \overline{\boldsymbol{\varepsilon}}^h + \mathbf{K}_{\text{coup}} \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\ &+ \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{K}_{\text{coup}} \overline{\boldsymbol{\varepsilon}}^h + \mathbf{K}_{\text{bend}} \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\ &- \int_{(\Gamma_t^s)_{2h}} \mathbf{w}_2^h \cdot \mathbf{h_2}\, \mathrm{d}\Gamma = 0. \end{aligned} \tag{51}$$

Here, $\overline{\rho_2}$ is the through-thickness-averaged shell density given by

$$\overline{\rho_2} = \frac{1}{t_{th}} \int_{t_{th}} \rho_2 \, \mathrm{d}\xi_3. \tag{52}$$

For real-life structures, the shell midsurface is typically described using a patch-wise smooth (C^1- or higher-order continuous) geometric mapping with reduced regularity of the mapping (C^0-continuity) at the patch interfaces. This reduction in continuity makes the integrals involving the bending terms ill defined. To alleviate this problem, the integrations in Eq. (51) are carried out patch-wise and the formulation is augmented on the left-hand side with the term

$$+ \int_{\Gamma_0^b} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{K}^b_{\text{bend}} \overline{\boldsymbol{\kappa}}^h \, \mathrm{d}\Gamma, \tag{53}$$

which represents the contribution of bending-strip patches that are placed at the NURBS-patch interfaces. The material in bending strip patches is assumed to have zero mass, zero membrane stiffness, and non-zero bending stiffness (denoted by $\mathbf{K}^b_{\text{bend}}$ in the above expression) only in the direction transverse to the patch interface. This results in a correct transfer of the bending moment between the patches (see [15, 17, 44] for more details).

Remark 18 Recently, in [192], a method for patch coupling based on the Nitsche approach was developed for Kirchhoff–Love shells. In this case, C^0-continuity of the basis functions is no longer required at patch interfaces, which simplifies model building significantly.

4.2 Bending-Stabilized Cable Formulation

We denote the cable middle curve as L_0^s and L_t^s in the reference and deformed configuration, respectively. The cable formulation is written purely in terms of displacement degrees of freedom. The variational formulation is derived from the principle of virtual work with the appropriate assumptions on the kinematics, and may be stated as: find the velocity $\mathbf{u}_2^h \in \mathcal{S}_y^h$, such that $\forall \mathbf{w}_2^h \in \mathcal{V}_y^h$

$$\begin{aligned}
&\int_{L_t^s} \mathbf{w}_2^h \cdot \overline{\rho_2} A_0 \left(\frac{\mathrm{d}\mathbf{u}_2^h}{\mathrm{d}t} - \mathbf{f}_2^h \right) \mathrm{d}L \\
&+ \int_{L_0^s} \delta\overline{\varepsilon}^h \cdot A_0 E_0 \overline{\varepsilon}^h \, \mathrm{d}L \\
&+ \int_{L_0^s} \delta\overline{\kappa}^h \cdot I_0 E_0 \overline{\kappa}^h \, \mathrm{d}L \\
&- \int_{(L_t^s)_{2h}} \mathbf{w}_2^h \cdot \mathbf{h_2} \, \mathrm{d}L = 0,
\end{aligned} \tag{54}$$

where A_0, I_0, and E_0 are the cable cross-section area, second moment of area and Young's modulus, respectively. The formulation given by Eq. (51) models the membrane effects, as well as bending action that is confined to the osculating plane of the middle curve. For a complete derivation of the bending-stabilized cable formulation, see [187].

Remark 19 In the present work the cable-shell coupling is achieved through common control points. In the case when multiple cables are joined together, the bending-strip formulation for rotation-free Kirchhoff–Love shells [15, 17, 44] is adapted for the cable case.

5 FSI Coupling and Mesh Update

5.1 FSI Coupling Strategy for Wind-Turbine Simulations

In this section we briefly summarize our FSI coupling procedures for wind-turbine simulations. Simulations based ALE-VMS make use of the Generalized-α time integration method [193]. Within each time step, the coupled equations are solved using an inexact Newton approach. For every Newton iteration the following steps are performed. (1) We obtain the fluid solution increment holding the structure and mesh fixed. (2) We update the fluid solution, compute the aerodynamic force on the structure and compute the structural solution increment. The aerodynamic force at control points or nodes is computed using the conservative definition (see, e.g., [194, 195] for the importance of using the conservative definitions of fluxes near essential boundaries and in coupled problems). (3) We update the structural solution, and use the equations of linear elastostatics and Jacobian-based stiffening [196–198] to update the fluid mechanics mesh position and velocity. The above three-step iteration is repeated until convergence to an appropriately coupled discrete solution is achieved. The proposed approach, also referred to as "block-iterative coupling" (see [96, 100, 162] for the terminology), is stable because the wind-turbine blades are relatively heavy structures. For a variety of mesh update strategies, the reader is referred to [96, 100, 105, 107, 118, 199–203].

5.2 A Novel Mesh Update Technique for Sliding Interfaces in Motion

To enable the FSI simulations of wind turbines with more complicated rotor motion than just rotation around a fixed axis (e.g., yawing rotation or gyroscopic-like behavior of floating wind turbines), a new mesh moving technique was developed in [67] and is presented in what follows.

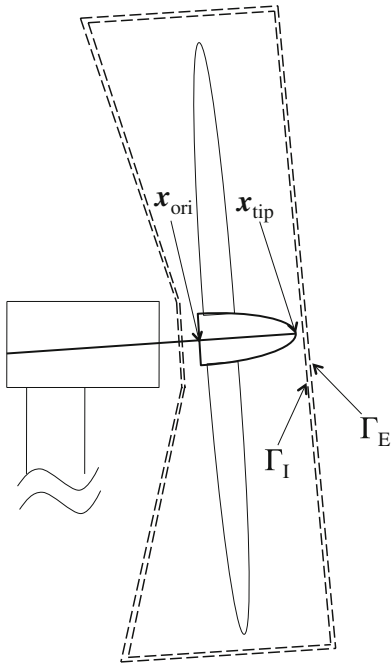


Fig. 6 Illustration of the wind turbine, the sliding interface (dashed line), and the key locations on the nacelle. A slight offset is used to illustrate that the sliding interface has two definitions, Γ_I and Γ_E. In the numerical formulation the two interfaces occupy the same region in 3D space

We denote by Γ_I and Γ_E the two sides of the sliding interface coming from the interior and exterior subdomains (see Fig. 6). We begin with describing the motion of Γ_I. For this, we let $\mathbf{x}$ and $\mathbf{X}$ denote the position vector of the points on Γ_I in the current and reference configuration, respectively. We define $\mathbf{x}_{ori}$ and $\mathbf{x}_{tip}$ to be the current-configuration positions of the back and tip of the nacelle (see Fig. 6), and we let $\mathbf{X}_{ori}$ and $\mathbf{X}_{tip}$ be their reference-configuration counterparts. We restrict the motion of Γ_I to be that of a rigid object and write

$$\mathbf{x} = \mathbf{R}(\mathbf{X} - \mathbf{X}_{ori}) + \mathbf{x}_{ori}. \tag{55}$$

While $\mathbf{x}_{ori}$ is obtained directly from the motion of the wind-turbine structure, the main challenge here is to obtain a suitable rotation matrix $\mathbf{R}$ in the above equation. For this, we extract the instantaneous mean angular velocity of the wind-turbine rotor as

$$\boldsymbol{\omega} = \mathbf{J}^{-1}\mathbf{m}, \tag{56}$$

where $\mathbf{J}$ is the rotor moment-of-inertia tensor in the current configuration given by

$$\begin{aligned}\mathbf{J} = \int_{\Omega_R} & \rho(\mathbf{x} - \mathbf{x}_{ori}) \cdot (\mathbf{x} - \mathbf{x}_{ori})\mathbf{I} \\ & - \rho(\mathbf{x} - \mathbf{x}_{ori}) \otimes (\mathbf{x} - \mathbf{x}_{ori})\, d\Omega,\end{aligned} \tag{57}$$

and $\mathbf{m}$ is the rotor angular momentum given by

$$\mathbf{m} = \int_{\Omega_R} (\mathbf{x} - \mathbf{x}_{ori}) \times \rho \mathbf{u} \, d\Omega. \tag{58}$$

In the above integrals Ω_R is the rotor structural domain in the current configuration, ρ is the rotor material density, $\mathbf{u}$ is the rotor velocity, and $\mathbf{I}$ is the identity tensor. This technique of extracting $\boldsymbol{\omega}$ was employed earlier in [107] to remove a spinning component of the structure motion in FSI modeling of parachute clusters.

Given the rotor instantaneous mean angular velocity $\boldsymbol{\omega}$, we compute the rotation matrix $\mathbf{R}$ needed in Eq. (55) using the following ODE (see, e.g., [189]):

$$\frac{d}{dt}\mathbf{R} = \boldsymbol{\omega} \times \mathbf{R}, \tag{59}$$

where the cross-product is taken column-wise.

To handle Γ_E in the computations we use a similar approach. We also restrict the motion of Γ_E to be that of a rigid object and write

$$\mathbf{x} = \mathbf{R}_\tau (\mathbf{X} - \mathbf{X}_{ori}) + \mathbf{x}_{ori}, \tag{60}$$

where the rotation matrix $\mathbf{R}_\tau$ is obtained from the ODE:

$$\frac{d}{dt}\mathbf{R}_\tau = \boldsymbol{\omega}_\tau \times \mathbf{R}_\tau. \tag{61}$$

The above ODE is driven by the angular velocity vector $\boldsymbol{\omega}_\tau$, which we define as

$$\boldsymbol{\omega}_\tau = \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \mathbf{n}_{rot})\mathbf{n}_{rot}, \tag{62}$$

where

$$\mathbf{n}_{rot} = \frac{\mathbf{x}_{tip} - \mathbf{x}_{ori}}{\|\mathbf{x}_{tip} - \mathbf{x}_{ori}\|}. \tag{63}$$

Equation (62) effectively removes the spinning component from the motion of Γ_E, as desired. The location of the nacelle tip $\mathbf{x}_{tip}$ is also obtained directly from the motion of the wind-turbine structure.

Both Eqs. (59) and (61) are integrated in time using the midpoint rule, which guarantees that $\mathbf{R}$ and $\mathbf{R}_\tau$ retain their orthonormal property, and thus remain true rotation matrices. This result is due to [204].

Long-time FSI simulations using the mesh-moving technique described above may encounter a slight misalignment between Γ_I and Γ_E. In this case, it is necessary to periodically correct the motion of Γ_E to make sure that it is aligned with Γ_I. This may be accomplished using a modified version of Eq. (60):

$$\mathbf{x} = \mathbf{R}_{cor}\mathbf{R}_\tau (\mathbf{X} - \mathbf{X}_{ori}) + \mathbf{x}_{ori}, \tag{64}$$

where $\mathbf{R}_{\text{cor}}$ is the rotation matrix between $\mathbf{n}_{\text{I}}$ and $\mathbf{n}_{\text{E}}$, the two outward unit normal vectors to Γ_{I} and Γ_{E}, respectively. (That is, $\mathbf{R}_{\text{cor}}$ is such that $\mathbf{n}_{\text{I}} = \mathbf{R}_{\text{cor}}\mathbf{n}_{\text{E}}$.) The normal vectors may be taken, for example, on the inflow plane of the cylindrical domains. The matrix $\mathbf{R}_{\text{cor}}$ can be computed explicitly using the Rodrigues formula [205] as

$$\mathbf{R}_{\text{cor}} = \cos\theta\mathbf{I} + \sin\theta\boldsymbol{\Upsilon} + (1 - \cos\theta)\mathbf{r} \otimes \mathbf{r}, \tag{65}$$

where θ is the angle between the two normal vectors, $\mathbf{r}$ is the rotation axis defined as

$$\mathbf{r} = \frac{\mathbf{n}_{\text{I}} \times \mathbf{n}_{\text{E}}}{\|\mathbf{n}_{\text{I}} \times \mathbf{n}_{\text{E}}\|}, \tag{66}$$

and the components of $\boldsymbol{\Upsilon}$ are given by

$$\Upsilon_{ik} = \epsilon_{ijk} r_j, \tag{67}$$

where ϵ_{ijk} are the components of the alternator tensor.

6 Aerodynamic Simulations of a 5 MW Wind-Turbine Rotor in Atmospheric Boundary Layer Flow

In this section we apply the methods presented to investigate the behavior of a full-scale offshore wind-turbine rotor placed in a thermally stratified ABL. For this, a standalone 3D large-eddy simulation (LES) computation of an ABL is first performed on a cube with dimension 400 m and uniform mesh size of 5 m. The LES [206, 207] employs a mixed spectral-finite difference algorithm and dynamic eddy viscosity and eddy diffusivity models to simulate stratified flows. The reference temperature is set to 260 K, and a vertical temperature gradient of 0.01 K/m is prescribed starting at 100 m above ground. The geostrophic wind speed is set to 8 m/s, and the Coriolis parameter $f_c = 1.39 \times 10^{-4}\,\text{s}^{-1}$ is employed in the computation. The computational set-up is similar to that in [208]. The data from the LES computation is used as inlet boundary conditions for the ALE-VMS wind-turbine rotor computation, which is described in what follows.

The wind-turbine rotor problem domain is a cylinder 240 m in length and diameter. A 5 MW wind-turbine rotor with 61 m blades is placed in the cylindrical domain. The wind turbine blade is obtained by scaling down the 100 m baseline blade designed by Sandia National Laboratory (SNL), based on the geometry of the 61 m baseline offshore designs employed in the NREL, DOWEC, and UpWind projects [209]. From the aerodynamics side, the wind turbine blade is identical to the one used in [8, 9, 17, 63, 113, 114, 143] and only has minor modifications to internal structures and composite layup to increase the load carrying capacity (the

Table 1 Geometry data for the SNL blade

Blade fraction	Chord length (m)	Twist angle (deg)	Pitch angle fraction	Airfoil type
0.000	5.694	13.308	0.500	Cylinder
0.195	7.628	12.915	0.380	DU99-W-405
0.358	6.923	9.166	0.375	DU97-W-300
0.602	5.417	4.743	0.375	DU93-W-210
0.732	4.621	2.735	0.375	NACA-64-618
0.765	4.422	2.348	0.375	NACA-64-618
0.846	3.925	1.380	0.375	NACA-64-618
0.895	3.619	0.799	0.375	NACA-64-618
0.944	2.824	0.280	0.375	NACA-64-618
0.957	2.375	0.210	0.375	NACA-64-618
0.972	1.836	0.140	0.375	NACA-64-618
0.986	1.208	0.070	0.375	NACA-64-618
1.000	0.100	0.000	0.375	NACA-64-618

The nomenclature follows that used in [8]. "Pitch Angle Fraction" is the distance from the leading edge of the blade pitch axis expressed as a cord-length fraction

structural details will be given in a later section). The details of the blade geometry are provided in Table 1.

The rotor spins with a prescribed, constant angular velocity, and the problem domain spins with it. The problem domain is discretized using triangular prisms in the boundary layer region and tetrahedra elsewhere, resulting in 7,431,784 linear elements (see Fig. 7). The total of 10 layers of boundary-layer elements are employed, and the size of the first element in the wall-normal direction is 1 cm. The mesh is refined in the inner region of the domain in order to better resolve the ABL turbulent flow that impacts the rotor.

To impose turbulent inlet velocity and temperature boundary conditions, the solution data is transferred from the structured grid of the LES simulation to the unstructured mesh of the spinning rotor problem. The spinning rotor domain is immersed into a larger LES domain and the velocity and temperature data assigned to the nodes of the cylinder inflow plane and lateral boundaries is obtained by interpolating the finite-difference solution (see Fig. 8). At the outlet, traction boundary conditions are prescribed. The distribution of outlet traction is obtained by computing the problem on a non-spinning domain, with the rotor removed, and with zero outlet traction boundary conditions. The inlet tractions produced as a result of this computation are assigned as outlet traction boundary conditions in the rotating-domain computation. We note that a similar concept was used in [93] for detailed thermo-fluid analysis of the rear tires of a ground vehicle. In [93], the thermo-fluid computation over the global domain (including all the tires) with a reasonable mesh refinement is followed by a higher-resolution computation over the local domain containing the rear set of tires, with the boundary and initial conditions coming

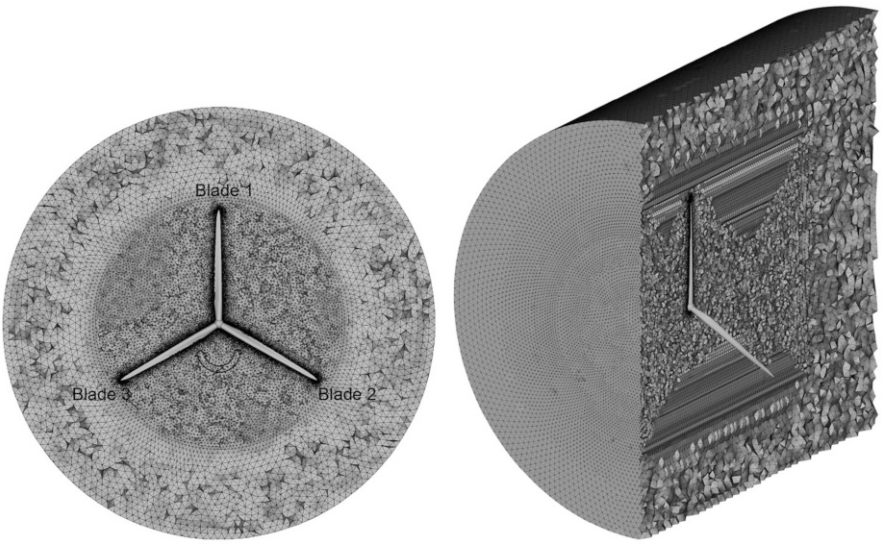


Fig. 7 Wind-turbine rotor in an ABL. Mesh of computational domain in 2D and 3D views. The rotor blades are numbered clockwise and arrow indicates rotation direction

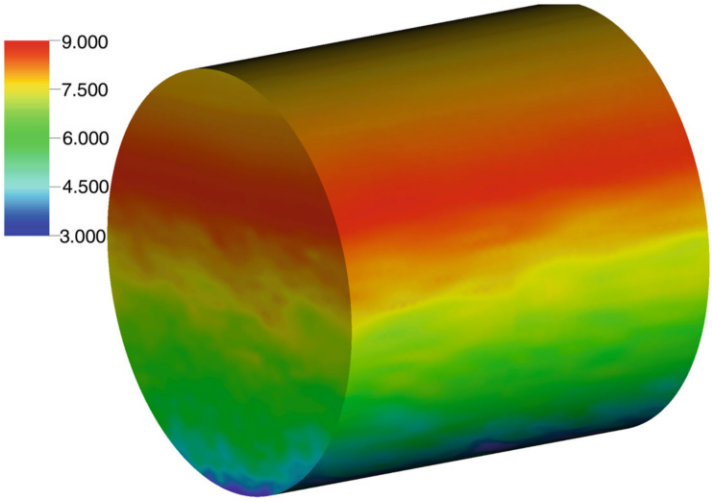


Fig. 8 Wind-turbine rotor in an ABL. Velocity (in m/s) initial and boundary conditions obtained by "immersing" the cylindrical spinning domain into a larger LES domain and performing interpolation of the LES solution

from the data computed over the global domain. The data computed over the global domain is stored using the data compression method introduced in [145].

Figure 9 shows the mean velocity and temperature as a function of the vertical coordinate, at $x_1 = 120\,\text{m}$ and $x_1 = 240\,\text{m}$ locations downstream of the inflow.

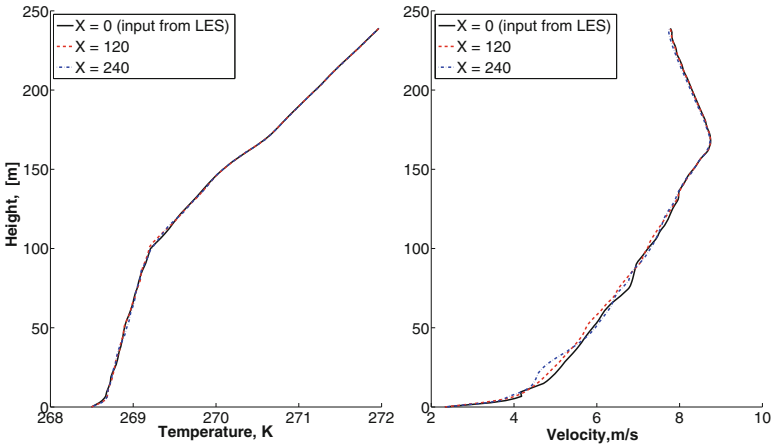


Fig. 9 Wind-turbine rotor in an ABL. Mean velocity and temperature as a function of the vertical coordinate

The results are compared to the mean of the velocity and temperature imposed on the inlet, which corresponds to the interpolated LES data. The two profiles are very close, suggesting that ALE-VMS and LES ABL are mutually consistent.

For the wind-turbine simulation the rotor speed is set to 9 rpm. This rotor speed chosen gives the optimal tip-speed ratio for 8 m/s wind speed [210], which is also the geostrophic wind speed used in the computation. The time-step size of 2.0×10^{-4} s is employed. The flow is initialized using the LES data interpolated to the interior of the rotor mesh, and the computation is started impulsively. Figure 10 shows vorticity isosurfaces at $t = 8.5$ s. Due to the presence of the inversion layer, tip vortices travel with different speeds, faster near the top and slower near the bottom of the domain. As a result, the perfect helical pattern of tip vortices, which is expected in the case of uniform flow, is no longer present. As the rotor turns and blades travel in and out of the inversion layer, they introduce a certain amount of mixing in the flow, which propagates downstream and gives a complex and largely unstudied wake behavior.

The next set of figures examine the time-dependent rotor loads. The rotor-thrust time history is shown in Fig. 11, where thrust is plotted for each individual blade (see Fig. 7 for blade numbering). The presence of ABL produces an 18% fluctuation in the thrust load during the cycle. Figure 12 shows rotor-torque time history for each individual blade, which exhibits an even larger variation during the cycle.

7 Simulation of Rotor–Tower Interaction

We present aerodynamics simulations of a three-blade, high-solidity VAWT with rated power of 3.5 kW, taken from reference [68]. The prototype is a Darrieus H-type turbine designed by Cleanfield Energy Corporation. Full-scale tests for this

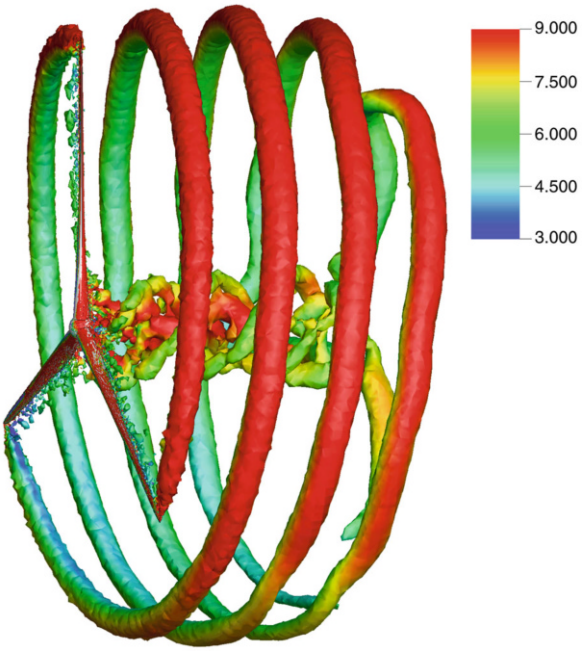


Fig. 10 Wind-turbine rotor in an ABL. Vorticity isosurfaces colored by flow speed (in m/s) at $t = 8.5$ s

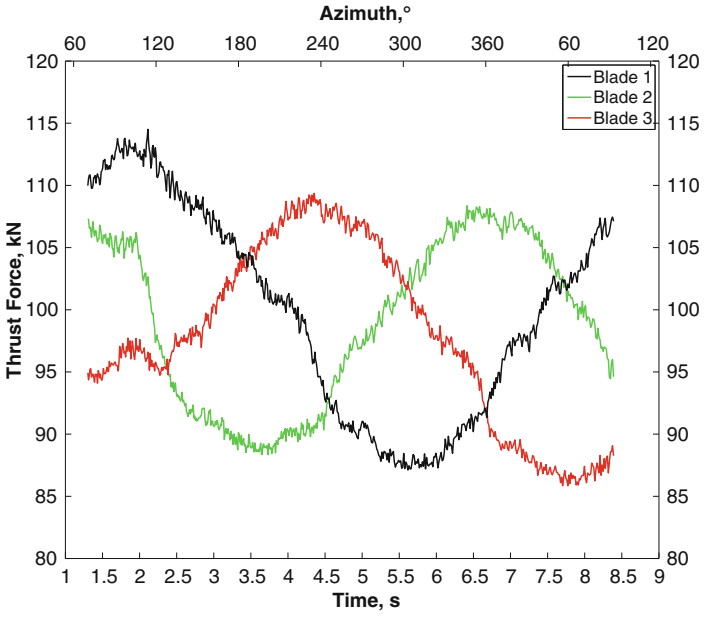


Fig. 11 Wind-turbine rotor in an ABL. Time history of the thrust force acting on each blade

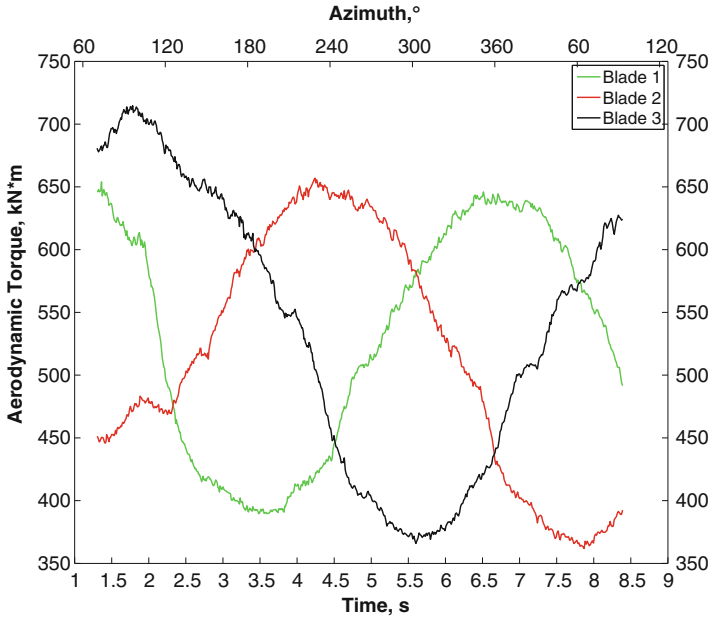


Fig. 12 Wind-turbine rotor in an ABL. Time history of the aerodynamic torque acting on each blade

turbine were conducted in the National Research Council (NRC) low-speed wind tunnel at McMaster University (see Fig. 13). Experimental studies for this turbine focused on the application of VAWTs in urban areas [211].

The turbine has a tower height of 7 m. The blades, 3 m in height, are connected to the tower by the struts of length 1.25 m. This value is taken as the rotor radius. A symmetric NACA0015 airfoil profile with chord length of 0.4 m is employed along the entire length of the blades.

The computations were carried out for constant inflow wind speed of 10 m/s, and constant, fixed rotor speed of 115 rpm. This set-up corresponds to the tip speed ratio of 1.5, which gave maximum rotor power as reported in [211, 212]. However, it was also reported for the wind tunnel tests that the control mechanism employed was able to maintain an *average* rotor speed of 115 rpm with the deviation of ± 2.5 rpm. This means the actual rotor speed was never constant.

The air density and viscosity are set to $1.23\,\mathrm{kg/m^3}$ and $1.78 \times 10^{-5}\,\mathrm{kg/(m{\cdot}s)}$, respectively. On the inflow, the wind speed of 10 m/s is prescribed. On the top, bottom and side surfaces of the stationary domain no-penetration boundary conditions are prescribed, while zero traction boundary condition is set on the outflow. No-slip boundary conditions are imposed weakly on the rotor blades and tower. The struts are not modeled in this work to reduce computational cost. The time step is set to 1.0×10^{-5} s for all cases.

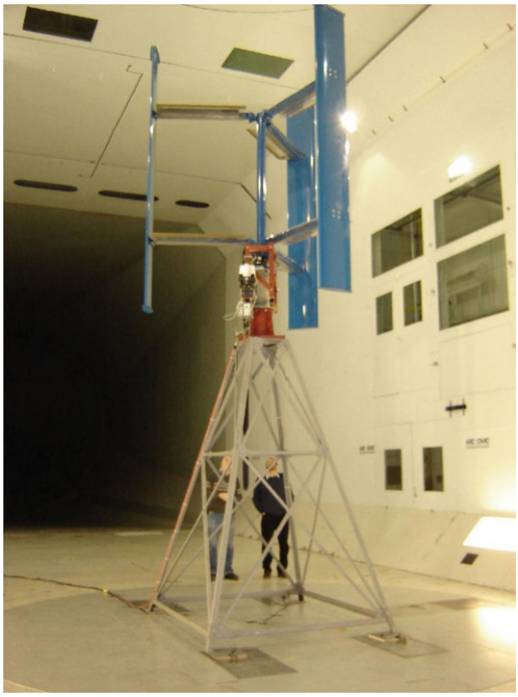


Fig. 13 Full-scale prototype of VAWT with Darrieus H-type rotor in NRC low-speed wind tunnel at McMaster University

7.1 Single-Turbine Simulation

We first compute a single VAWT and assess the resolution demands for this class of problems. The stationary domain has the outer dimensions of 50, 20, and 30 m in the stream-wise, vertical, and span-wise directions, respectively. The VAWT centerline is located 15 m from the inflow and side boundaries. The radius and height of the spinning cylinder are both 4 m.

Three meshes are used with increasing levels of refinement. The overall mesh statistics are summarized in Table 2. The finest mesh has over 17M elements. The details of the boundary-layer discretization are as follows. For Mesh 1, the size of the first element in the wall-normal direction is 0.000667 m, and 15 layers of prismatic elements were generated with a growth ratio of 1.15. For Mesh 2, the size of the first element in the wall-normal direction is 0.000470 m, and 21 layers of prismatic elements were generated with a growth ratio of 1.1. For Mesh 3, the size of the first element in the wall-normal direction is 0.000333 m, and 30 layers of prismatic elements were generated with a growth ratio of 1.05. Figure 14 shows a 2D slice of Mesh 2, focusing on the boundary-layer discretization of the blade.

Time history of the computed aerodynamic torque is plotted in Fig. 15 together with the experimental value reported for these operating conditions. Only the mean value of the torque was reported in [211, 212]. Note that after a couple of cycles a nearly periodic solution is attained. Mesh 1 predicts the average torque of about

Table 2 Statistics of the finite element meshes of the VAWT

	Number of nodes	Number of elements
Mesh 1	1,143,609	4,064,358
Mesh 2	2,478,993	7,324,964
Mesh 3	6,401,238	17,434,372

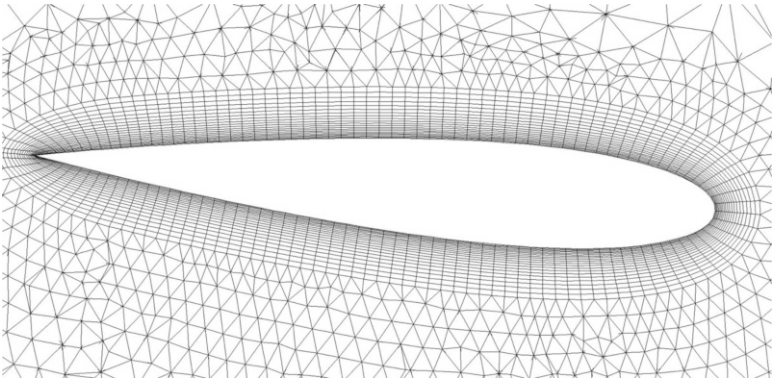


Fig. 14 2D slice of Mesh 2, focusing on the blade boundary-layer discretization

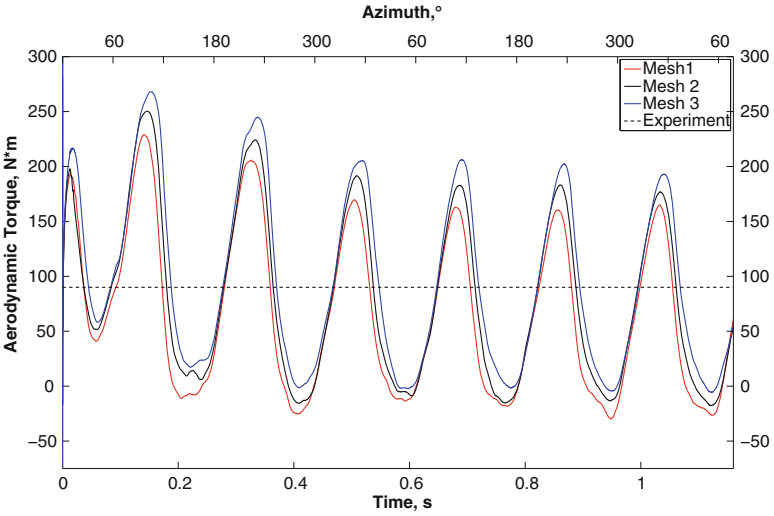


Fig. 15 Time history of the aerodynamic torque for the three meshes used. The experimental result is plotted for comparison

52 Nm, Mesh 2 gives the average torque of about 70 Nm, and Mesh 3 predicts the average torque of about 80 Nm, while the targeted experimental value is about 90 Nm. Looking further at the curves we observe that the largest differences between the predicted values of the torque between the meshes occur at the maxima and minima of the curves. Also note that the torque fluctuation during the cycle is nearly

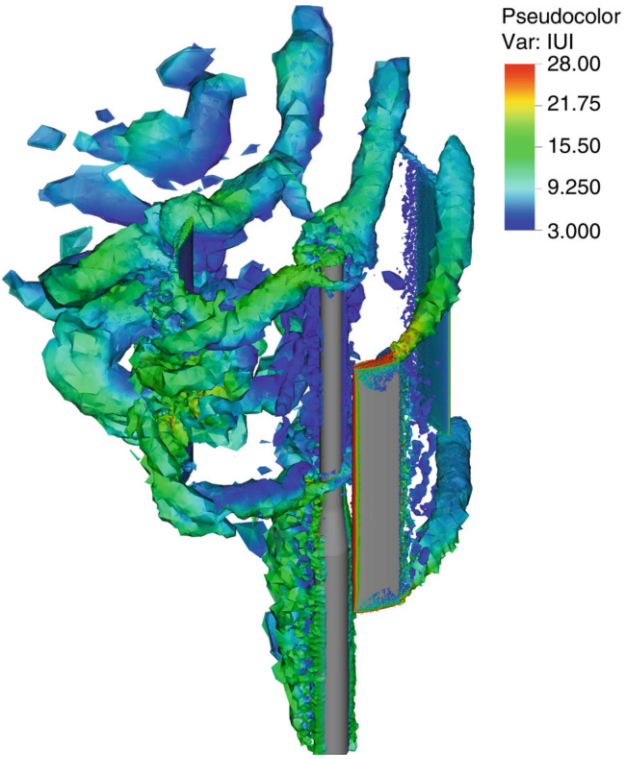


Fig. 16 Vorticity isosurfaces at a time instant colored by velocity magnitude

200 Nm, which is over twice the average. One way to mitigate such high torque variations is to allow variable rotor speed.

Figure 16 shows a snapshot of vorticity colored by flow speed. The upstream blade generates tip vortices near its top and bottom sections. Note that no large vortices are present in the middle section of the blade. There, as the flow separates on the airfoil surface, larger vortices immediately break up into fine-grained trailing-edge turbulence. The tip vortex and trailing-edge turbulence are then convected with the ambient wind velocity, and impact the tower, as well as the blade that happens to be in the downwind position in the spin cycle. However, as it is evident from the torque time histories shown in Fig. 15, these do not produce a major impact on the rotor loads, at least for a chosen set of wind and rotor speeds. The situation may, of course, change for a different set of operating conditions.

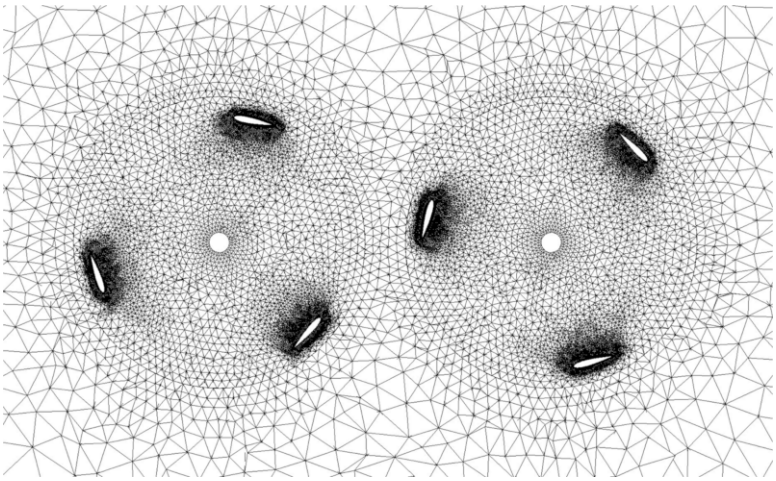


Fig. 17 Cross-section of the mesh for the two-VAWTs rotating counterclockwise

7.2 Two-Turbine Simulation

Here we investigate two counter-rotating turbines placed side-by-side in close proximity to one another. The wind and rotor speeds are the same as before, however, the turbines rotate out of phase, with the difference of 60°. The distance between the towers of the two turbines is $2.64R$, where $R = 1.25$ m is the rotor radius. This distance between the turbines falls in the range investigated in the experimental work of [213].

The stationary domain has the outer dimensions of 50, 20, and 33.3 m in the stream-wise, vertical, and span-wise directions, respectively. The centerline of each VAWT is located 15 m from the inflow and 15 m from its closest side boundary. The radius and height of the spinning cylinders are 1.45 and 4 m, respectively.

A 2D slice of the computational-domain mesh focusing on the two rotors is shown in Fig. 17. The boundary layer discretization employed for this computation is the same as that of Mesh 2 in the previous section.

Figure 18 shows the time history of the aerodynamic torque for the two-turbine case. The curve corresponding to the second turbine is shifted by 60° such that both curves are in phase. The time history of the torque for a single VAWT simulation is shown for comparison. Note that while the maxima of all curves are virtually coincident, the minima are lower for the case of multiple turbines. Also note that the multiple-turbine torque curves exhibit some fluctuation near their minima, while the single-turbine torque curve is smooth near its minima. This is likely due to the fact that the blade from one turbine, as it approaches the plane defined by the centerlines of the two towers, encounters the wake of the blade from another turbine. This, in turn, produces higher drag on that blade and results in reduction of the aerodynamic torque. A snapshot of vorticity colored by flow speed depicted in Fig. 19 shows

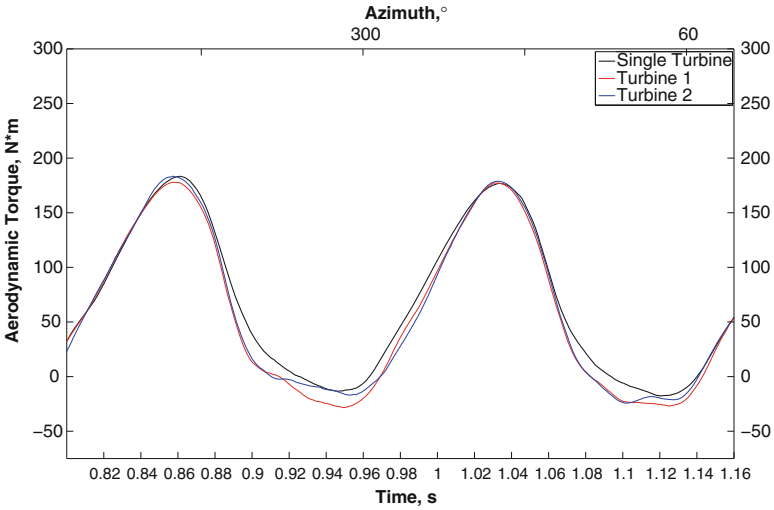


Fig. 18 Time history of the aerodynamic torque for the two VAWTs. The data for the second turbine is shifted by 60° to be in phase with the first turbine. Results from a single turbine simulation are plotted for comparison

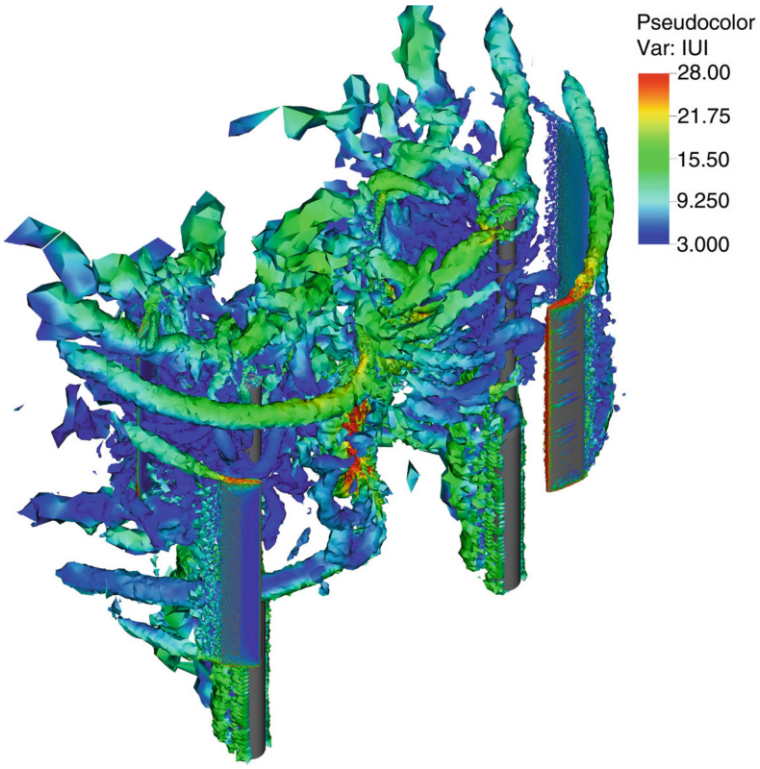


Fig. 19 Vorticity isosurfaces colored by flow speed for two VAWTs rotating counterclockwise

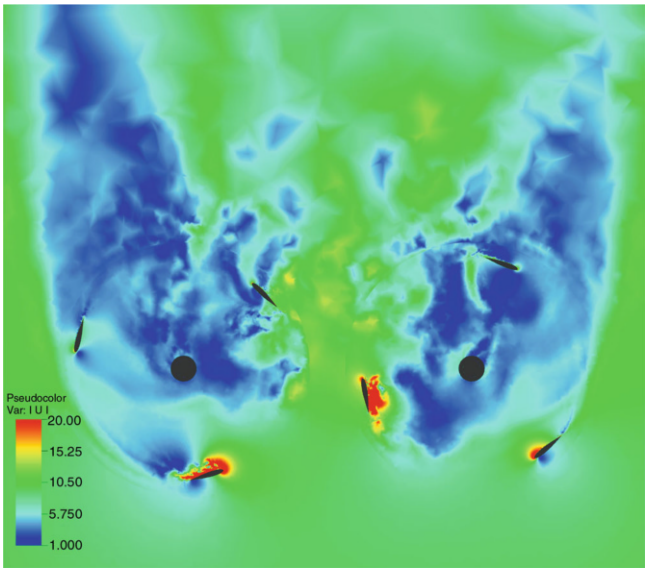


Fig. 20 Air speed at a 2D cross-section for the two VAWTs rotating counterclockwise

that the short distance between the turbines has a noticeable effect on the resulting aerodynamics. This effect may be seen more clearly in Fig. 20, which shows that one of the blades from the VAWT on the right is about to enter the turbulent region between the turbines.

8 ST Computational Flow Analysis of a VAWT

In the ST computational flow analysis of a VAWT we include from [117], we conduct test computations with 2D and 3D models. The wind turbine has four support columns at the periphery. In the 2D case we also test the ST-SUPS. In the 3D case, both the fluid–fluid SI and fluid–solid SI versions of the ST-SI are used.

Figure 21 shows the VAWT. The model is based on the wind turbine in [214]. The rotor diameter is 16 m, and the machine height is 45 m. The three blades are based on the NACA0015 airfoil, and the cord length and the blade height are 1.5 and 18 m, respectively. There are two connecting rods from the hub to each blade, and the blades are supported without any tilt with respect to the tangent of the rotation path. The four support columns are cylindrical with circular cross-section, and they provide enough strength to support the rotor, which is estimated to weigh 3 t.

We carry out the computations at a constant free-stream velocity U_∞ and with prescribed rotor motion at constant angular velocity. The rotation is clockwise viewed from the top. The air density and kinematic viscosity are 1.205 kg/m^3 and 1.511×10^{-5} m^2/s. We extract from the computations the instantaneous power coefficient C_{POW}, defined as

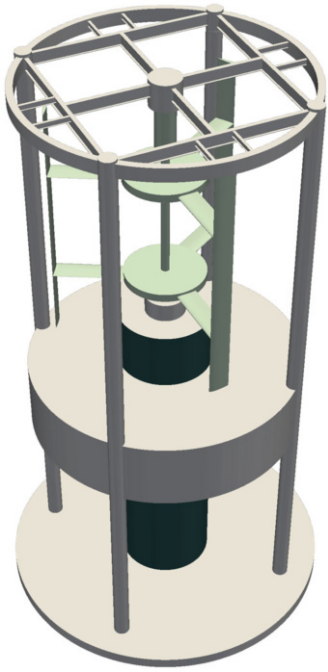


Fig. 21 ST computational flow analysis of a VAWT. The model

$$C_{\text{POW}} = \frac{P}{\frac{1}{2}\rho U_\infty^3 A}, \tag{68}$$

where P and A are the generated power and the projected area of the wind turbine. We report the power coefficient as a function of the blade orientation as represented by the angle ϕ seen in Fig. 22. With that orientation, the flow speed seen by a blade can be calculated as

$$V = U_\infty \sqrt{1 - 2\lambda \sin\phi + \lambda^2}, \tag{69}$$

where λ is the tip-speed ratio (TSR).

Based on that speed, we define the pressure coefficient as

$$C_p \equiv \frac{p - p_\infty}{\frac{1}{2}\rho V^2} \tag{70}$$

$$\approx 1 + \frac{p - p_{\max}}{\frac{1}{2}\rho V^2}, \tag{71}$$

where p_∞ is the free-stream pressure, and $p_{\max}$ is the maximum pressure on the blade surface, which is essentially the stagnation pressure.

The computational-domain size is 62.5 times the rotor diameter in the wind direction, with a distance of 18.75 times the rotor diameter between the upstream

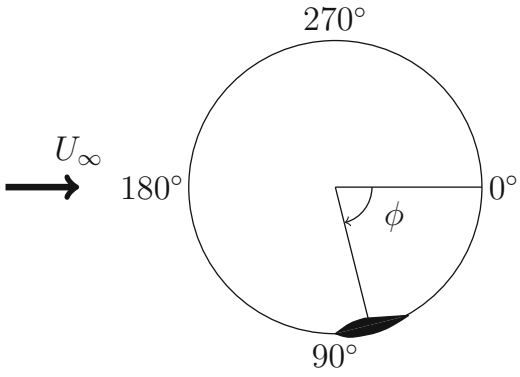


Fig. 22 ST computational flow analysis of a VAWT. Blade orientation as represented by ϕ

boundary and the center of the rotor. In the cross-wind direction, the domain size is 37.5 times the rotor diameter. In the 3D case, the domain height is 10 times the rotor diameter.

The mesh position is represented by quadratic NURBS in time. There are three patches that are 120° each, and the secondary mapping introduced in [105, 114, 118] is used to achieve the constant angular velocity. In the computations reported here, a patch has 120 time steps, resulting in 1.0° per time step. We report results for a free-stream velocity of 12.56 m/s and a TSR value of 6. This results in a time-step size of 0.001852 s.

8.1 2D Computations

Figure 23 shows the mesh used in the computations. There are 84,565 nodes and 83,304 quadrilateral elements. Figure 24 shows the mesh around a blade and also the SI. The first element thickness is about 0.9 mm. There are a total of 18 layers, each increasing in thickness with a progression ratio of about 1.1.

At the inflow and outflow boundaries we have free-stream velocity and traction-free conditions, respectively. At the other lateral boundaries we have slip conditions. The velocity at the rotor surface is prescribed as a strong Dirichlet boundary condition. The prescribed velocity is evaluated at the temporal integration points with the values extracted from the NURBS representation of the rotor surface velocity.

We compute six cycles, the first cycle with the ST-SUPS only, and the second to sixth cycles with both the ST-SUPS and ST-VMS. The number of nonlinear iterations per time step is 4, and the number of GMRES iterations per nonlinear iteration is 200. In the ST-VMS computation, the first two nonlinear iterations are based on the ST-SUPS, and the last two iterations are based on the ST-VMS. The stabilization parameters are from Eqs. (2.4)–(2.8) in [117]. In the case of the ST-SUPS, we exclude "τ_{SUGN4}" from Eq. (2.4) in [117].

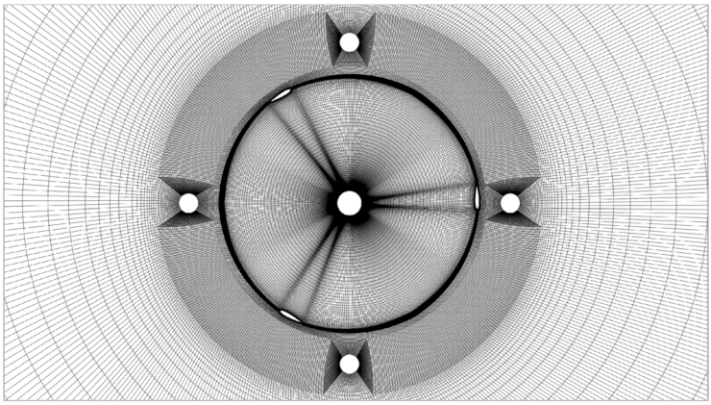


Fig. 23 ST computational flow analysis of a VAWT. 2D computation. Mesh around the turbine

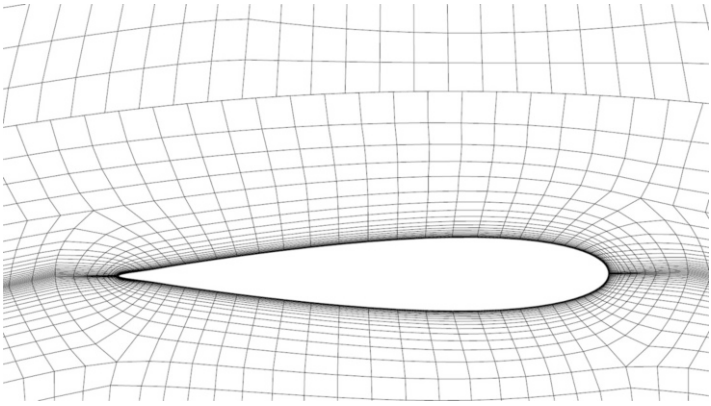


Fig. 24 ST computational flow analysis of a VAWT. 2D computation. Mesh around a blade and the SI

We report results averaged over the third to sixth cycles. Figures 25 and 26 show the pressure coefficient at four different angles from computations with the ST-SUPS and ST-VMS, respectively. Figure 27 shows the instantaneous power coefficient. The power coefficient averaged over full rotation is -2.0 for the ST-SUPS, and 0.24 for the ST-VMS.

Remark 20 The ST-SUPS solution is clearly not accurate, producing no power. The reason can be understood by comparing Figs. 25 and 26. In the ST-SUPS case, the flow on the suction side separates earlier than it does in the ST-VMS case.

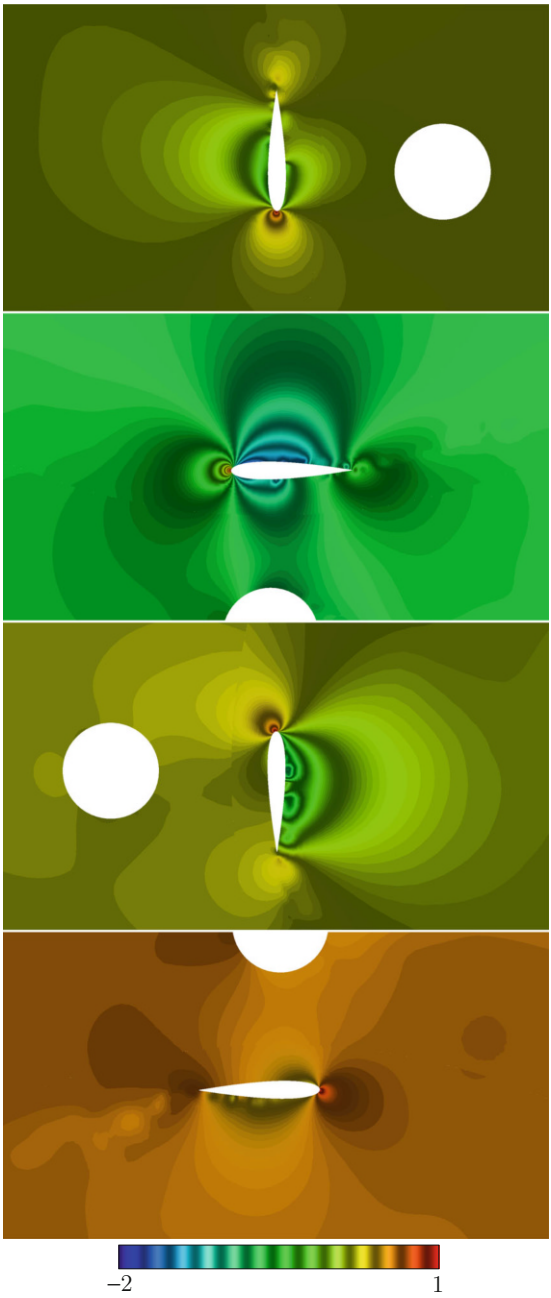


Fig. 25 ST computational flow analysis of a VAWT. 2D computation. Pressure coefficient from computation with the ST-SUPS, at $\phi = 0°$, 90°, 180°, and 270°

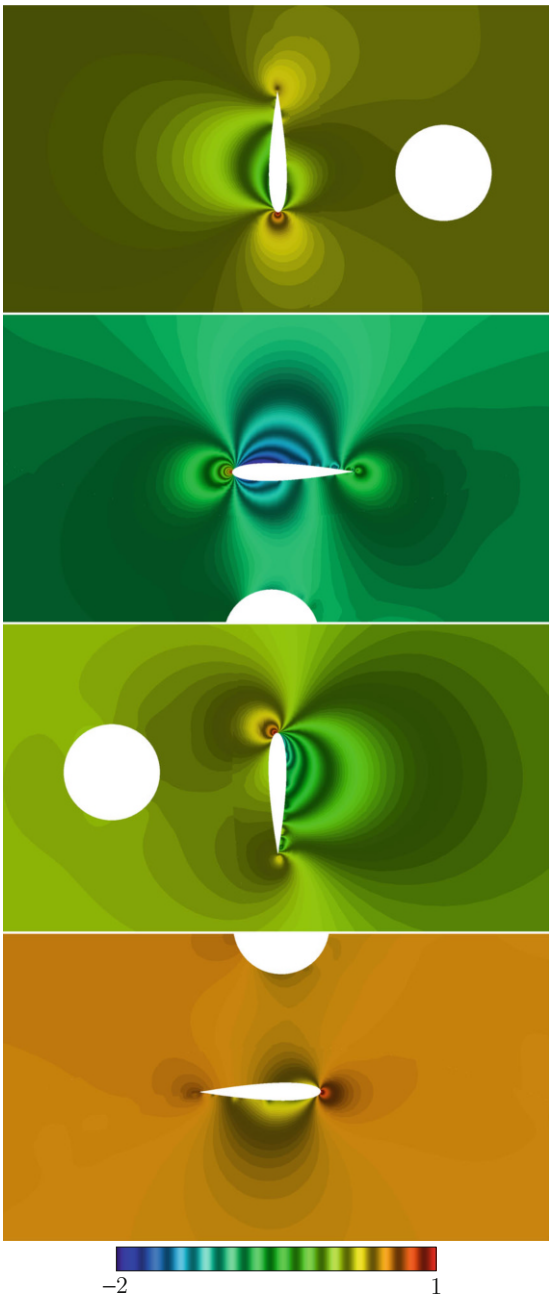


Fig. 26 ST computational flow analysis of a VAWT. 2D computation. Pressure coefficient from computation with the ST-VMS, at $\phi = 0°$, 90°, 180°, and 270°

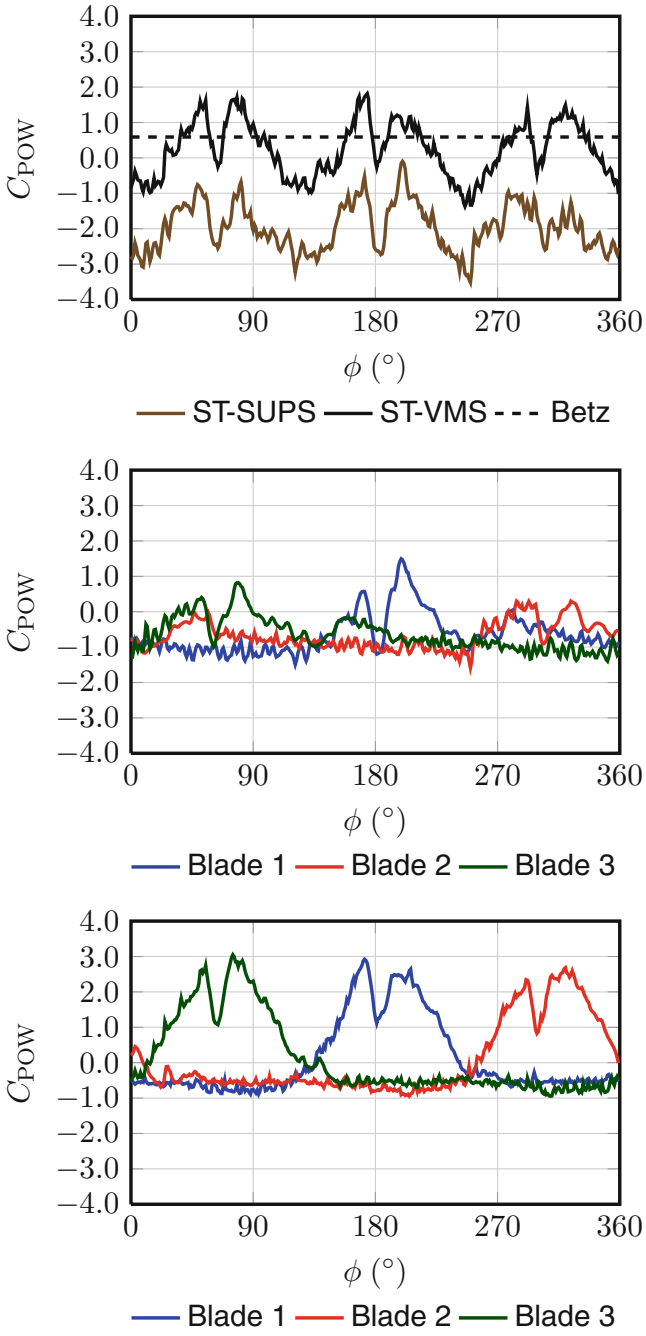


Fig. 27 ST computational flow analysis of a VAWT. 2D computation. Instantaneous power coefficient as a function of ϕ defined with respect to Blade 1. The total power coefficient (*top*) and the power coefficient for the individual blades from the ST-SUPS (*middle*) and ST-VMS (*bottom*) computations

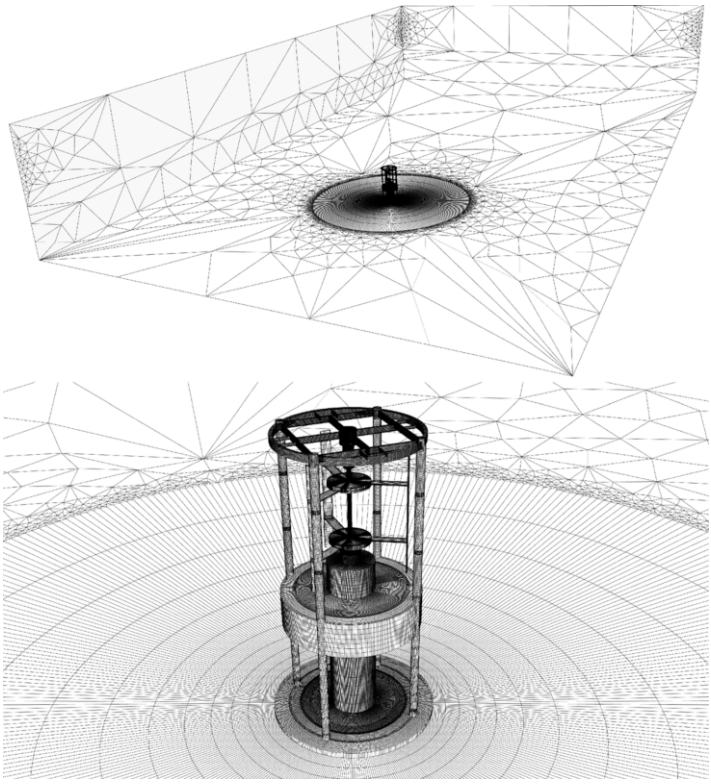


Fig. 28 ST computational flow analysis of a VAWT. Mesh for the 3D computation. Entire domain (*top*) and around the turbine (*bottom*)

8.2 3D Computation

Here we report results only from the ST-VMS computation. Figure 28 shows the mesh used in the computation. There are 2,822,109 nodes and 6,875,420 mixed elements (hexahedron, pyramid, and tetrahedron). Figure 29 shows the mesh around a blade and the SI. There are a total of 9 uniform-thickness layers with an element thickness of about 6 mm.

At the inflow and outflow boundaries we have free-stream velocity and traction-free conditions, respectively. At all other domain boundaries we have slip conditions. We do not try to resolve the boundary layer near the ground since the blades are positioned relatively high (see Fig. 21). The fluid velocity at the rotor surface is prescribed as a weakly imposed Dirichlet boundary condition. As in the 2D case, the prescribed velocity is evaluated at the integration points with the values extracted from the NURBS representation of the rotor surface velocity.

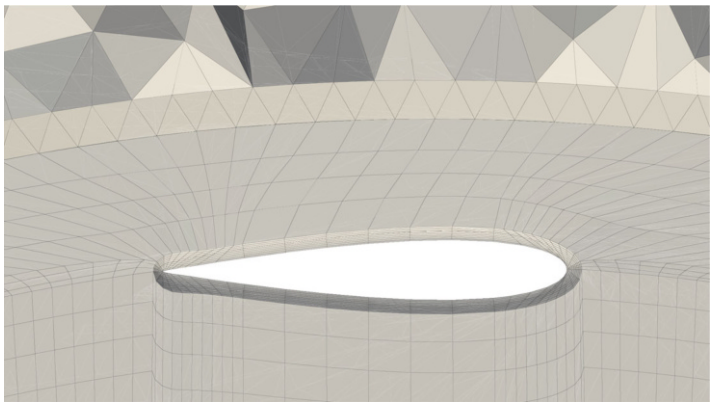


Fig. 29 ST computational flow analysis of a VAWT. A cut plane of the 3D mesh around a blade and the SI, which is between the hexahedral and tetrahedral elements seen in the figure

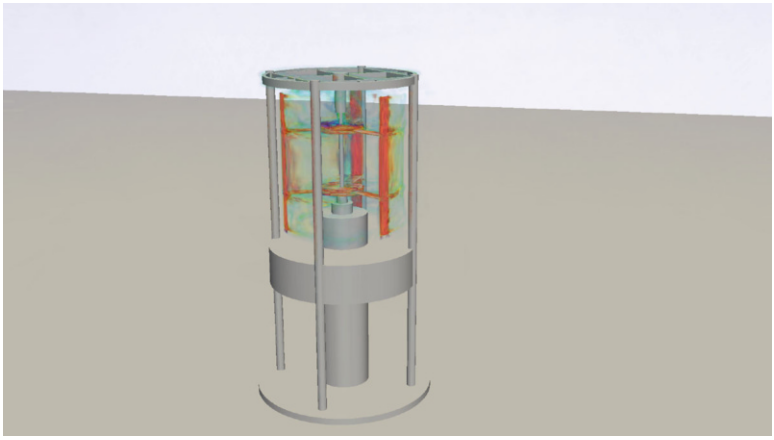


Fig. 30 ST computational flow analysis of a VAWT. 3D computation. Volume rendering of the vorticity magnitude at $\phi = 270°$. The flow is from left to right

We compute two cycles. The number of nonlinear iterations per time step is 4, and the number of GMRES iterations per nonlinear iteration is 400. The first three nonlinear iterations are based on the ST-SUPS, and the last iteration is based on the ST-VMS. The stabilization parameters are from Eqs. (2.4)–(2.8) in [117], and we use the symmetric version of τ_{SUGN4}, defined in Remark 3 in [117]. We report results from the second cycle. Figure 30 shows the vorticity at $\phi = 270°$. Figure 31 shows the pressure coefficient at four different angles. Figure 32 shows the instantaneous power coefficient. The power coefficient averaged over full rotation is about 0.54.

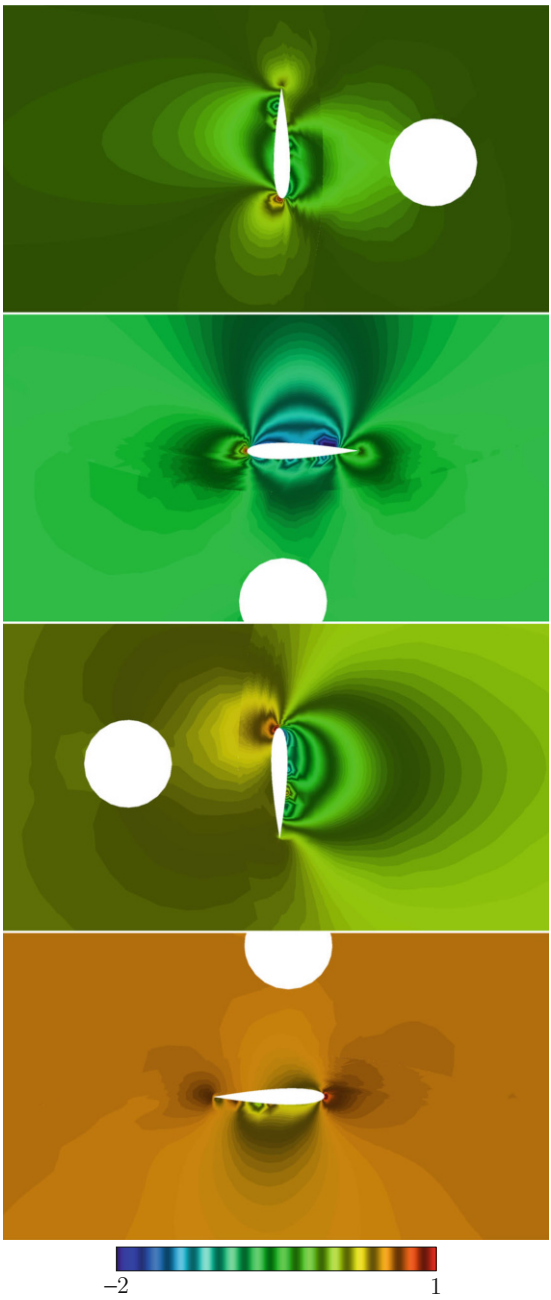


Fig. 31 ST computational flow analysis of a VAWT. 3D computation. Pressure coefficient on a cut plane at $\phi = 0°, 90°, 180°$, and $270°$

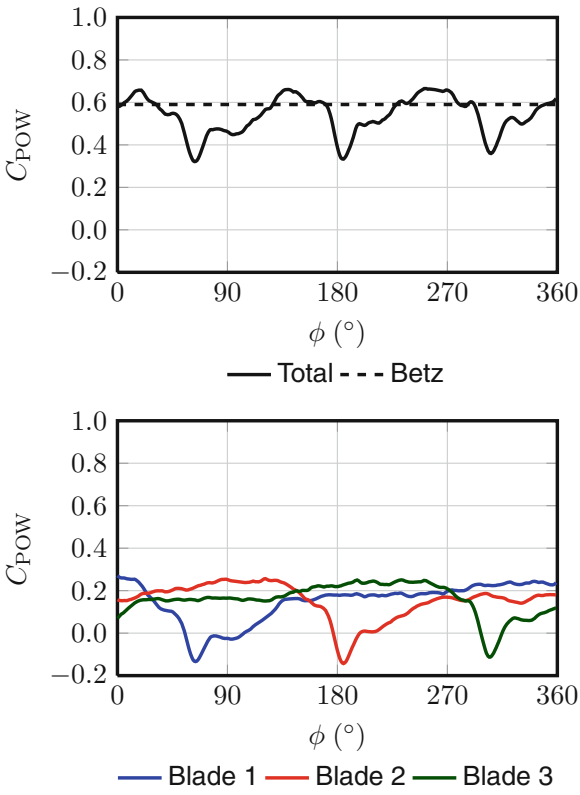


Fig. 32 ST computational flow analysis of a VAWT. 3D computation. Instantaneous power coefficient as a function of ϕ defined with respect to Blade 1. The total power coefficient (*top*) and the power coefficient for the individual blades (*botom*)

9 FSI Modeling of Single Wind Turbines

9.1 Simulation of a 5 MW HAWT Yawing Motion

Here we present an FSI simulation of a 5 MW offshore wind turbine undergoing yawing motion using the new mesh update technique described in Sect. 5.2. The material in this section is taken from [67]. The wind turbine is equipped with a 61 m blade, which is the scaled version of a 100 m baseline blade designed by Sandia National Laboratory for large offshore wind turbines (referred to as SNL 100–00 blade). The blade geometry was described in Sect. 6. The blade has the trailing and leading edge reinforcements together with the root build-up. Three shear webs are placed to minimize the length of the unsupported panel (see Fig. 33 for details). The blade laminate has six principal regions: root, spar cap, trailing edge reinforcement, leading edge panels, aft panels, and shear webs.

Tables 3 and 4 list the materials used in the blade design. The root buildup is composed of triaxial material (SNL Triax), and the whole internal and external blade surfaces have a 5 mm layer of this material. As the root buildup tapers down in thickness, the spar cap increases in thickness. The maximum thickness of the spar

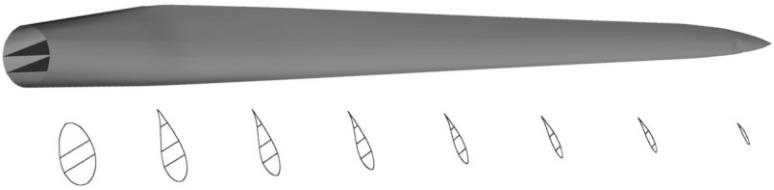


Fig. 33 SNL 100–00 blade shell model with several cross-section cuts to show the arrangement of the three shear webs

Table 3 Orthotropic materials used in the SNL 100–00 blade

Properties	E-LT-5500/ EP-3	Saertex/ EP-3	SNL triax
E_1 (GPa)	41.8	13.6	27.7
E_2 (GPa)	14	13.3	13.65
G_{12} (GPa)	2.63	11.8	7.2
ν_{12}	0.28	0.51	0.39
ρ_2 (kg/m^3)	1920	1780	1850
Lay-up	$[0]_2$	$[\pm 45]_4$	$[\pm 45]_4[0]_2$

Table 4 Isotropic materials used in the SNL 100–00 blade

Properties	Foam	Resin	Gel coat
E_1 (GPa)	0.256	3.5	3.44
E_2 (GPa)	0.256	3.5	3.44
G_{12} (GPa)	0.022	1.4	1.38
ν_{12}	0.3	0.3	0.3
ρ_2 (kg/m^3)	200	1100	1235

cap is 136 mm at maximum chord (19.5%), while the minimum thickness of the spar cap is 5 mm, starting at 94.4% of the blade span and continuing almost all the way to the tip. The trailing edge is reinforced with uniaxial laminate E-LT-5500/EP-3 and foam materials. The trailing edge reinforcement has a constant width of 1.0 m that continues until 94.4% span, and then tapers to the tip. To improve buckling resistance and minimize the weight, foam is also chosen as the core material for the leading panel and aft panels. Longitudinal fibers of E-LT-5500/EP-3 are placed on the spar cap to improve the flapwise bending stiffness. The spar cap has a constant width of 1.5 m. As a result, the two principal shear webs, which begin at 2.4 m and terminate at 94.4 m, are positioned 0.75 m before and after the pitch axis. The third shear web starts at 14.6 m and terminates at 60.2 m, and is positioned at 78% chord at its starting location and 68% chord at its terminal location. A combination of foam and Saertex/EP-3 is used in shear webs to enhance the shear stiffness. An extra 5 mm of epoxy resin is included in the internal blade surface, and the external surface includes 0.6 mm of gelcoat. The same layup is employed for both low- and high-pressure blade surfaces. Structural-mechanics validation of this blade was presented in [67].

The wind turbine rotor is positioned at 80 m above ground and is tilted by 5° to avoid the blade hitting the tower as the rotor spins. (Another way to have sufficient

tower clearance is to "prebend" the rotor blades into the wind. See [215] for details.) Furthermore, the wind turbine rotor plane is initially placed at 15° relative to the wind direction. A fixed yawing rotational speed is applied to the gearbox to slowly turn the rotor into the wind at 0.03 rad/s. The inflow wind speed is set to 11.4 m/s. The initial rotor speed is set to 12.1 rpm, and the rotor is allowed to spin freely during the prescribed yawing motion.

The structural mechanics mesh of the full turbine has 13,273 quadratic NURBS shell elements and two quadratic NURBS beam elements. The aerodynamics mesh has a total of 5,458,185 linear elements. Triangular prisms are employed in the blade boundary layers, and tetrahedral elements are used elsewhere in the aerodynamics domain. The size of the first boundary-layer element in the wall-normal direction is 1 cm. The time step of 0.0001 s is employed in the computation.

Snapshots of the structure deformed configuration are shown in Fig. 34, while isosurfaces of vorticity colored by flow speed are shown in Fig. 35.

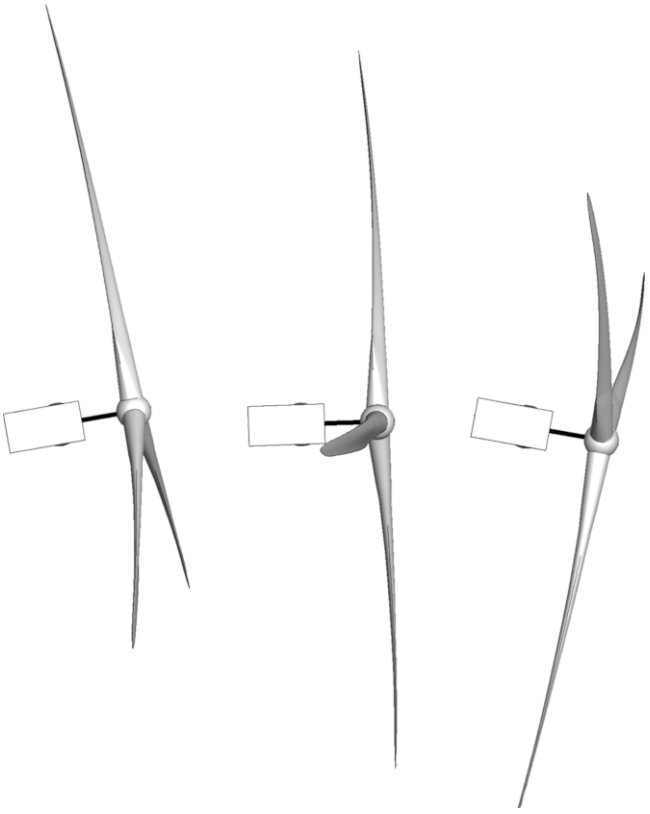


Fig. 34 FSI simulation of a HAWT undergoing a yawing motion. Snapshots of the top view of the wind-turbine structure current configuration. The blades appear to be quite flexible and care needs to be taken when designing the rotating subdomain to avoid the flexing blade crossing its boundary

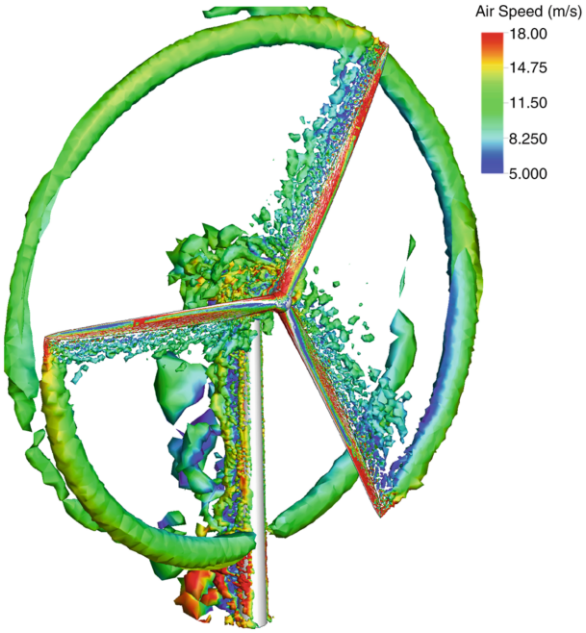


Fig. 35 FSI simulation of a HAWT undergoing a yawing motion. Snapshots of vorticity colored by air speed illustrating the air flow complexity

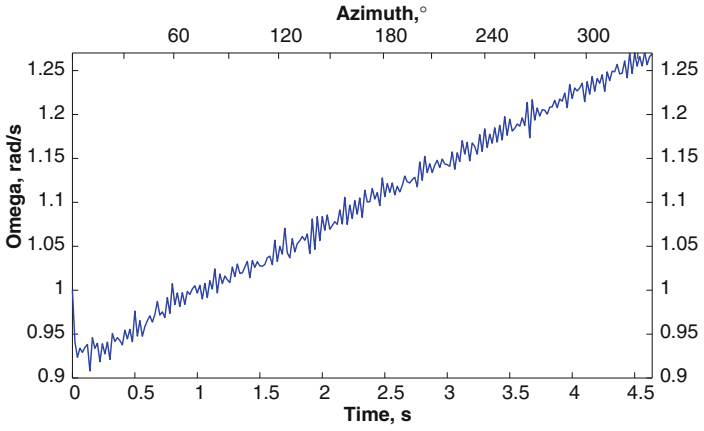


Fig. 36 FSI simulation of a HAWT undergoing a yawing motion. Time history of the angular speed

Figures 36 and 37 show the time history of the axial component of the aerodynamic torque and angular speed (i.e., the component in the direction of the vector $\mathbf{n}_{\text{rot}}$ in Eq. (63)). Both are slowly increasing as the rotor turns into the wind, as expected. The level of the computed aerodynamic torque is consistent with the earlier simulations for this wind turbine operating under similar wind- and rotor-speed conditions (see, e.g., [8, 9, 17, 63, 113, 114, 143]).

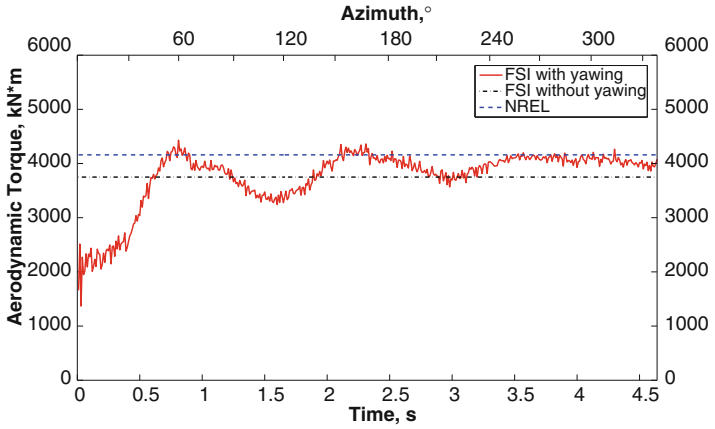


Fig. 37 FSI simulation of a HAWT undergoing a yawing motion. Time history of the rotor aerodynamic torque

9.2 Simulation of the Windspire VAWT Start-Up Conditions

The FSI computations presented here are performed for a 1.2 kW Windspire design [216], a three-bladed Darrieus VAWT. The material in this section is taken from [69]. The total height of the VAWT tower is 9.0 m, and the rotor height is 6.0 m. The rotor uses the DU06W200 airfoil profile with the chord length of 0.127 m, and is of the Giromill type with straight vertical blade sections attached to the main shaft with horizontal struts.

The outer aerodynamics computational domain has the dimensions of 50, 20, and 30 m in the stream-wise, vertical, and span-wise directions, respectively. The VAWT centerline is located 15 m from the inflow and side boundaries. The aerodynamics mesh has about 8 M elements, which are linear triangular prisms in the blade boundary layers, and linear tetrahedra elsewhere. The boundary layer mesh is constructed using 18 layers of elements, with the size of the first element in the wall-normal direction of 0.0003 m, and growth ratio of 1.1. The aerodynamic validation for this model may be found in [69].

The structural model is shown in Fig. 38. The rotor and struts are made of aluminum, and the tower is made of steel. Quadratic NURBS are employed for both the beam and shell discretizations. The total number of beam elements is 116, and total number of shell elements is 7029.

As a part of the FSI simulations, we present a preliminary investigation of the start-up issues in VAWTs. We fix the inflow wind speed at 11.4 m/s, and consider three initial rotor speeds: 0, 4, and 12 rad/s. Of interest is the transient response of the system. In particular, we focus on how the rotor angular speed responds to the prescribed initial conditions, and what is the range of the tower tip displacement during the VAWT operation. The VAWT is allowed to spin freely and accelerate under the action of the ambient wind. The time step in the computations is set to 2.0×10^{-5} s.

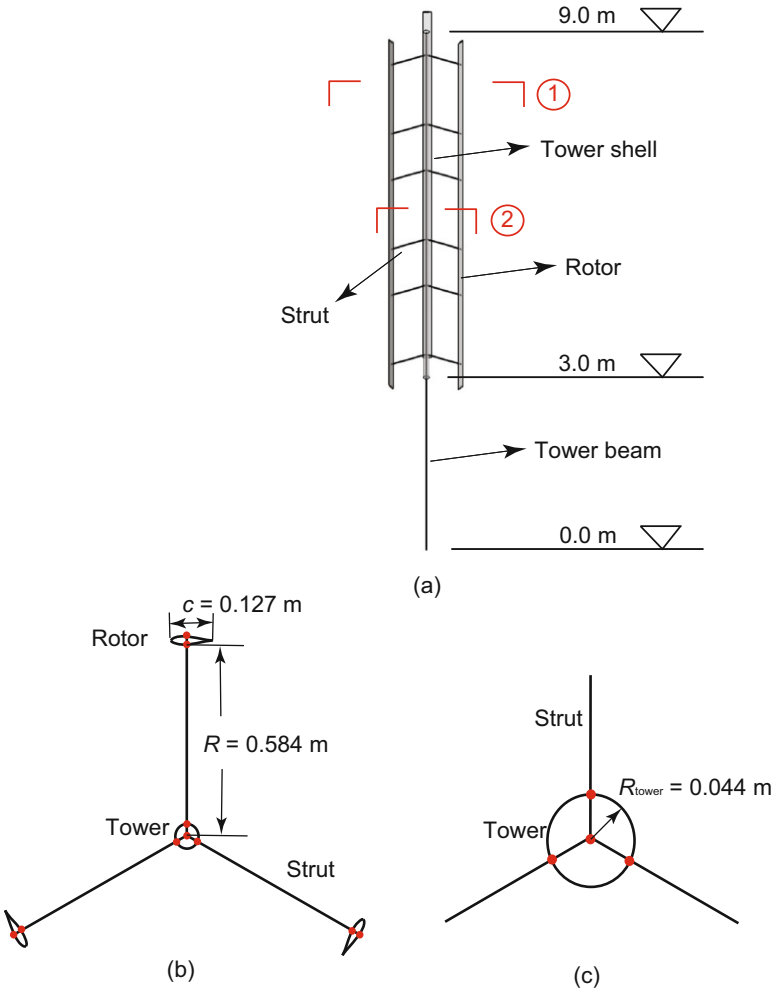


Fig. 38 Windspire VAWT structural model with dimensions included: (**a**) Full model using isogeometric NURBS-based rotation-free shells and beams; (**b**) Model cross-section 1 showing attachment of the struts to the blades and tower shell; (**c**) Model cross-section 2 showing attachment of the struts and tower shell

The mesh update technique described in Sect. 5.2 is applied to this case in a straightforward fashion. The radius and height of the inner cylindrical domain that encloses the rotor are 1.6 and 7 m, respectively. That is, the cylindrical domain extends 0.5 m above and below the rotor blades. The rotor axis direction $\mathbf{n}_{\text{rot}}$ is defined according to Eq. (63), where the points $\mathbf{x}_{\text{ori}}$ and $\mathbf{x}_{\text{tip}}$ are located at the bottom and top intersections of the tower beam and shell, respectively. The instantaneous rotor angular velocity is computed from Eq. (56), the spinning component is

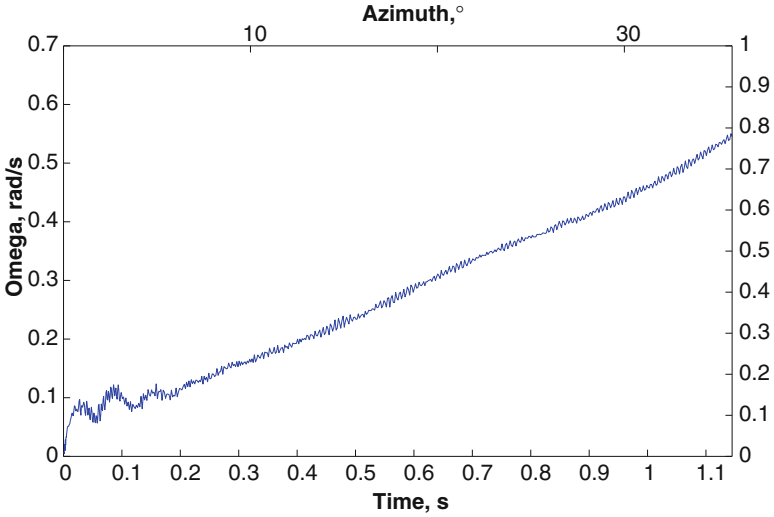


Fig. 39 Time history of the rotor speed starting from 0 rad/s

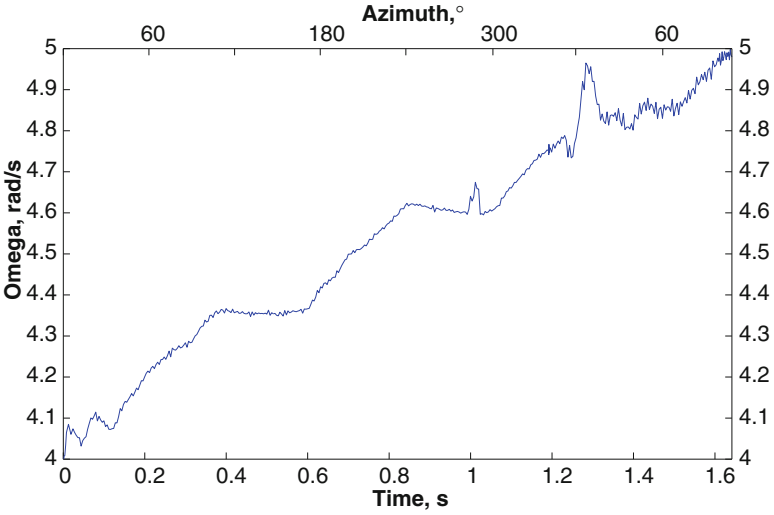


Fig. 40 Time history of the rotor speed starting from 4 rad/s

removed as per Eq. (62), and the two angular velocities are used to update the sliding-interface mesh positions.

Time history of the rotor speed is shown in Figs. 39, 40 and 41. For the 0 rad/s case the rotor speed begins to increase suggesting this configuration is favorable for self-starting. For the 4 rad/s case, the rotor speed has a nearly linear acceleration region followed by a *plateau* region. In [217] the plateau region is defined as the

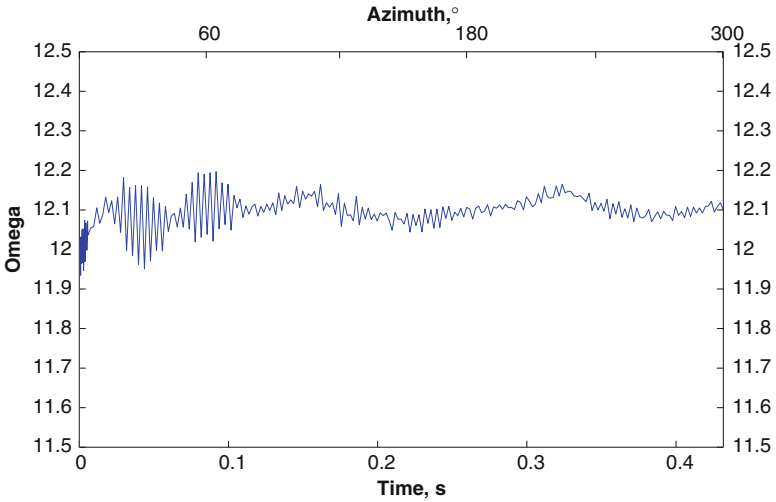


Fig. 41 Time history of the rotor speed starting from 12 rad/s

regime when the turbine operates at nearly constant (i.e., steady-state like) rotational speed. From the angular position of the blades in Fig. 40 it is evident that the plateau region occurs approximately every 120° when one of the blades is in a stalled position. It lasts until the blade clears the stalled region, and the lift forces are sufficiently high for the rotational speed to start increasing again. As the rotational speed increases, the angular velocity is starting to exhibit local unsteady behavior in the plateau region. While the overall growth of the angular velocity for the 4 rad/s case is promising for the VAWT to self start, the situation is different for the 12 rad/s case (see Fig. 41). Here the rotor speed has little dependence on the angular position and stays nearly constant, close to its initial value. It is not likely that the rotor speed will reach to the operational levels in these conditions without an applied external torque, or a sudden change in wind speed, which is consistent with the findings of [218].

Figure 42 shows, for a full turbine, a snapshot of vorticity colored by flow speed for the 4 rad/s case. The figures indicate the complexity of the underlying flow phenomena and the associated computational challenges. Note the presence of quasi-2D vortex tubes that are created due to massive flow separation, and that quickly disintegrate and turn into fine-grained 3D turbulence further downstream.

Figure 43 shows the turbine current configuration at two time instances during the cycle for the 4 rad/s case. The displacement is mostly in the direction of the wind, however, lateral tower displacements are also observed as a result of the rotor spinning motion. The displacement amplitude is around 0.10–0.12 m, which we find reasonable given the tower height of 9 m, and one of the VAWT design objectives being that the structure is not too flexible.

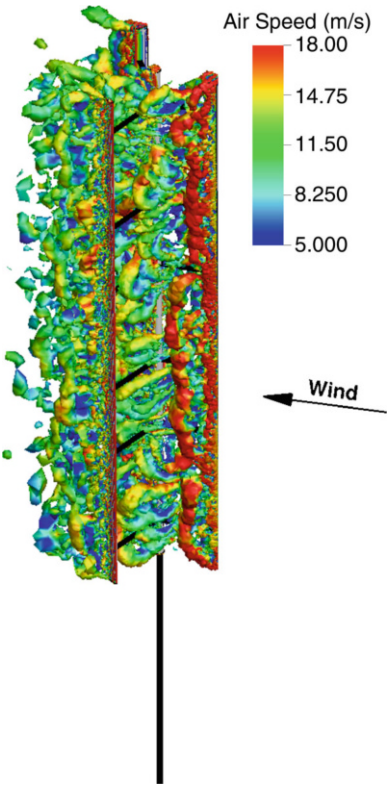


Fig. 42 Vorticity isosurfaces at a time instant colored by velocity magnitude for the 4 rad/s case

10 Two Back-to-Back Wind Turbines in Turbulent ABL Flow

In this section, the techniques described in Sect. 2 are applied to the simulation of two back-to-back NREL 5 MW wind turbines [210] operating in ABL flow. This turbine design was previously simulated under uniform flow conditions in [8, 9, 17, 63, 67, 113, 114, 143] and aerodynamic rotor-only simulation in stably stratified ABL was presented earlier in Sect. 6. Each turbine has a rotor with 61 m blades mounted on an 80 m tower and operating at constant, fixed rotor speed of 9 rpm. This rotor speed gives the optimal tip-speed ratio for 8 m/s wind [210], which is also the geostrophic wind speed used in the present computations. The material presented in this section is taken from [61].

10.1 Computational Set-up and Boundary Conditions

Two wind turbines are positioned one behind the other at a distance of 480 m, which corresponds to four rotor diameters. As a result, the wake generated by the upstream

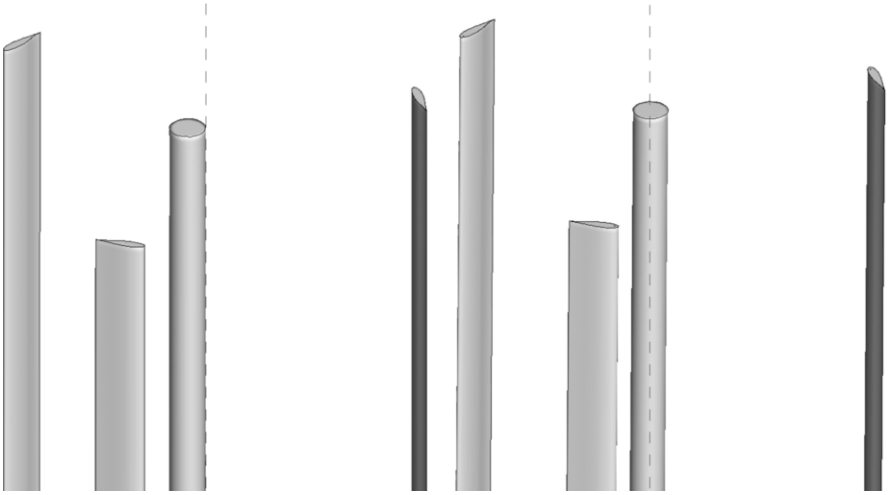


Fig. 43 Turbine current configuration at two time instances for the 4 rad/s case. The tower centerline in the reference configuration is shown using the dashed line to illustrate the range of turbine motion during the cycle. The range of the tower tip displacement during the cycle is about 0.10–0.12 m

turbine needs to be accurately computed over a long domain before it impacts the downstream turbine, which poses a significant computational challenge due to a very large problem size. To circumvent this difficulty, a multi-domain method (MDM), originally proposed in [146], is adopted in the present work to efficiently separate the two turbine domains. We note that the MDM was successfully applied to a number of challenging 3D problems, including flow around a small wing placed in the wake of a larger wing [146], flow in the wake of a circular cylinder up to 300 diameters downstream [219, 220], and aerodynamics and FSI of a parachute crossing the far wake of an aircraft [221, 222]. More recently, a spatially multiscale version of the MDM was applied to thermo-fluid analysis of a ground vehicle and its tires in [93].

In the present work the MDM is employed as follows. The problem domain is divided into three subdomains (see Fig. 44 for dimensions and notation). Domains labeled *Turbine 1* and *Turbine 2* contain the upstream and downstream turbines, respectively, and domain labeled Box contains the space between the turbines. The three domains are simulated in a sequential manner. Velocity and temperature boundary conditions on the inflow boundary of *Turbine 1*, as well as lateral boundaries of all subdomains, are obtained from a standalone 3D LES computation of a stratified ABL with a uniform grid size of 5 m. This stratified flow computational model [207, 223, 224], which can be run in DNS or LES modes, makes use of a mixed spectral/finite-difference algorithm and a subgrid model based on dynamic eddy viscosity and diffusivity. Nodal values of the velocity and temperature boundary conditions are obtained by interpolating the finite-difference solution from

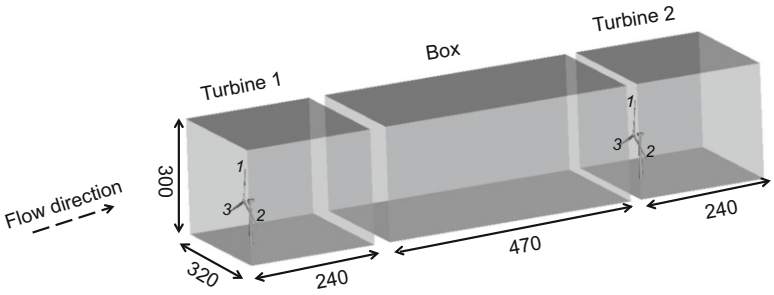


Fig. 44 Problem set-up for the multi-domain wind-turbine simulation. The subdomains are labeled and all the dimensions are in m

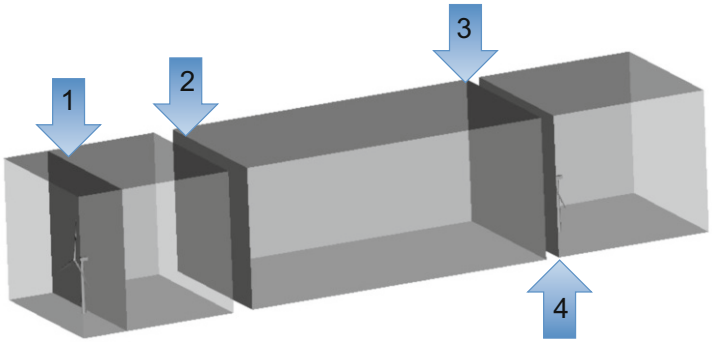


Fig. 45 Data flow between the subdomains. Velocity and temperature collected at location 1 are applied as inlet boundary conditions at location 2. Velocity and temperature collected at location 3 are applied as inlet boundary conditions at location 4. At all lateral boundaries velocity and temperature boundary conditions come from a standalone spectral/fine-difference LES

the structured grid of the LES simulation to the unstructured grids of the wind-turbine simulations. This data transfer strategy, employing the same dataset as in the present work, was successfully tested for the rotor-only ABL simulation in [59]. The background temperature $\bar{\phi}(x_3)$ is set to 260 K up to 100 m with an overlying inversion of strength 0.01 Km^{-1} for all domains. The geostrophic wind speed is set to 8 m/s, and the Coriolis parameter to $f_c = 1.39 \times 10^{-4}$. Velocity and temperature inflow boundary conditions for *Box* are obtained using a similar data transfer strategy, where, in this case, the data is obtained by interpolating the solution on a plane positioned 10 m behind the turbine during pure aerodynamic simulation on *Turbine 1*. Inflow boundary conditions for *Turbine 2* are obtained by interpolating the solution on the outflow plane of *Box* (see Fig. 45 for details.)

Traction boundary conditions are prescribed at the outlet boundaries of all subdomains. To generate the traction values, a simulation in *Turbine 1* is performed first with the wind turbine removed, and with zero outlet traction boundary conditions. The inlet tractions produced as a result of this computation, shown in Fig. 46, are

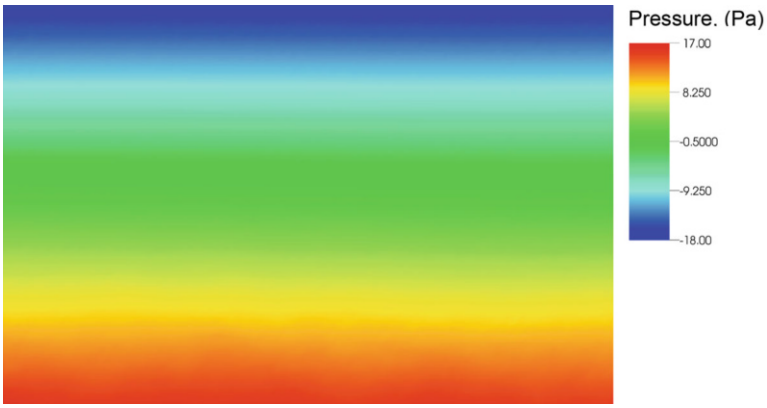


Fig. 46 Pressure profile used for outflow boundary conditions on all subdomains

then assigned as outlet boundary conditions for all subdomains. A similar strategy was successfully employed in [59], as well as in [93] to perform a detailed thermo-fluid analysis of the rear tires of a ground vehicle.

The subdomains are discretized using triangular prisms in the boundary layer region near the wind-turbine rotors, and tetrahedra elsewhere (see Fig. 47). For *Turbine 1* and *Turbine 2* the boundary-layer mesh design is based on that reported in [67]. For *Turbine 1* a total of 7,824,602 elements are used with a 4 m element length on the outer boundaries. A finer grid resolution with 2 m element length is used on a plane behind the upstream turbine where inlet data is collected for the *Box* simulation. The *Box* domain, which has a refined inner region to more accurately represent the wake turbulence, is discretized using 15,436,631 elements The *Turbine 2* domain, with a total of 9,153,426 elements, also contains a refined inner region in front of the turbine for better wake resolution. The time step size is set to 10^{-4} s for *Turbine 1* and *Turbine 2* simulations, and to 10^{-2} s for the *Box* simulation.

10.2 Aerodynamics Simulation Results

Pure aerodynamics simulation results, which are also referred as "CFD," are reported in this section. During the CFD simulations the wind turbine rotor is considered as a rigid rotating body. Figure 48 shows the velocity and temperature contours at the domain centerline. No discernible discontinuity between the subdomains is observed. A slight growth of the shear layer from the upper edge of the upstream-turbine rotor can also be seen in Fig. 48. The bottom shear layer grows much more rapidly, due to higher turbulent mixing and presence of the tower.

Figure 49 shows vorticity isosurfaces. Rotor-tip vortices of the upstream turbine maintain a helical pattern for a distance of about one rotor diameter. They later break

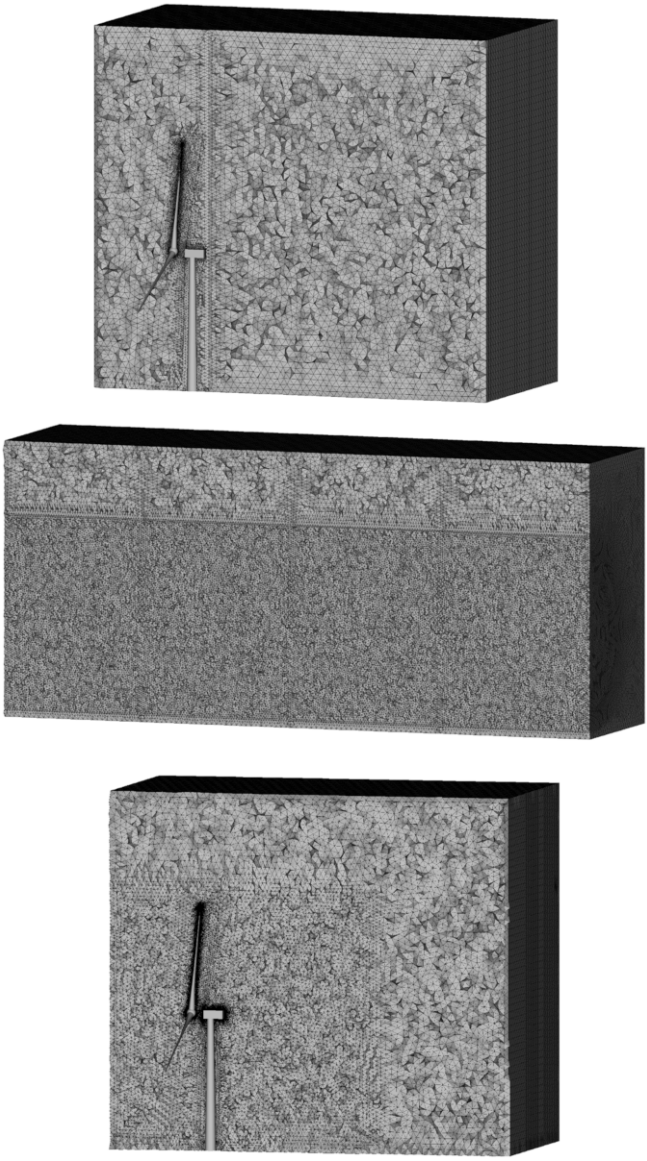


Fig. 47 Meshes employed in the wind-turbine aerodynamics and FSI simulations. Top-to-bottom: *Turbine 1*, *Box*, *Turbine 2*

up, and eventually merge with vortices shed from the root and tower to form larger structures at a distance between two and three rotor diameters (see Fig. 49.) These larger flow structures impact the downstream-turbine rotor and tower, and break up together with the rotor-tip vortices. The helical pattern of the rotor-tip vortices for

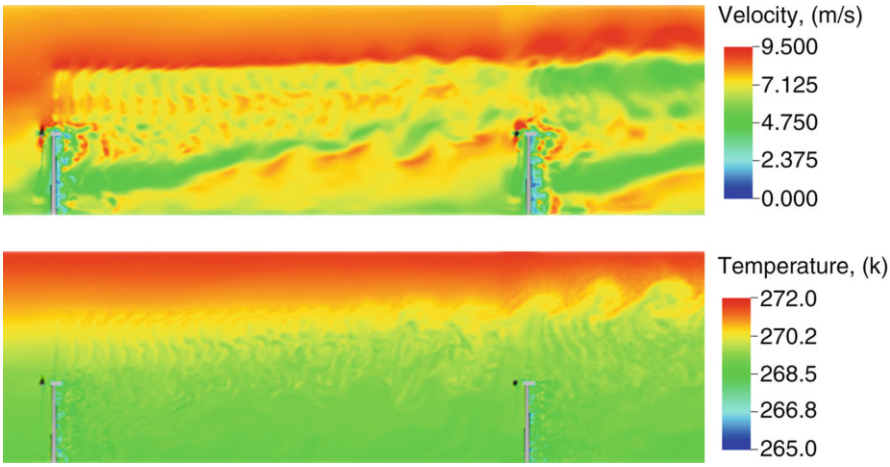


Fig. 48 2D slice of the air speed (top) and temperature (bottom)

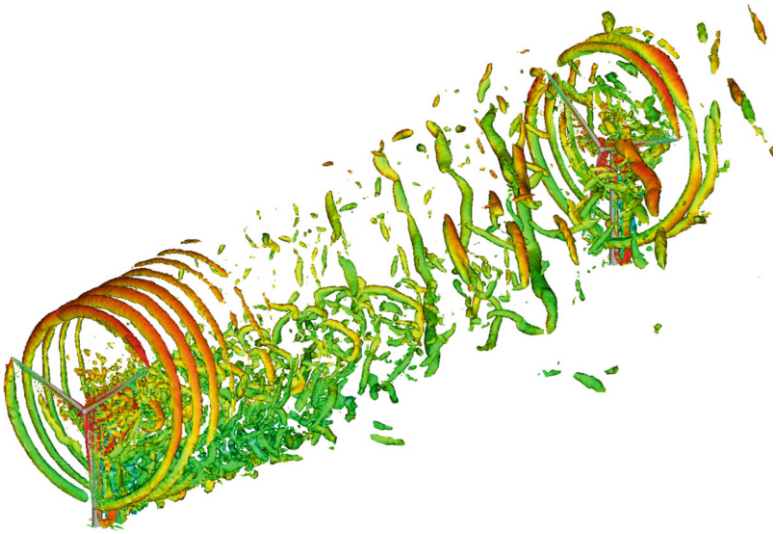


Fig. 49 3D view of vorticity isosurfaces colored by air speed

the downstream turbine is only maintained for a short distance behind the rotor. This enhanced turbulent mixing gives a faster growth of the shear layer behind the downstream turbine.

Remark 21 When simulating ABL flows, the computational domain should be large enough to account for the wake drift due to side wind and Coriolis force. Figure 50 shows the front view of the vorticity isosurfaces, where the wake drift is clearly

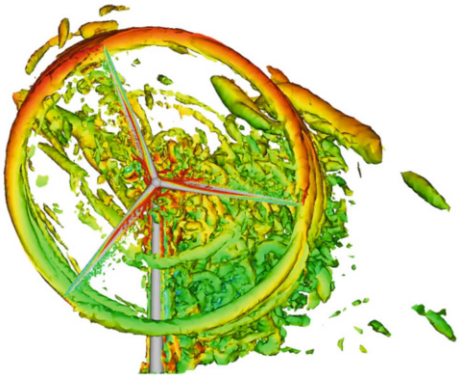


Fig. 50 Front view of vorticity isosurfaces colored by air speed

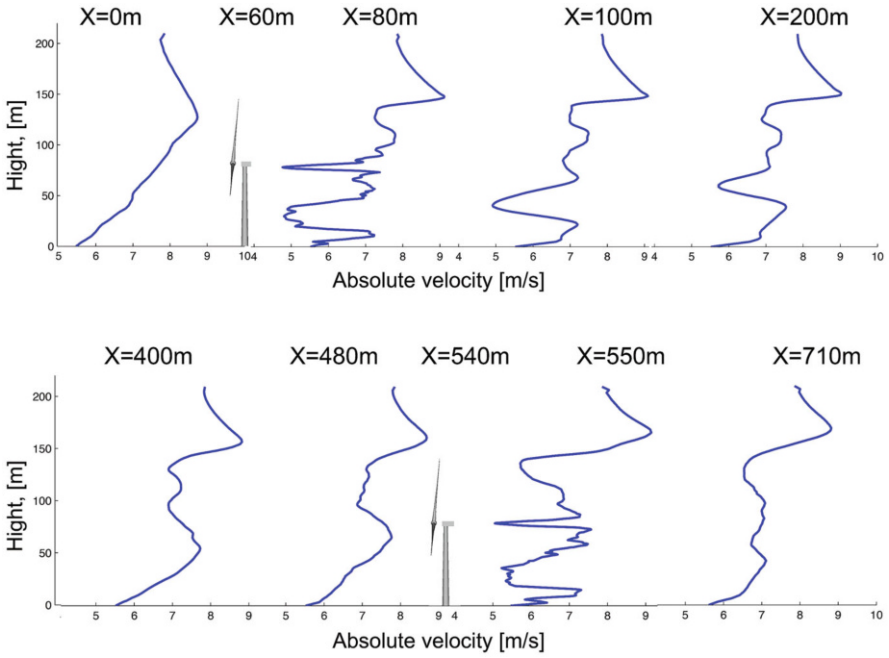


Fig. 51 Air speed averaged over six rotor revolutions and plotted at different locations along the centerline as a function of the vertical coordinate

seen. While in the present simulations wake drift is not as significant, for stronger side winds the computational domain needs to have a larger spanwise dimension.

Figure 51 shows the air speed, averaged over six rotor revolutions, at different locations along the centerline as a function of the vertical coordinate. Air speed profile at the inlet corresponds to that imposed from the LES simulation. A short

distance past *Turbine 1* the profile appears distorted, and slowly begins to recover with increasing distance from the upstream turbine. By the location of *Turbine 2* the profile begins to recover up to the hub height and above the upper-blade tip. However, qualitative differences w.r.t. the inflow profile, e.g., less near-ground shear and a higher shear above the top of the upper rotor, may be observed. In between the hub-height and upper-blade-tip locations one can clearly see the velocity deficit, which is on the order of 1–2 m/s. This velocity deficit leads to the power-production drop, as discussed in the next section.

10.3 FSI Simulation Results

In this section we present FSI simulations of the same multi-domain set-up. The wind-turbine geometry, materials, and mesh, which is comprised of 13,273 quadratic NURBS shell elements, are identical to those described in Sect. 9.1. Figure 52 shows the aerodynamic torque acting on each blade of the upstream-turbine rotor, and compares the pure aerodynamics (labeled "CFD") and FSI results. The FSI-simulation curves exhibit low frequency modes coming from the blade flapwise bending motions, as well as high frequency modes coming from the blade axial torsion motions. These modes are obviously not present in the CFD curves,

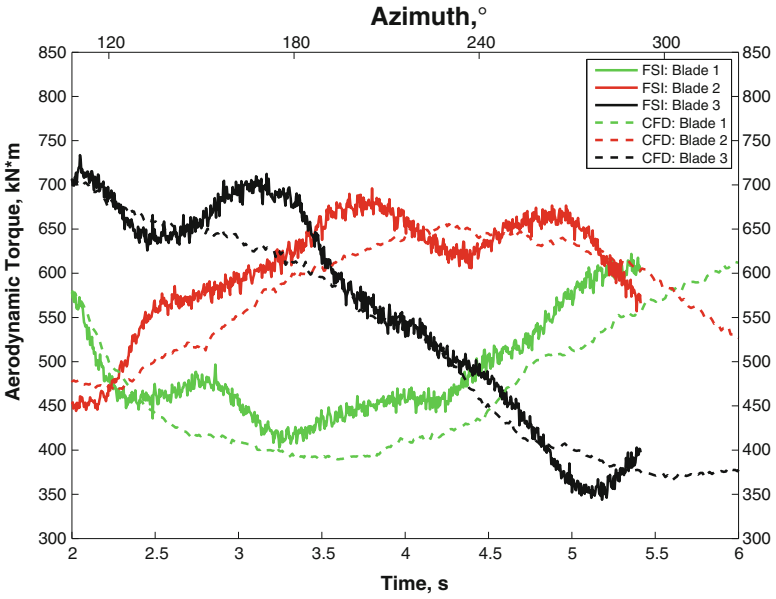


Fig. 52 Time history of the aerodynamic torque for each blade of the upstream-turbine. Comparison of pure aerodynamics (labeled "CFD") and FSI simulation results. See Fig. 44 for blade numbering

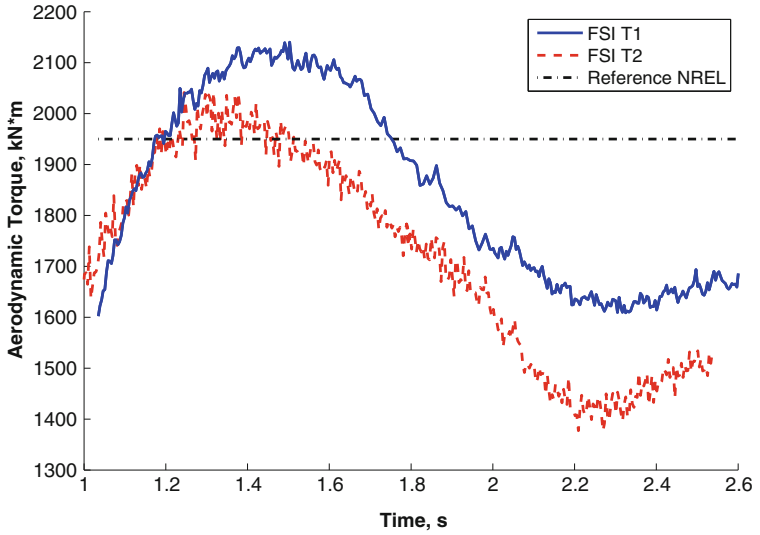


Fig. 53 Time history of the aerodynamic torque from FSI simulations for upstream (T1) and downstream (T2) turbines. Aerodynamic torque for uniform wind speed of 8 m/s from [210] is plotted for comparison

which underscore the importance of including FSI in the wind-turbine modeling, especially if one is interested in predicting the remaining useful fatigue life of wind-turbine structural components (see, e.g., [225]).

Figure 53 shows a comparison of the aerodynamic torque acting on the upstream and downstream turbines. The results confirm power losses for the downstream turbine of 10–15% relative to the upstream turbine, which are due to the velocity deficit in the upstream-turbine wake. Also note that the amplitude of high-frequency oscillations due to the blade torsional motions is a little higher for the downstream turbine, which is due to higher turbulence intensity in the upstream-turbine wake than in the free stream. The nominal aerodynamic torque from the NREL baseline design for a uniform wind speed of 8 m/s [210] is also plotted for comparison to underscore the importance of including realistic boundary-layer flow in the aerodynamics and FSI modeling of wind turbines at full scale.

11 Concluding Remarks

We described the recent advances our teams made in ALE-VMS and ST-VMS computational analysis of wind-turbine aerodynamics and FSI. The ALE-VMS and ST-VMS are the VMS versions of the ALE and DSD/SST. The VMS components are from the RBVMS. The ALE-VMS and ST-VMS served as the core methods in the computations. They were complemented by special methods that included the

ALE-VMS versions for stratified flows, sliding interfaces and weak enforcement of Dirichlet boundary conditions, ST-SI, NURBS-based isogeometric analysis, STNMUM, Kirchhoff–Love shell modeling of wind-turbine structures, and full FSI coupling. The ST framework, in a general context, provides higher-order accuracy. The VMS feature of the ALE-VMS and ST-VMS addressed the computational challenges associated with the multiscale nature of the unsteady flow, and the moving-mesh feature of the ALE and ST frameworks enabled high-resolution computation near the rotor surface. The ALE-VMS version for sliding interfaces and the ST-SI enabled moving-mesh computation of the spinning rotor. The mesh covering the rotor spins with it, and the sliding interface or the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. The ST-SI also enabled prescribing the fluid velocity at the turbine rotor surface as weakly enforced Dirichlet boundary condition. The STNMUM enabled exact representation of the mesh rotation. The analysis cases reported included both the HAWTs and VAWTs, stratified and unstratified flows, standalone wind turbines, wind turbines with tower or support columns, aerodynamic interaction between two wind turbines, and the FSI between the aerodynamics and structural dynamics of wind turbines. Comparisons with experimental data were also included where applicable. The reported cases demonstrated the effectiveness of the ALE-VMS, ST-VMS and the accompanying special methods in computational analysis of wind-turbine aerodynamics and FSI. These methods brought the aerodynamic and FSI analysis of wind turbines to a new level, where such analyses can actually support design and testing.

Acknowledgements First and second authors wish to thank the Texas Advanced Computing Center (TACC) and the San Diego Supercomputing Center (SDSC) for providing HPC resources that have contributed to the research results reported in this paper. The second author acknowledges the support of the AFOSR Award FA9550-16-1-0131 and ARO grant W911NF-14-1-0296. The work on the ST computational analysis was supported (third and fourth authors) in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (SIP), "Innovative Combustion Technology" (Funding agency: JST); and Rice–Waseda research agreement (third author). The work on the ST computational analysis was also supported (fourth author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University.

References

1. "Renewable power generation costs in 2012: An overview", Report, International Renewable Energy Agency, 2012, Available at: http://www.irena.org/Publications/.
2. "Annual energy outlook 2014, doe/eia-0383(2014)", Report, U.S. Energy Information Administration, April 2014, Available at: http://www.eia.gov/forecasts/aeo/.
3. "Eu energy in figures, statistical pocketbook 2014", Report, European Commission, 2014, https://doi.org/10.2833/24150.

4. "Levelized cost of electricity renewable energy technologies", Study, FRAUNHOFER INSTITUT FOR SOLAR ENERGY SYSTEMS ISE, November 2013.
5. N.N. Sørensen, J.A. Michelsen, and S. Schreck, "Navier–Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80 ft × 120 ft wind tunnel", *Wind Energy*, **5** (2002) 151–169.
6. A.L. Pape and J. Lecanu, "3D Navier–Stokes computations of a stall-regulated wind turbine", *Wind Energy*, **7** (2004) 309–324.
7. F. Zahle, N.N. Sørensen, and J. Johansen, "Wind turbine rotor-tower interaction using an incompressible overset grid method", *Wind Energy*, **12** (2009) 594–619.
8. 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, https://doi.org/10.1002/fld.2400.
9. 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, https://doi.org/10.1007/s00466-011-0589-2.
10. Y. Li and P.M.C. Kim-Jong Paik, T. Xing, "Dynamic overset CFD simulations of wind turbine aerodynamics", *Renewable Energy*, **37** (2012) 285–298.
11. E. Guttierez, S. Primi, F. Taucer, P. Caperan, D. Tirelli, J. Mieres, I. Calvo, J. Rodriguez, F. Vallano, G. Galiotis, and D. Mouzakis, "A wind turbine tower design based on fibre-reinforced composites", Technical report, Joint Research Centre - Ispra, European Laboratory for Structural Assessment (ELSA), Institute For Protection and Security of the Citizen (IPSC), European Commission, 2003.
12. C. Kong, J. Bang, and Y. Sugiyama, "Structural investigation of composite wind turbine blade considering various load cases and fatigue life", *Energy*, **30** (2005) 2101–2114.
13. M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N. Sørensen, and H.A. Madsen, "State of the art in wind turbine aerodynamics and aeroelasticity", *Progress in Aerospace Sciences*, **42** (2006) 285–330.
14. F.M. Jensen, B.G. Falzon, J. Ankersen, and H. Stang, "Structural testing and numerical simulation of a 34 m composite wind turbine blade", *Composite Structures*, **76** (2006) 52–61.
15. J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger, "The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 2403–2416.
16. 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.
17. 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.
18. S. Schreck, J. Lundquist, and W.Shaw, "U.s. department of energy workshop report: Research needs for wind resource characterization", Technical Report NREL/TP-500-43521, National Renewable Energy Laboratory, 2008.
19. R. Barthelmie1, S. Frandsen, O. Rathmann, K. Hansen, E. Politis, J. Prospathopoulos, J. Schepers, K. Rados, D. Cabezn, W. Schlez, A. Neubert, and M. Heath, "Flow and wakes in large wind farms: Final report for upwind wp8", Technical Report Report number Ris-R-1765(EN), Danmarks Tekniske Universitet, Ris Nationallaboratoriet for Bredygtig Energ, 2011.
20. A. Westerhellweg, B. Caadillas, F. Kinder, and T. Neumann, "Wake measurements at alpha ventus dependency on stability and turbulence intensity", *Journal of Physics: Conference Series (Online)*, **555** (2014), https://doi.org/10.1088/1742-6596/555/1/012106.
21. 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.

22. J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes, "Isogeometric analysis of structural vibrations", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 5257–5297.
23. Y. Bazilevs, L.B. da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli, "Isogeometric analysis: Approximation, stability and error estimates for *h*-refined meshes", *Mathematical Models and Methods in Applied Sciences*, **16** (2006) 1031–1090.
24. J.A. Cottrell, T.J.R. Hughes, and A. Reali, "Studies of refinement and continuity in isogeometric structural analysis", *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 4160–4183.
25. W.A. Wall, M.A. Frenzel, and C. Cyron, "Isogeometric structural shape optimization", *Computer Methods in Applied Mechanics and Engineering*, **197** (2008) 2976–2988.
26. J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.
27. J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes, "*n*-Widths, sup-infs, and optimality ratios for the *k*-version of the isogeometric finite element method", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 1726–1741.
28. M.R. Dörfel, B. Jüttler, and B. Simeon, "Adaptive isogeometric analysis by local *h*-refinement with T-splines", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 264–275.
29. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg, "Isogeometric analysis using T-splines", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 229–263.
30. F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali, "The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: Mixed FEMs versus NURBS-based approximations", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 314–323.
31. W. Wang and Y. Zhang, "Wavelets-based NURBS simplification and fairing", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 290–300.
32. E. Cohen, T. Martin, R.M. Kirby, T. Lyche, and R.F. Riesenfeld, "Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 334–356.
33. V. Srinivasan, S. Radhakrishnan, and G. Subbarayan, "Coordinated synthesis of hierarchical engineering systems", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 392–404.
34. 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.
35. Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes, "Weak Dirichlet boundary conditions for wall-bounded turbulent flows", *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 4853–4862.
36. 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.
37. 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.
38. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "Improving stability of stabilized and multiscale formulations in flow simulations at small time steps", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840, https://doi.org/10.1016/j.cma.2009.06.019.
39. 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.

40. T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes, "B-bar and F-bar projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order nurbs elements", *Computer Methods in Applied Mechanics and Engineering*, **197** (2008) 2732–2762.
41. S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, and T.J.R. Hughes, "Robustness of isogeometric structural discretizations under severe mesh distortion", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 357–373.
42. D.J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T.J.R. Hughes, and T. Belytschko, "A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM", *International Journal for Numerical Methods in Engineering*, **83** (2010) 765–785.
43. D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes, "Isogeometric shell analysis: The Reissner–Mindlin shell", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 276–289.
44. J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner, "Isogeometric shell analysis with Kirchhoff–Love elements", *Computer Methods in Applied Mechanics and Engineering*, **198** (2009) 3902–3914.
45. Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T.J.R. Hughes, "Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow", *Computer Methods in Applied Mechanics and Engineering*, **196** (2007) 2943–2959.
46. 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.
47. 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.
48. J.G. Isaksen, Y. Bazilevs, T. Kvamsdal, Y. Zhang, J.H. Kaspersen, K. Waterloo, B. Romner, and T. Ingebrigtsen, "Determination of wall tension in cerebral artery aneurysms by numerical simulation", *Stroke*, **39** (2008) 3172–3178.
49. 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.
50. F. Cirak, M. Ortiz, and P. Schröder, "Subdivision surfaces: a new paradigm for thin shell analysis", *International Journal for Numerical Methods in Engineering*, **47** (2000) 2039–2072.
51. F. Cirak and M. Ortiz, "Fully C^1-conforming subdivision elements for finite deformation thin shell analysis", *International Journal for Numerical Methods in Engineering*, **51** (2001) 813–833.
52. F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, and P. Schröder, "Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision", *Computer-Aided Design*, **34** (2002) 137–148.
53. 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.
54. 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.
55. 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.
56. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", *Computers and Fluids*, **36** (2007) 12–26.
57. J. Nitsche, "Uber ein variationsprinzip zur losung von Dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind", *Abh. Math. Univ. Hamburg*, **36** (1971) 9–15.

58. D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini, "Unified analysis of Discontinuous Galerkin methods for elliptic problems", *SIAM Journal of Numerical Analysis*, **39** (2002) 1749–1779.
59. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S. Gohari, and S. Sarkar, "ALE–VMS formulation for stratified turbulent incompressible flows with applications", *Mathematical Models and Methods in Applied Sciences*, **25** (12) (2015) 2349–2375.
60. J. Yan, A. Korobenko, A. Tejada-Martinez, R. Golshan, and Y. Bazilevs, "A new variational multiscale formulation for stratified incompressible turbulent flows", *Computers & Fluids*, **158** (2017) 150–156.
61. A. Korobenko, J. Yan, S. Gohari, S. Sarkar, and Y. Bazilevs, "FSI simulation of two back-to-back wind turbines in atmospheric boundary layer flow", *Computers & Fluids*, **158** (2017) 167–175.
62. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "High-performance computing of wind turbine aerodynamics using isogeometric analysis", *Computers and Fluids*, **49** (2011) 93–100.
63. Y. Bazilevs, M.-C. Hsu, and M.A. Scott, "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines", *Computer Methods in Applied Mechanics and Engineering*, **249–252** (2012) 28–41.
64. 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.
65. 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.
66. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre, "Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods", *Archives of Computational Methods in Engineering*, **21** (2014) 359–398, https://doi.org/10.1007/s11831-014-9119-7.
67. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines", *International Journal for Numerical Methods in Engineering*, **102** (2015) 766–783, https://doi.org/10.1002/nme.4738.
68. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Aerodynamic simulation of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2013) 021011, https://doi.org/10.1115/1.4024415.
69. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel, and J.O. Dabiri, "FSI modeling of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2014) 081006, https://doi.org/10.1115/1.4027466.
70. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs, "Computational free-surface fluid–structure interaction with application to floating offshore wind turbines", *Computers and Fluids*, **141** (2016) 155–174,
71. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S.M.I. Gohari, and S. Sarkar, "ALE–VMS formulation for stratified turbulent incompressible flows with applications", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2349–2375, https://doi.org/10.1142/S0218202515400114.
72. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades", *Journal of Applied Mechanics*, **83** (6) (2016) 061010.
73. 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.
74. 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.

75. 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.
76. 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.
77. 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.
78. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Fluid–structure interaction simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **52** (2013) 971–981, https://doi.org/10.1007/s00466-013-0858-3.
79. C.C. Long, M. Esmaily-Moghadam, A.L. Marsden, and Y. Bazilevs, "Computation of residence time in the simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **54** (2014) 911–919, https://doi.org/10.1007/s00466-013-0931-y.
80. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk", *Computational Mechanics*, **54** (2014) 921–932, https://doi.org/10.1007/s00466-013-0967-z.
81. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation", *Computational Mechanics*, **54** (2014) 1055–1071, https://doi.org/10.1007/s00466-014-1059-4.
82. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks, "Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models", *Computational Mechanics*, **55** (2015) 1211–1225, https://doi.org/10.1007/s00466-015-1166-x.
83. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 1005–1053.
84. 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.
85. 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.
86. C. Wang, M.C.H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs, "Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis", *Computers and Fluids*, **142** (2017) 3–14, https://doi.org/10.1016/j.compfluid.2015.12.004.
87. M.C.H. Wu, D. Kamensky, C. Wang, A.J. Herrema, F. Xu, M.S. Pigazzini, A. Verma, A.L. Marsden, Y. Bazilevs, and M.-C. Hsu, "Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear", *Computer Methods in Applied Mechanics and Engineering*, (2017), Published online. https://doi.org/10.1016/j.cma.2016.09.032.
88. J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs, "Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines", *Computers and Fluids*, **158** (2017) 157–166, https://doi.org/10.1016/j.compfluid.2016.06.016.
89. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "Experimental and numerical FSI study of compliant hydrofoils", *Computational Mechanics*, **55** (2015) 1079–1090, https://doi.org/10.1007/s00466-014-1090-5.

90. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration", *Computers and Fluids*, **141** (2016) 201–211, https://doi.org/10.1016/j.compfluid.2015.07.013.
91. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", *Computational Mechanics*, **48** (2011) 247–267, https://doi.org/10.1007/s00466-011-0571-z.
92. K. Takizawa and T.E. Tezduyar, "Space–time fluid–structure interaction methods", *Mathematical Models and Methods in Applied Sciences*, **22** (supp02) (2012) 1230001, https://doi.org/10.1142/S0218202512300013.
93. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, "Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, https://doi.org/10.1142/S0218202515400072.
94. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44, https://doi.org/10.1016/S0065-2156(08)70153-4.
95. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, https://doi.org/10.1002/fld.505.
96. 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, https://doi.org/10.1002/fld.1430.
97. 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, https://doi.org/10.1002/fld.1650150911.
98. 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, https://doi.org/10.1002/fld.1650211011.
99. 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, https://doi.org/10.1016/S0045-7825(00)00204-8.
100. 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, https://doi.org/10.1016/j.cma.2004.09.014.
101. 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, https://doi.org/10.1007/s00466-008-0261-7.
102. 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, https://doi.org/10.1007/s00466-008-0260-8.
103. 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.
104. K. Takizawa and T.E. Tezduyar, "Computational methods for parachute fluid–structure interactions", *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, https://doi.org/10.1007/s11831-012-9070-4.
105. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN: 978-0470978771.
106. 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, https://doi.org/10.1007/s00466-012-0761-3.

107. 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, https://doi.org/10.1007/s00466-013-0880-5.
108. K. Takizawa, T.E. Tezduyar, C. Boswell, Y. Tsutsui, and K. Montel, "Special methods for aerodynamic-moment calculations from parachute FSI modeling", *Computational Mechanics*, **55** (2015) 1059–1069, https://doi.org/10.1007/s00466-014-1074-5.
109. 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, https://doi.org/10.1142/S0218202513400058.
110. K. Takizawa, T.E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel, "FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1203–1220, https://doi.org/10.1007/s00466-014-1052-y.
111. K. Takizawa, T.E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel, "Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1461–1476, https://doi.org/10.1007/s00466-014-1069-2.
112. K. Takizawa, T.E. Tezduyar, and R. Kolesar, "FSI modeling of the Orion spacecraft drogue parachutes", *Computational Mechanics*, **55** (2015) 1167–1179, https://doi.org/10.1007/s00466-014-1108-z.
113. 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, https://doi.org/10.1007/s00466-011-0614-5.
114. 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, https://doi.org/10.1007/s00466-013-0888-x.
115. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, M.-C. Hsu, O. Øiseth, K.M. Mathisen, N. Kostov, and S. McIntyre, "Engineering analysis and design with ALE-VMS and space–time methods", *Archives of Computational Methods in Engineering*, **21** (2014) 481–508, https://doi.org/10.1007/s11831-014-9113-0.
116. K. Takizawa, "Computational engineering analysis with the new-generation space–time methods", *Computational Mechanics*, **54** (2014) 193–211, https://doi.org/10.1007/s00466-014-0999-z.
117. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, "Space–time VMS method for flow computations with slip interfaces (ST-SI)", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, https://doi.org/10.1142/S0218202515400126.
118. 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, https://doi.org/10.1115/1.4005073.
119. 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, https://doi.org/10.1007/s00466-012-0759-x.
120. 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, https://doi.org/10.1016/j.compfluid.2012.11.008.
121. 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, https://doi.org/10.1007/s00466-012-0758-y.

122. K. Takizawa, T.E. Tezduyar, and N. Kostov, "Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV", *Computational Mechanics*, **54** (2014) 213–233, https://doi.org/10.1007/s00466-014-0980-x.
123. 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, https://doi.org/10.1007/s00466-013-0935-7.
124. K. Takizawa, T.E. Tezduyar, and A. Buscher, "Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping", *Computational Mechanics*, **55** (2015) 1131–1141, https://doi.org/10.1007/s00466-014-1095-0.
125. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt, "ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling", *Mathematical Models and Methods in Applied Sciences*, **24** (2014) 2437–2486, https://doi.org/10.1142/S0218202514500250.
126. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent", *Computational Mechanics*, **50** (2012) 675–686, https://doi.org/10.1007/s00466-012-0760-4.
127. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms", *Computational Mechanics*, **51** (2013) 1061–1073, https://doi.org/10.1007/s00466-012-0790-y.
128. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, and T. Ueda, "FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta", *Computational Mechanics*, **54** (2014) 1035–1045, https://doi.org/10.1007/s00466-014-1017-1.
129. K. Takizawa, T.E. Tezduyar, H. Uchikawa, T. Terahara, T. Sasaki, K. Shiozaki, A. Yoshida, K. Komiya, and G. Inoue, "Aorta flow analysis and heart valve flow and structure analysis", to appear in a special volume to be published by Springer, 2018.
130. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, "Space–time fluid mechanics computation of heart valve models", *Computational Mechanics*, **54** (2014) 973–986, https://doi.org/10.1007/s00466-014-1046-9.
131. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the Space–Time Slip Interface Topology Change (ST-SI-TC) method and Isogeometric Analysis (IGA)", in P. Wriggers and T. Lenarz, editors, *Biomedical Technology: Modeling, Experiments and Simulation*, Lecture Notes in Applied and Computational Mechanics, 77–99, Springer, 2018, ISBN: 978-3-319-59547-4.
132. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods", *Computers & Fluids*, **158** (2017) 176–188, https://doi.org/10.1016/j.compfluid.2016.11.012.
133. K. Takizawa, D. Montes, S. McIntyre, and T.E. Tezduyar, "Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers", *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 223–248, https://doi.org/10.1142/s0218202513400022.
134. K. Takizawa, T.E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi, "Computational thermo-fluid analysis of a disk brake", *Computational Mechanics*, **57** (2016) 965–977, https://doi.org/10.1007/s00466-016-1272-4.
135. K. Takizawa, T.E. Tezduyar, and H. Hattori, "Computational analysis of flow-driven string dynamics in turbomachinery", *Computers & Fluids*, **142** (2017) 109–117, https://doi.org/10.1016/j.compfluid.2016.02.019.
136. K. Takizawa, T.E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori, "Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA)", *Computers & Fluids*, **142** (2017) 15–20, https://doi.org/10.1016/j.compfluid.2016.02.021.
137. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method", *Computers & Fluids*, **158** (2017) 189–200, https://doi.org/10.1016/j.compfluid.2017.04.017.

138. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "A general-purpose NURBS mesh generation method for complex geometries", to appear in a special volume to be published by Springer, 2018.
139. K. Takizawa, T.E. Tezduyar, S. Asada, and T. Kuraishi, "Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC)", *Computers & Fluids*, **141** (2016) 124–134, https://doi.org/10.1016/j.compfluid.2016.05.006.
140. T. Kuraishi, K. Takizawa, and T.E. Tezduyar, "Space–time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation", to appear in a special volume to be published by Springer, 2018.
141. K. Takizawa, T.E. Tezduyar, and T. Terahara, "Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)", *Computers & Fluids*, **141** (2016) 191–200, https://doi.org/10.1016/j.compfluid.2016.05.027.
142. K. Takizawa, T.E. Tezduyar, and T. Kanai, "Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity", *Mathematical Models and Methods in Applied Sciences*, **27** (2017) 771–806, https://doi.org/10.1142/S0218202517500166.
143. M.-C. Hsu and Y. Bazilevs, "Fluid–structure interaction modeling of wind turbines: simulating the full machine", *Computational Mechanics*, **50** (2012) 821–833.
144. 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, https://doi.org/10.1016/0045-7825(94)00082-4.
145. K. Takizawa and T.E. Tezduyar, "Space–time computation techniques with continuous representation in time (ST-C)", *Computational Mechanics*, **53** (2014) 91–99, https://doi.org/10.1007/s00466-013-0895-y.
146. Y. Osawa, V. Kalro, and T. Tezduyar, "Multi-domain parallel computation of wake flows", *Computer Methods in Applied Mechanics and Engineering*, **174** (1999) 371–391, https://doi.org/10.1016/S0045-7825(98)00305-3.
147. C. Johnson, *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, Sweden, 1987.
148. S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, 2002.
149. A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*. Springer, 2004.
150. T.J.R. Hughes and T.E. Tezduyar, "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations", *Computer Methods in Applied Mechanics and Engineering*, **45** (1984) 217–284, https://doi.org/10.1016/0045-7825(84)90157-9.
151. T.E. Tezduyar and Y.J. Park, "Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 307–325, https://doi.org/10.1016/0045-7825(86)90003-4.
152. T.J.R. Hughes, L.P. Franca, and M. Balestra, "A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 85–99.
153. T.E. Tezduyar and Y. Osawa, "Finite element stabilization parameters computed from element matrices and vectors", *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 411–430, https://doi.org/10.1016/S0045-7825(00)00211-5.
154. T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.-B. Quincy, "The variational multiscale method–A paradigm for computational mechanics", *Computer Methods in Applied Mechanics and Engineering*, **166** (1998) 3–24.
155. T.J.R. Hughes and G. Sangalli, "Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods", *SIAM Journal of Numerical Analysis*, **45** (2007) 539–557.

156. F. Shakib, T.J.R. Hughes, and Z. Johan, "A multi-element group preconditionined GMRES algorithm for nonsymmetric systems arising in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, **75** (1989) 415–456.
157. T.J.R. Hughes and M. Mallet, "A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems", *Computer Methods in Applied Mechanics and Engineering*, **58** (1986) 305–328.
158. T.J.R. Hughes, G. Scovazzi, and T.E. Tezduyar, "Stabilized methods for compressible flows", *Journal of Scientific Computing*, **43** (2010) 343–368, https://doi.org/10.1007/s10915-008-9233-5.
159. B.E. Launder and D.B. Spalding, "The numerical computation of turbulent flows", *Computer Methods in Applied Mechanics and Engineering*, **3** (1974) 269–289.
160. D.C. Wilcox, *Turbulence Modeling for CFD*. DCW Industries, La Canada, CA, 1998.
161. R. Golshan, A. Tejada-Martínez, M. Juha, and Y. Bazilevs, "Large-eddy simulation with near-wall modeling using weakly enforced no-slip boundary conditions", *Computers & Fluids*, **118** (2015) 172–181.
162. 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, https://doi.org/10.1016/j.cma.2005.08.023.
163. T.E. Tezduyar and D.K. Ganjoo, "Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion problems", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 49–71, https://doi.org/10.1016/0045-7825(86)90023-X.
164. G.J. Le Beau, S.E. Ray, S.K. Aliabadi, and T.E. Tezduyar, "SUPG finite element computation of compressible flows with the entropy and conservation variables formulations", *Computer Methods in Applied Mechanics and Engineering*, **104** (1993) 397–422, https://doi.org/10.1016/0045-7825(93)90033-T.
165. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, https://doi.org/10.1016/j.compfluid.2005.02.011.
166. 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, https://doi.org/10.1016/j.cma.2005.05.032.
167. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", *Computers & Fluids*, **36** (2007) 147–159, https://doi.org/10.1016/j.compfluid.2005.07.009.
168. 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, https://doi.org/10.1007/s00466-005-0025-6.
169. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, https://doi.org/10.1007/s00466-006-0056-7.
170. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364, https://doi.org/10.1007/s00466-006-0045-x.
171. 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, https://doi.org/10.1016/j.compfluid.2005.07.004.
172. 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, https://doi.org/10.1002/fld.1743.
173. 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, https://doi.org/10.1002/fld.1447.

174. 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, https://doi.org/10.1002/fld.1484.
175. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms", *Journal of Applied Mechanics*, **76** (2009) 021211, https://doi.org/10.1115/1.3062967.
176. 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, https://doi.org/10.1115/1.3057496.
177. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167, https://doi.org/10.1007/s00466-009-0441-0.
178. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270, https://doi.org/10.1002/fld.2451.
179. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique", *Journal of Applied Mechanics*, **79** (2012) 010910, https://doi.org/10.1115/1.4005060.
180. A. Corsini, F. Rispoli, A.G. Sheard, and T.E. Tezduyar, "Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations", *Computational Mechanics*, **50** (2012) 695–705, https://doi.org/10.1007/s00466-012-0789-4.
181. P.A. Kler, L.D. Dalcin, R.R. Paz, and T.E. Tezduyar, "SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems", *Computational Mechanics*, **51** (2013) 171–185, https://doi.org/10.1007/s00466-012-0712-z.
182. A. Corsini, F. Rispoli, A.G. Sheard, K. Takizawa, T.E. Tezduyar, and P. Venturini, "A variational multiscale method for particle-cloud tracking in turbomachinery flows", *Computational Mechanics*, **54** (2014) 1191–1202, https://doi.org/10.1007/s00466-014-1050-0.
183. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T.E. Tezduyar, "Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing", *Computational Mechanics*, **55** (2015) 1201–1209, https://doi.org/10.1007/s00466-015-1160-3.
184. K. Takizawa, T.E. Tezduyar, and Y. Otoguro, "Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations", *Computational Mechanics*, published online, https://doi.org/10.1007/s00466-018-1557-x, April 2018.
185. 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.
186. Y. Otoguro, K. Takizawa, T.E. Tezduyar, K. Nagaoka, and S. Mei, "Turbocharger turbine and exhaust manifold flow computation with the Space–Time Variational Multiscale Method and Isogeometric Analysis", *Computers & Fluids*, submitted, 2018.
187. S.B. Raknes, X. Deng, Y. Bazilevs, D.J. Benson, K.M. Mathisen, and T. Kvamsdal, "Isogeometric rotation-free bending-stabilized cables: Statics, dynamics, bending strips and coupling with shells", *Computer Methods in Applied Mechanics and Engineering*, **263** (2013) 127–143.
188. L. Piegl and W. Tiller, *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.
189. T. Belytschko, W.K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

190. M. Bischoff, W.A. Wall, K.-U. Bletzinger, and E. Ramm, "Models and finite elements for thin-walled structures", in E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 2, Solids, Structures and Coupled Problems*, Chapter 3, Wiley, 2004.
191. J.N. Reddy, *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, 2nd ed.* CRC Press, Boca Raton, FL, 2004.
192. Y. Guo and M. Ruess, "Nitsche's method for a coupling of isogeometric thin shells and blended shell structures", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 881–905.
193. 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.
194. H. Melbø and T. Kvamsdal, "Goal oriented error estimators for Stokes equations based on variationally consistent postprocessing", *Computer Methods in Applied Mechanics and Engineering*, **192** (2003) 613–633.
195. E.H. van Brummelen, V.V. Garg, S. Prudhomme, and K.G. van der Zee, "Flux evaluation in primal and dual boundary-coupled problems", *Journal of Applied Mechanics*, **79** (2011) 010904.
196. 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.
197. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, "Parallel finite-element computation of 3D flows", *Computer*, **26** (10) (1993) 27–36, https://doi.org/10.1109/2.237441.
198. 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, https://doi.org/10.1016/0045-7825(94)00077-8.
199. T.E. Tezduyar, "Finite element methods for flow problems with moving boundaries and interfaces", *Archives of Computational Methods in Engineering*, **8** (2001) 83–130, https://doi.org/10.1007/BF02897870.
200. T. Tezduyar, "Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces", in *Proceedings of the ASME Symposium on Fluid-Physics and Heat Transfer for Macro- and Micro-Scale Gas-Liquid and Phase-Change Flows (CD-ROM)*, ASME Paper IMECE2001/HTD-24206, ASME, New York, New York, (2001).
201. T.E. Tezduyar, "Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows", in M.M. Hafez, editor, *Numerical Simulations of Incompressible Flows*, World Scientific, New Jersey, (2003) 221–239, https://doi.org/10.1142/9789812796837_0013.
202. K. Stein, T.E. Tezduyar, and R. Benney, "Automatic mesh update with the solid-extension mesh moving technique", *Computer Methods in Applied Mechanics and Engineering*, **193** (2004) 2019–2032, https://doi.org/10.1016/j.cma.2003.12.046.
203. T.E. Tezduyar, S. Sathe, M. Senga, L. Aureli, K. Stein, and B. Griffin, "Finite element modeling of fluid–structure interactions with space–time and advanced mesh update techniques", in *Proceedings of the 10th International Conference on Numerical Methods in Continuum Mechanics (CD-ROM)*, Zilina, Slovakia, (2005).
204. 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.
205. O. Rodrigues, "Des lois geometriques qui regissent les deplacements dun systeme solide dans lespace, et de la variation des coordonnees provenant de ces deplacements consideres independamment des causes qui peuvent les produire", *Journal de Mathematiques*, **5** (1840) 380–440.

206. B. Gayen, S. Sarkar, and J.R. Taylor, "Large eddy simulation of a stratified boundary layer under an oscillatory current", *Journal of Fluid Mechanics*, **643** (2010) 233–266.
207. B. Gayen and S. Sarkar, "Direct and large-eddy simulations of internal tide generation at a near-critical slope.", *Journal of Fluid Mechanics*, **681** (2011) 48–79.
208. R. Beare, M. Macvean, A. Holtslag, J. Cuxart, I. Esau, J.-C. Golaz, M. Jimenez, M. Khairoutdinov, B. Kosovic, D. Lewellen, T. Lund, J. Lundquist, A. Mccabe, A. Moene, Y. Noh, S. Raasch, and P. Sullivan, "An intercomparison of large-eddy simulations of the stable boundary layer", *Boundary-Layer Meteor.*, **118** (2006) 247–272.
209. D.T. Griffith and T.D. Ashwill, "The sandia 100-meter all-glass baseline wind turbine blade: Snl100-00", *SANDIA REPORT, SAND2011-3779*, (2011).
210. J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development", Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.
211. R. Bravo, S. Tullis, and S. Ziada, "Performance testing of a small vertical-axis wind turbine", *Proceedings of the 21st Canadian Congress of Applied Mechanics*, (2007) 470–471.
212. K. McLaren, S. Tullis, and S. Ziada, "Computational fluid dynamics simulation of the aerodynamics of a high solidity, small-scale vertical axis wind turbine", *Wind Energy*, **15** (2012) 349–361, Published online. https://doi.org/10.1002/we.472.
213. J.O. Dabiri, "Potential order-of-magnitude enhancement of wind farm power density via counter-rotating vertical-axis wind turbine arrays", *Journal of Renewable and Sustainable Energy*, **3** (2011) 043104.
214. "Life tower", http://cosmosunfarm.co.jp/lifetower.html.
215. Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D.J. Benson, "A computational procedure for prebending of wind turbine blades", *International Journal for Numerical Methods in Engineering*, **89** (2012) 323–336.
216. "Windspire Energy", Available at: http://www.windspireenergy.com/.
217. N. Hill, R. Dominy, G. Ingram, and J. Dominy, "Darrieus turbines: the physics of self-starting", *Proc. IMechE Part A: J. Power and Energy*, **223(1)** (2009) 21–29.
218. J.R. Baker, "Features to aid or enable self starting of pitched low solidity vertical axis wind turbines", *J. Wind Eng. Ind. Aerodyn.*, **15** (1983) 369–380.
219. Y. Osawa and T. Tezduyar, "A multi-domain method for 3D computation of wake flow behind a circular cylinder", *Computational Fluid Dynamics Journal*, **8** (1999) 296–308.
220. T. Tezduyar and Y. Osawa, "Methods for parallel computation of complex flow problems", *Parallel Computing*, **25** (1999) 2039–2066, https://doi.org/10.1016/S0167-8191(99)00080-0.
221. 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, https://doi.org/10.1016/S0045-7825(01)00310-3.
222. 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, https://doi.org/10.1016/S0045-7825(01)00311-5.
223. M. Jalali, N. Rapaka, and S. Sarkar, "Tidal flow over topography: effect of excursion number on wave energetics and turbulence", *Journal of Fluid Mechanics*, **750** (2014) 259–283.
224. S. Gohari and S. Sarkar, "Tidal flow over topography: effect of excursion number on wave energetics and turbulence", *Boundary-Layer Meteorology*, (2016), Accepted for publication.
225. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "Fluid–structure interaction modeling for fatigue-damage prediction in full-scale wind-turbine blades", *Journal of Applied Mechanics*, **83** (6) (2016) 061010.

Space–Time Computational Analysis of Tire Aerodynamics with Actual Geometry, Road Contact, and Tire Deformation

Takashi Kuraishi, Kenji Takizawa, and Tayfun E. Tezduyar

Abstract A new space–time (ST) computational method, "ST-SI-TC-IGA," is enabling us to address the challenges faced in computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation. The core component of the ST-SI-TC-IGA is the ST Variational Multiscale (ST-VMS) method, and the other key components are the ST Slip Interface (ST-SI) and ST Topology Change (ST-TC) methods and the ST Isogeometric Analysis (ST-IGA). The VMS feature of the ST-VMS addresses the challenge created by the turbulent nature of the flow, the moving-mesh feature of the ST framework enables high-resolution computation near the moving fluid–solid interfaces, and the higher-order accuracy of the ST framework strengthens both features. The ST-SI enables high-resolution representation of the boundary layers near the tire. The mesh covering the tire spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. The ST-TC enables moving-mesh computation even with the TC created by the contact between the tire and the road. It deals with the contact while maintaining high-resolution representation near the tire. Integration of the ST-SI and ST-TC enables high-resolution representation even though parts of the SI are coinciding with the tire and road surfaces. It also enables dealing with the tire-road contact location change and contact sliding. By integrating the ST-IGA with the ST-SI and ST-TC, in addition to having a more accurate representation of the tire surfaces and increased accuracy in the flow solution, the

T. Kuraishi (✉) · K. Takizawa
Department of Modern Mechanical Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: Takashi.Kuraishi@tafsm.org; Kenji.Takizawa@tafsm.org

T. E. Tezduyar
Mechanical Engineering, Rice University, Houston, TX, USA

Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: tezduyar@tafsm.org

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_8

element density in the tire grooves and in the narrow spaces near the contact areas is kept at a reasonable level. We present computations with the ST-SI-TC-IGA and two models of flow around a rotating tire with road contact and prescribed deformation. One is a simple 2D model, and one is a 3D model with an actual tire geometry that includes the longitudinal and transverse grooves. The computations show the effectiveness of the ST-SI-TC-IGA in tire aerodynamics.

1 Introduction

A new space–time (ST) method, "ST-SI-TC-IGA" [1], is enabling successful computational analysis of tire aerodynamics with actual geometry, road contact, and tire deformation. The computational challenges addressed with the ST-SI-TC-IGA include the complexity of an actual tire geometry with longitudinal and transverse grooves, turbulent nature of the flow, moving fluid–solid interfaces, spin of the tire, and the topology change (TC) created by the road contact and tire deformation. The ST-SI-TC-IGA was introduced [1] in the context of heart valve flow analysis. Its core component is the ST Variational Multiscale (ST-VMS) method [2–4], and the other key components are the ST Slip Interface (ST-SI) [5, 6] and ST-TC [7, 8] methods and the ST Isogeometric Analysis (ST-IGA) [2, 9, 10].

1.1 ST-VMS

The ST-VMS is the VMS version of the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [11–13]. The DSD/SST was introduced for computation of flows with moving boundaries and interfaces (MBI), including fluid–structure interactions (FSI). In such computations, the DSD/SST functions as a moving-mesh method. Moving the fluid mechanics mesh to track (i.e., follow) a fluid–solid interface enables mesh-resolution control near the interface and, consequently, high-resolution representation of the boundary layer. Because the stabilization components of the DSD/SST are the Streamline-Upwind/Petrov-Galerkin (SUPG) [14] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [11] stabilizations, the method is now also called "ST-SUPS." The VMS components of the ST-VMS are from the residual-based VMS (RBVMS) method [15–18]. There are two more stabilization terms beyond those the ST-SUPS has, and these additional terms give the method better turbulence modeling features. The ST-SUPS and ST-VMS, because of the higher-order accuracy of the ST framework (see [2, 3]), are desirable also in computations that do not involve any MBI.

The Arbitrary Lagrangian–Eulerian (ALE) method is an older and more commonly used moving-mesh method. The ALE finite element method was introduced in 1981 [19]. The ALE-VMS method [20–25] is the VMS version of the ALE. It was introduced after the ST-SUPS [11] and ALE-SUPS [26] methods and preceded the ST-VMS. The ALE-VMS and RBVMS are often supplemented with special methods, such as those for weakly-enforced no-slip boundary condition [27–29], "sliding interfaces" [30, 31] and backflow stabilization [32]. They have been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include wind-turbine aerodynamics and FSI [33–40], more specifically, vertical-axis wind turbines [41, 42], floating wind turbines [43], wind turbines in atmospheric boundary layers [44], and fatigue damage in wind-turbine blades [45], patient-specific cardiovascular fluid mechanics and FSI [20, 46–51], biomedical-device FSI [52–57], ship hydrodynamics with free-surface flow and fluid–object interaction [58, 59], hydrodynamics and FSI of a hydraulic arresting gear [60, 61], hydrodynamics of tidal-stream turbines with free-surface flow [62], and bioinspired FSI for marine propulsion [63, 64].

The ST-SUPS and ST-VMS have also been successfully applied to many classes of FSI, MBI, and fluid mechanics problems. The classes of problems include spacecraft parachute analysis for the main parachutes [23, 65–68], cover-separation parachutes [69] and the drogue parachutes [70–72], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [23, 33, 73, 74], full horizontal-axis wind-turbines [39, 75–77] and vertical-axis wind-turbines [5], flapping-wing aerodynamics for an actual locust [9, 23, 78, 79], bioinspired MAVs [76, 77, 80, 81] and wing-clapping [7, 82], blood flow analysis of cerebral aneurysms [76, 83], stent-blocked aneurysms [83–85], aortas [86] and heart valves [1, 7, 8, 77], spacecraft aerodynamics [69, 87], thermo-fluid analysis of ground vehicles and their tires [4], thermo-fluid analysis of disk brakes [6], flow-driven string dynamics in turbomachinery [88], flow analysis of turbocharger turbines [10, 89, 90], flow around tires with road contact and deformation [91], ram-air parachutes [92], and compressible-flow parachute aerodynamics [93].

In the tire-aerodynamics computational analysis presented here, the VMS feature of the ST-VMS addresses the challenge created by the turbulent nature of the flow, the moving-mesh feature of the ST framework enables high-resolution computation near the moving air–tire interface, and the higher-order accuracy of the ST framework strengthens both features.

1.2 ST-SI

The ST-SI was introduced in [5], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS when we have spinning

solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers. The SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the flow field. The starting point in the development of the ST-SI was the ALE-VMS version for "sliding interfaces" [30, 31]. In the ST-SI, interface terms similar to those in the ALE-VMS version are added to the ST-VMS formulation to account for the compatibility conditions for the velocity and stress. An ST-SI version where the SI is between fluid and solid domains with weakly enforced Dirichlet boundary conditions for the fluid was also presented in [5]. The SI in this case is a "fluid–solid SI" rather than a standard "fluid–fluid SI." The ST-SI method introduced in [6] for the coupled incompressible-flow and thermal-transport equations retain the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been successfully applied to aerodynamic analysis of vertical-axis wind turbines [5], thermo-fluid analysis of disk brakes [6], flow-driven string dynamics in turbomachinery [88], flow analysis of turbocharger turbines [10, 89, 90], flow around tires with road contact and deformation [91], aerodynamic analysis of ram-air parachutes [92], and flow analysis of heart valves [1].

In another version of the ST-SI method presented in [5], the SI is between a thin porous structure and the fluid on its two sides. This enables dealing with the fabric porosity in a fashion consistent with how the standard fluid–fluid SIs are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid SIs. Furthermore, this version enables handling thin structures that have T-junctions. This method has been successfully used in incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [92]. The compressible-flow ST-SI methods were introduced in [93], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [93]. These, together with the compressible-flow ST SUPG method [94], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled successful ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [93]. The computations were in the context of finite element discretization.

In the tire aerodynamics, the ST-SI enables high-resolution representation of the boundary layers near the tire. The mesh covering the tire spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution.

1.3 ST-TC

The ST-TC [7, 8] was introduced for moving-mesh computation of flow problems with TC, such as contact between solid surfaces. Even before the ST-TC, the ST-

SUPS and ST-VMS, when used with robust mesh update methods, have proven effective in flow computations where the solid surfaces are in near contact or create other near TC, if the nearness is sufficiently near for the purpose of solving the problem. Many classes of problems can be solved that way with sufficient accuracy. For examples of such computations, see the references mentioned in [7]. The ST-TC made moving-mesh computations possible even when there is an actual contact between solid surfaces or other TC. By collapsing elements as needed, without changing the connectivity of the "parent" mesh, the ST-TC can handle an actual TC while maintaining high-resolution boundary layer representation near solid surfaces. This enabled successful moving-mesh computation of heart valve flows [1, 8], wing clapping [82], and flow around a rotating tire with road contact and prescribed deformation [91].

In the tire aerodynamics, the ST-TC enables moving-mesh computation even with the TC created by the actual contact between the tire and the road. It deals with the contact while maintaining high-resolution representation near the tire.

1.4 ST-SI-TC

The ST-SI-TC is the integration of the ST-SI and ST-TC. A fluid–fluid SI requires elements on both sides of the SI. When part of an SI needs to coincide with a solid surface, which happens, for example, when the solid surfaces on two sides of an SI come into contact or when an SI reaches a solid surface, the elements between the coinciding SI part and the solid surface need to collapse with the ST-TC mechanism. The collapse switches the SI from fluid–fluid SI to fluid–solid SI. With that, an SI can be a mixture of fluid–fluid and fluid–solid SIs. With the ST-SI-TC, the elements collapse and are reborn independent of the nodes representing a solid surface. The ST-SI-TC enables high-resolution representation even when parts of the SI are coinciding with a solid surface. It also enables dealing with contact location change and contact sliding. This was used in the flow analysis of heart valves [1] and finite element flow analysis around a tire with road contact and deformation [91].

In the tire aerodynamics, the ST-SI-TC enables high-resolution representation even though parts of the SI are coinciding with the tire and road surfaces. It also enables dealing with tire–road contact location change and contact sliding.

1.5 ST-IGA

The ST-IGA was introduced in [2]. It is the integration of the ST framework with isogeometric discretization. First computations with the ST-VMS and ST-IGA were reported in [2] in a 2D context, with IGA basis functions in space for flow past

an airfoil, and in both space and time for the advection equation. The stability and accuracy analysis given [2] for the advection equation showed that using higher-order basis functions in time would be essential in getting full benefit out of using higher-order basis functions in space.

In the early stages of the ST-IGA, the emphasis was on IGA basis functions in time. As pointed out in [2, 3] and demonstrated in [9, 78, 80], higher-order NURBS basis functions in time provide a more accurate representation of the motion of the solid surfaces and a mesh motion consistent with that. They also provide more efficiency in temporal representation of the motion and deformation of the volume meshes, and better efficiency in remeshing. That is how the ST/NURBS Mesh Update Method (STNMUM) was introduced and demonstrated in [9, 78, 80]. The name "STNMUM" was given in [75]. The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS basis functions in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly, and a "secondary mapping" [2, 3, 9, 23] enables also specifying a constant angular velocity for invariant speeds along the paths. The ST framework and NURBS in time also enable, with the "ST-C" method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [4, 6, 88, 95]. The STNMUM and desirable features of the ST-IGA with IGA basis functions in time have been demonstrated in many 3D computations. The classes of problems solved are flapping-wing aerodynamics for an actual locust [9, 23, 78, 79], bioinspired MAVs [76, 77, 80, 81] and wing-clapping [7, 82], separation aerodynamics of spacecraft [69], aerodynamics of horizontal-axis [39, 75–77] and vertical-axis [5] wind-turbines, thermo-fluid analysis of ground vehicles and their tires [4], thermo-fluid analysis of disk brakes [6], flow-driven string dynamics in turbomachinery [88], and flow analysis of turbocharger turbines [10, 89, 90].

The ST-IGA with IGA basis functions in space have been utilized in ST computational flow analysis of turbocharger turbines [10, 89, 90], ram-air parachutes [92], and heart valves [1]. These computations were accomplished with the integration of the ST-IGA and ST-SI or ST-IGA, ST-SI and ST-TC.

1.6 ST-SI-TC-IGA

The turbocharger turbine analysis was based on the integration of the ST-SI and ST-IGA. The IGA basis functions were used in the spatial discretization of the fluid mechanics equations and also in the temporal representation of the rotor and spinning-mesh motion. That enabled accurate representation of the turbine surfaces and rotor motion and increased accuracy in the flow solution. The ram-air parachute analysis was based on the integration of the ST-IGA, the ST-SI version that weakly enforces the Dirichlet conditions, and the ST-SI version that accounts for the porosity of a thin structure. The ST-IGA with IGA basis functions in space

enabled, with relatively few number of unknowns, accurate representation of the parafoil geometry and increased accuracy in the flow solution. The volume mesh needed to be generated both inside and outside the parafoil. Mesh generation inside was challenging near the trailing edge because of the narrowing space. Using IGA basis functions addressed that challenge and still kept the element density near the trailing edge at a reasonable level. The heart valve analysis was based on the integration of the ST-SI, ST-TC, and ST-IGA. The ST-SI-TC-IGA, beyond enabling a more accurate representation of the surfaces and increased accuracy in the flow solution, kept the element density in the narrow spaces near the contact areas at a reasonable level. When solid surfaces come into contact, the elements between the surface and the SI collapse. Before the elements collapse, the boundaries could be curved and rather complex, and the narrow spaces might have high-aspect-ratio elements. With NURBS elements, it was possible to deal with such adverse conditions rather effectively.

An SI provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used in the flow analysis of heart valves [1] and turbocharger turbines [10, 89, 90].

In the tire aerodynamics, the ST-SI-TC-IGA enables a more accurate representation of the geometry and motion of the tire surfaces, a mesh motion consistent with that, and increased accuracy in the flow solution. It also keeps the element density in the tire grooves and in the narrow spaces near the contact areas at a reasonable level.

1.7 Tire Models

We present computations with the ST-SI-TC-IGA and two models of flow around a rotating tire with road contact and prescribed deformation. One is a simple 2D model for verification purposes, and one is a 3D model with an actual tire geometry that includes the longitudinal and transverse grooves. The tire deformation in the 3D model comes from the tire company.

1.8 Outline of the Remaining Sections

In Sect. 2 we describe the ST-VMS and ST-SI. The ST-SI-TC-IGA is described in Sect. 3. The computations with the 2D and 3D models are presented in Sects. 4 and 5, and the concluding remarks are given in Sect. 6.

2 ST-VMS and ST-SI

For completeness, in this section, we include from [1, 5, 91] the ST-VMS and ST-SI.

2.1 ST-VMS

The ST-VMS is given as

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

where

$$
\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \qquad (2)
$$

$$
r_\mathrm{C}(\mathbf{u}^h) = \nabla \cdot \mathbf{u}^h \qquad (3)
$$

are the residuals of the momentum equation and incompressibility constraint. Here, ρ, $\mathbf{u}$, p, $\mathbf{f}$, $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$, and $\mathbf{h}$ are the density, velocity, pressure, external force, stress

tensor, strain rate tensor, and the traction specified at the boundary. The test functions associated with the velocity and pressure are $\mathbf{w}$ and q. A superscript "h" indicates that the function is coming from a finite-dimensional space. The symbol Q_n represents the ST slice between time levels n and $n+1$, $(P_n)_{\mathrm{h}}$ is the part of the lateral boundary of that slice associated with the traction boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and n_{el} is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level. See [4, 5, 12, 13, 75] for the definitions used here for the stabilization parameters τ_{SUPS} and ν_{LSIC}. For more ways of calculating the stabilization parameters in finite element computation of flow problems, see [96–117].

Remark 1 The ST-SUPS can be obtained from the ST-VMS by dropping the eighth and ninth integrations.

2.2 ST-SI

2.2.1 Two-Side Formulation (Fluid–Fluid SI)

Labels "Side A" and "Side B" represent the two sides of the SI. We add boundary terms to Eq. (1). The boundary terms are first added separately for the two sides, using test functions $\mathbf{w}_{\mathrm{A}}^h$ and q_{A}^h and $\mathbf{w}_{\mathrm{B}}^h$ and q_{B}^h. Putting them together, the complete set of terms added becomes

$$
\begin{aligned}
&-\int_{(P_n)_{\mathrm{SI}}} \left(q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} - q_{\mathrm{A}}^h \mathbf{n}_{\mathrm{A}}\right) \cdot \frac{1}{2}\left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2}\left(\left(\mathscr{F}_{\mathrm{B}}^h - \left|\mathscr{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h \right. \\
&\qquad\qquad \left. -\left(\mathscr{F}_{\mathrm{B}}^h - \left|\mathscr{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{A}}^h \cdot \frac{1}{2}\left(\left(\mathscr{F}_{\mathrm{A}}^h - \left|\mathscr{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h \right. \\
&\qquad\qquad \left. -\left(\mathscr{F}_{\mathrm{A}}^h - \left|\mathscr{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h\right) \mathrm{d}P \\
&+\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{n}_{\mathrm{B}} \cdot \mathbf{w}_{\mathrm{B}}^h + \mathbf{n}_{\mathrm{A}} \cdot \mathbf{w}_{\mathrm{A}}^h\right) \frac{1}{2}\left(p_{\mathrm{B}}^h + p_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h\right) \cdot \left(\hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu\left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^h) + \boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^h)\right)\right) \mathrm{d}P
\end{aligned}
$$

$$- \gamma \int_{(P_n)_{\mathrm{SI}}} \hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu \left(\boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{B}}^h \right) + \boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{A}}^h \right) \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P$$
$$+ \int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P, \tag{4}$$

where

$$\mathscr{F}_{\mathrm{B}}^h = \mathbf{n}_{\mathrm{B}} \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{v}_{\mathrm{B}}^h \right), \tag{5}$$

$$\mathscr{F}_{\mathrm{A}}^h = \mathbf{n}_{\mathrm{A}} \cdot \left(\mathbf{u}_{\mathrm{A}}^h - \mathbf{v}_{\mathrm{A}}^h \right), \tag{6}$$

$$h = \frac{h_{\mathrm{B}} + h_{\mathrm{A}}}{2}, \tag{7}$$

$$h_{\mathrm{B}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{B}} \cdot \boldsymbol{\nabla} N_a^{\alpha} \right| \right)^{-1} \text{(for Side B)}, \tag{8}$$

$$h_{\mathrm{A}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{A}} \cdot \boldsymbol{\nabla} N_a^{\alpha} \right| \right)^{-1} \text{(for Side A)}, \tag{9}$$

$$\hat{\mathbf{n}}_{\mathrm{B}} = \frac{\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}}{\| \mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}} \|}. \tag{10}$$

Here, $(P_n)_{\mathrm{SI}}$ is the SI in the ST domain, $\mathbf{n}$ is the unit normal vector, $\mathbf{v}$ is the mesh velocity, n_{ens} and n_{ent} are the number of spatial and temporal element nodes, N_a^{α} is the basis function associated with spatial and temporal nodes a and α, $\gamma = 1$, and C is a nondimensional constant. For our element length definition, we set $C = 1$.

2.2.2 One-Side Formulation (Fluid–Solid SI)

Sometimes we prefer to specify on solid surfaces weakly imposed Dirichlet conditions for the fluid [27, 29]. In such cases we use the ST-SI version where the SI is between the fluid and solid domains. This version is obtained (see [5]) by starting with the terms added for Side B and replacing the Side A velocity with the velocity $\mathbf{g}^h$ coming from the solid domain. Then the terms added to Eq. (1) to represent the weakly imposed Dirichlet conditions become

$$- \int_{(P_n)_{\mathrm{SI}}} q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} \cdot \mathbf{u}_{\mathrm{B}}^h \mathrm{d}P - \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \mathscr{F}_{\mathrm{B}}^h \mathbf{u}_{\mathrm{B}}^h \mathrm{d}P$$
$$+ \int_{(P_n)_{\mathrm{SI}}} q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} \cdot \mathbf{g}^h \mathrm{d}P$$

$$
\begin{aligned}
&+\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2}\left(\left(\mathscr{F}_{\mathrm{B}}^h+\left|\mathscr{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h\right.\\
&\qquad\qquad\left.+\left(\mathscr{F}_{\mathrm{B}}^h-\left|\mathscr{F}_{\mathrm{B}}^h\right|\right)\mathbf{g}^h\right)\mathrm{d}P\\
&-\int_{(P_n)_{\mathrm{SI}}} \mathbf{w}_{\mathrm{B}}^h \cdot \left(\mathbf{n}_{\mathrm{B}} \cdot \boldsymbol{\sigma}_{\mathrm{B}}^h\right)\mathrm{d}P\\
&-\gamma\int_{(P_n)_{\mathrm{SI}}} \mathbf{n}_{\mathrm{B}} \cdot 2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{B}}^h\right)\cdot\left(\mathbf{u}_{\mathrm{B}}^h-\mathbf{g}^h\right)\mathrm{d}P\\
&+\int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h_{\mathrm{B}}}\mathbf{w}_{\mathrm{B}}^h \cdot \left(\mathbf{u}_{\mathrm{B}}^h-\mathbf{g}^h\right)\mathrm{d}P. \qquad (11)
\end{aligned}
$$

3 ST-SI-TC-IGA

For completeness, in this section, we include (1) from [1, 91] the aspects of the ST-SI [5] and ST-TC [7] related to their integration as the ST-SI-TC [91] and the advantages of the IGA in this context, and (2) from [1] the integration of all three components as the ST-SI-TC-IGA.

3.1 ST-SI

We note that the ST-SI allows mesh slipping also in the one-side formulation, that is, when the SI is between the fluid and solid domains (fluid–solid SI) where we have weakly imposed Dirichlet conditions. The boundary terms added to Eq. (1) to connect the two sides and to connect the fluid to the solid in the one-side formulation were given in Sects. 2.2.1 and 2.2.2. The added terms (see Eqs. (4) and (11)) include derivatives in the direction normal to the SI. Therefore the elements bordering the SI need to have finite thickness in the normal direction. This places a limitation on the meshes that can be used with the ST-SI because if an element bordering the interface degenerates it might lead to a zero element thickness in the normal direction.

3.2 ST-TC

The ST-TC can deal with TC in ST moving-mesh computations. The discretization is unstructured in time, but based on a parent mesh that is structured in time. The ST parent mesh is extruded from a single spatial mesh. The key technology is element degeneration, whenever and wherever needed, by using a special master–slave

system. This special system allows changing, within an ST slab, master nodes to slave nodes and slave nodes to master nodes. With that, elements can collapse or be reborn. This way, within an ST slab, we can represent closing and opening motions. Since an ST method naturally allows discretizations that are unstructured in time, the rest of the method needs no modification. The method is very flexible, and computationally as effective as a typical moving-mesh method. However, the master–slave relationship has to be node to node; a point on a solid surface that is not a node cannot be a master or slave node.

3.3 ST-IGA

With NURBS meshes, we can represent curved boundaries with less elements compared to finite element meshes. With this desirable feature, a volume can also be meshed with high-aspect-ratio elements. This is particularly helpful when we need to generate meshes in very narrow spaces.

3.4 ST-SI-TC-IGA

Integration of the ST-SI, ST-TC, and ST-IGA brings a number of advantages. (1) It enables high-resolution boundary layer representation near the solid surfaces in contact even when the surfaces are covered by meshes with SI. (2) It enables dealing with contact location change and contact sliding. This overcomes the ST-TC restriction that a point on a solid surface that is not a node cannot be a master or slave node. (3) When part of an SI needs to coincide with a solid surface, which happens, for example, when the solid surfaces on two sides of an SI come into contact or when an SI reaches a solid surface, the elements between the coinciding SI part and the solid surface need to collapse with the ST-TC mechanism. Before the elements collapse, the boundaries could be curved and complex, and the narrow space might have high-aspect-ratio elements. With NURBS elements, we can deal with such adverse conditions rather effectively.

Figure 1 is an example of Case (1), where we have a spinning solid surface in contact with a planar solid surface. An SI is created around the spinning surface (see Fig. 2). The SI enables the solid surface to spin together with the mesh around it. The elements collapsed with the ST-TC mechanism are in the stationary mesh on the lower side of the SI and in the spinning mesh in the contact area. Actually, the collapse decision, which is the selection between the two-side and one-side formulations, is made integration-point-wise, for each side separately, based on the element length in the normal direction, as given by Eqs. (8) and (9). For example, for Side B, the decision at an integration point is made with the following rules:

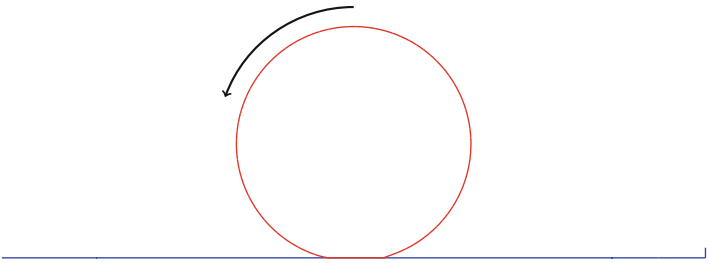


Fig. 1 A spinning solid surface (*red*) in contact with a planar solid surface (*blue*), with no slip between the surfaces

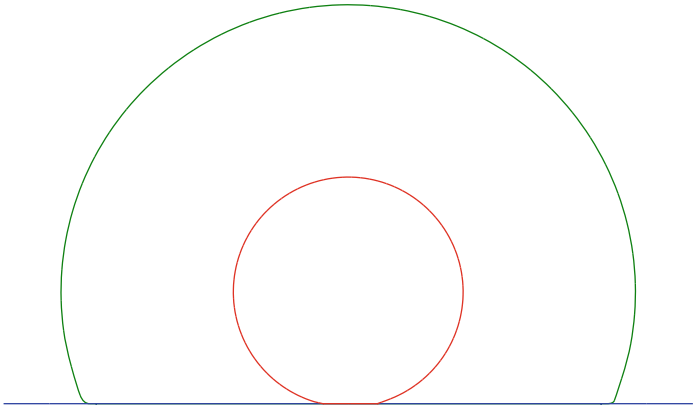


Fig. 2 A spinning solid surface (*red*) in contact with a planar solid surface (*blue*). The *green* line is the SI. It coincides with the *blue* line and the flat part of the *red* line

- If $h_{\mathrm{B}} = 0$, we disregard the integration point, regardless of the value of h_{A}.
- If $h_{\mathrm{B}} > 0$ and $h_{\mathrm{A}} = 0$, we use the one-side formulation.
- In other cases, we use the two-side formulation.

Figures 3 and 4 illustrate how the ST-SI-TC-IGA works.

Remark 2 A node on an SI coinciding with a solid surface must be a slave of the corresponding node on that solid surface.

Remark 3 When for all integration points of an element edge $h_{\mathrm{B}} = h_{\mathrm{A}} = 0$, that edge is a contact edge. Pressure is not treated as an unknown at a solid-surface master node whose all slave SI nodes live only on contact edges. That node has no role in the equation system beyond representing the geometry. Consequently, mesh resolution plays no role in regions made of only contact edges. Note that in Fig. 4, the stationary mesh in the contact area has very few elements.

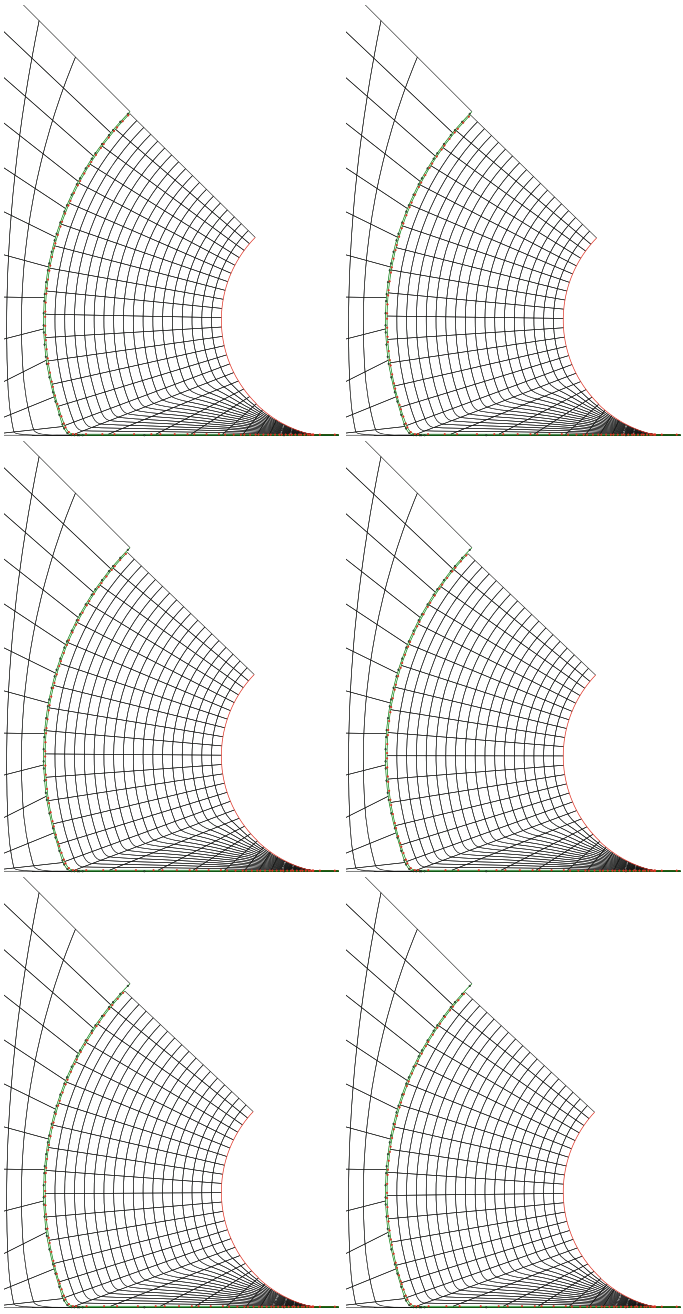


Fig. 3 Illustration of how the ST-SI-TC-IGA works for the example in Figs. 1 and 2. A part of the control mesh is shown. The *red* and *black* points are the integration points on the two sides of the SI. The outer part of the mesh is on the stationary side of the SI, and the inner part is rotating with the spinning surface (*red*). The elements collapsed with the ST-TC mechanism are in the stationary mesh on the lower side of the *green* surface and in the spinning mesh in the contact area

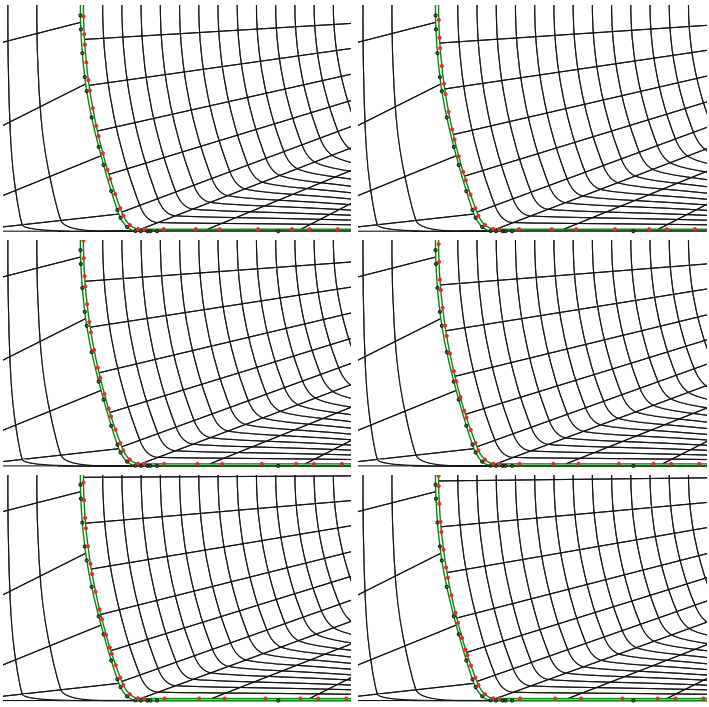


Fig. 4 The close-up view of Fig. 3, where the collapsed elements are only on the stationary side of the SI. The method switches to the one-side formulation on the part of the SI coinciding with the planar surface, and remains as the two-side formulation on the other parts

4 Verification with a Simple 2D Model

This is a problem where a nonmoving mesh can be used to obtain the solution. That will be the reference solution we will compare the ST-SI-TC-IGA solution to for verification purposes. We will also do a verification study based on comparing the solutions coming from two meshes with different refinements.

4.1 Problem Setup

A spinning solid surface is in contact with a planar solid surface and undergoes deformation. The geometry of the model and the deformation pattern are shown in Fig. 5. The rotation speed corresponds to a linear speed of $U = 100\,\text{km/h}$ at the undeformed tire periphery. There is no slip between the spinning and planar surfaces. The speed on the contact area is represented as

$$U_0 = \frac{\sin\theta}{\theta} U, \tag{12}$$

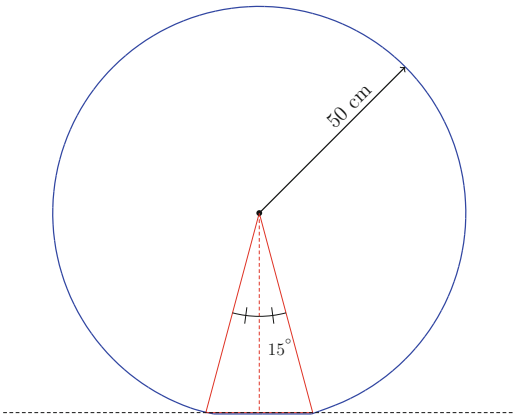


Fig. 5 A simple 2D model. The deformation region is the circular sector with central angle 30°

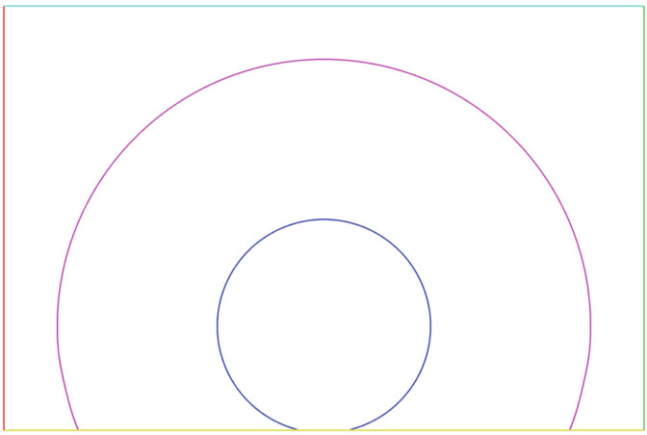


Fig. 6 A simple 2D model. Computational domain and boundary conditions. The left (*red*) and bottom (*yellow*) boundaries represent the inflow and the moving planar surface, where the velocity is U_0. The innermost (*blue*) circle is the spinning surface, where the velocity is U. The larger (*pink*) circle is the SI. The bottom of the SI is coinciding with the planar surface and the interface of the spinning and planar surfaces. The conditions at the right (*green*) and upper (*cyan*) boundaries are traction-free and slip, respectively

where $\theta = 15^\circ = 0.2618\,\mathrm{rad}$, giving $U_0 = 98.86\,\mathrm{km/h}$, which is also the speed of the planar surface. The density and kinematic viscosity of the air are $1.205\,\mathrm{kg/m^3}$ and $1.512\times10^{-5}\,\mathrm{m^2/s}$.

4.2 Computational Domain, Boundary Conditions and Meshes

Figure 6 shows the computational domain and boundary conditions. The domain size is $3.00\,\mathrm{m} \times 1.98\,\mathrm{m}$. We use two different quadratic NURBS meshes: a

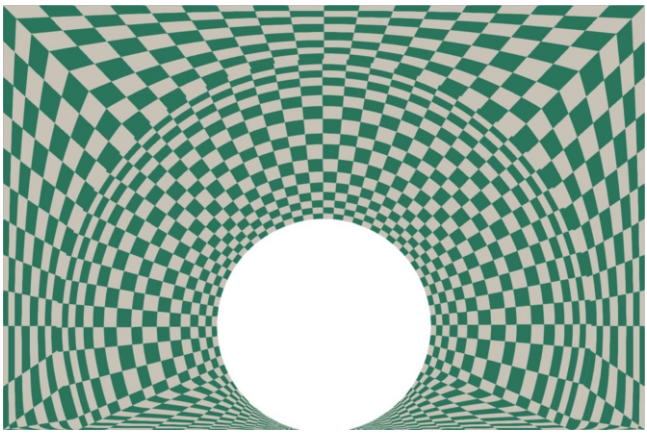


Fig. 7 A simple 2D model. Preliminary mesh. The number of control points and elements are 2204 and 1800. The checkerboard coloring is for differentiating between the NURBS elements

preliminary mesh and a refined mesh. Near the tire surface, the refined mesh has twice the resolution in both the circumferential and normal directions. Figure 7 shows the preliminary mesh. The number of control points and elements are 2204 and 1800. The SI is not needed for nonmoving-mesh computations, but we include it in computing the reference solution so that the solution with the ST-SI-TC-IGA and the reference solution come from the same mesh. Figure 8 shows the moving mesh at different instants during the computation with the ST-SI-TC-IGA. Figure 9 shows the mesh near the contact area during a period that is mostly between the 6th and 7th frames in Fig. 8. Figure 10 shows the refined mesh. The number of control points and elements are 7992 and 7200.

4.3 Computational Conditions

The computations are carried out with the ST-VMS. The stabilization parameters are those given by Eqs. (2.4)–(2.8) in [5]. In the computations with the preliminary mesh, there are 1000 time steps per rotation, which is equivalent to a time-step size of 1.131×10^{-4} s. In the refined-mesh computations, the time-step size is reduced to half of the value used with the preliminary mesh, making it 5.655×10^{-5} s. The number of nonlinear iterations per time step is 3, and the number of GMRES [118] iterations per nonlinear iteration is 300.

4.4 Results

Figures 11 and 12 show the velocity magnitude from the preliminary-mesh computations with the nonmoving-mesh (ST-SI-IGA) and ST-SI-TC-IGA methods.

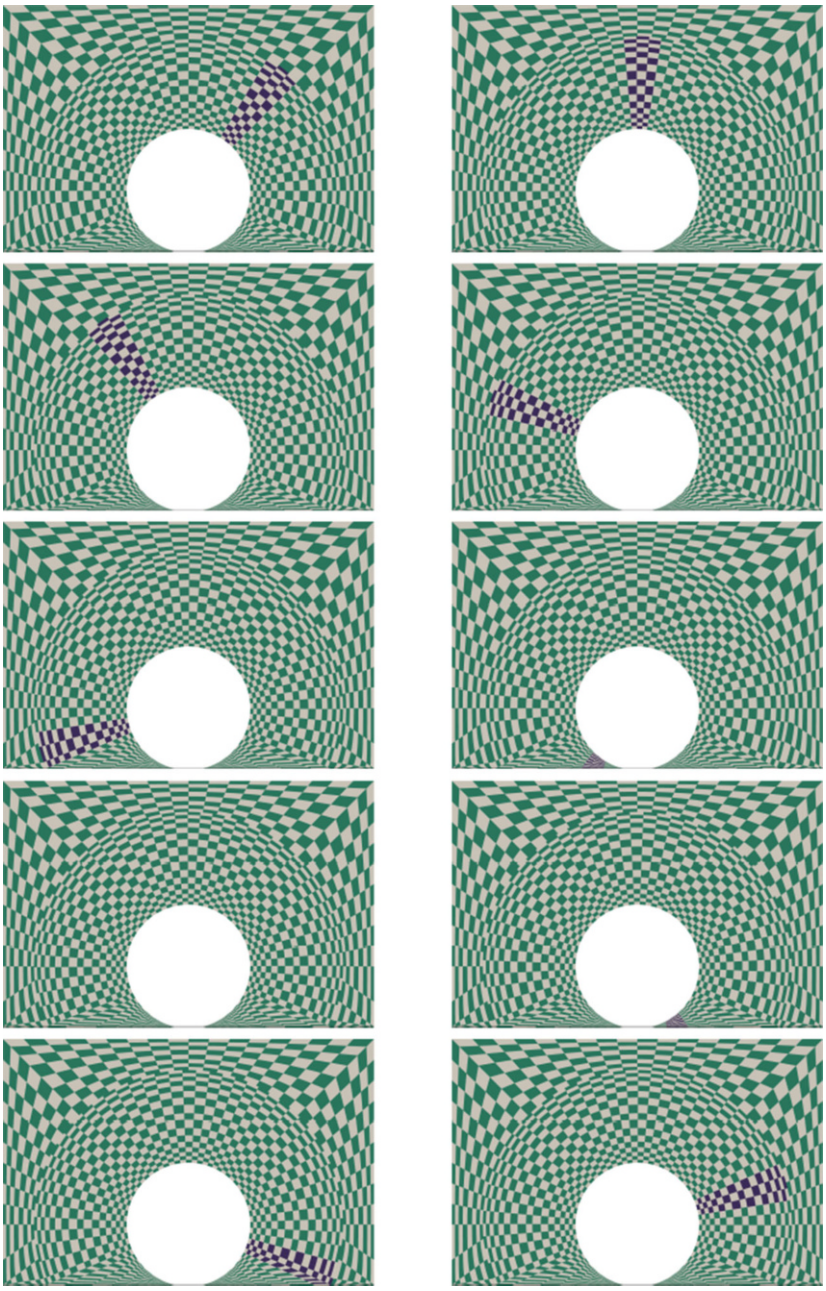


Fig. 8 A simple 2D model. Preliminary mesh at uniformly spaced instants during a one-rotation computation with the ST-SI-TC-IGA. The checkerboard coloring is for differentiating between the NURBS elements. A band of elements in the inner mesh are colored *blue* to illustrate the mesh rotation

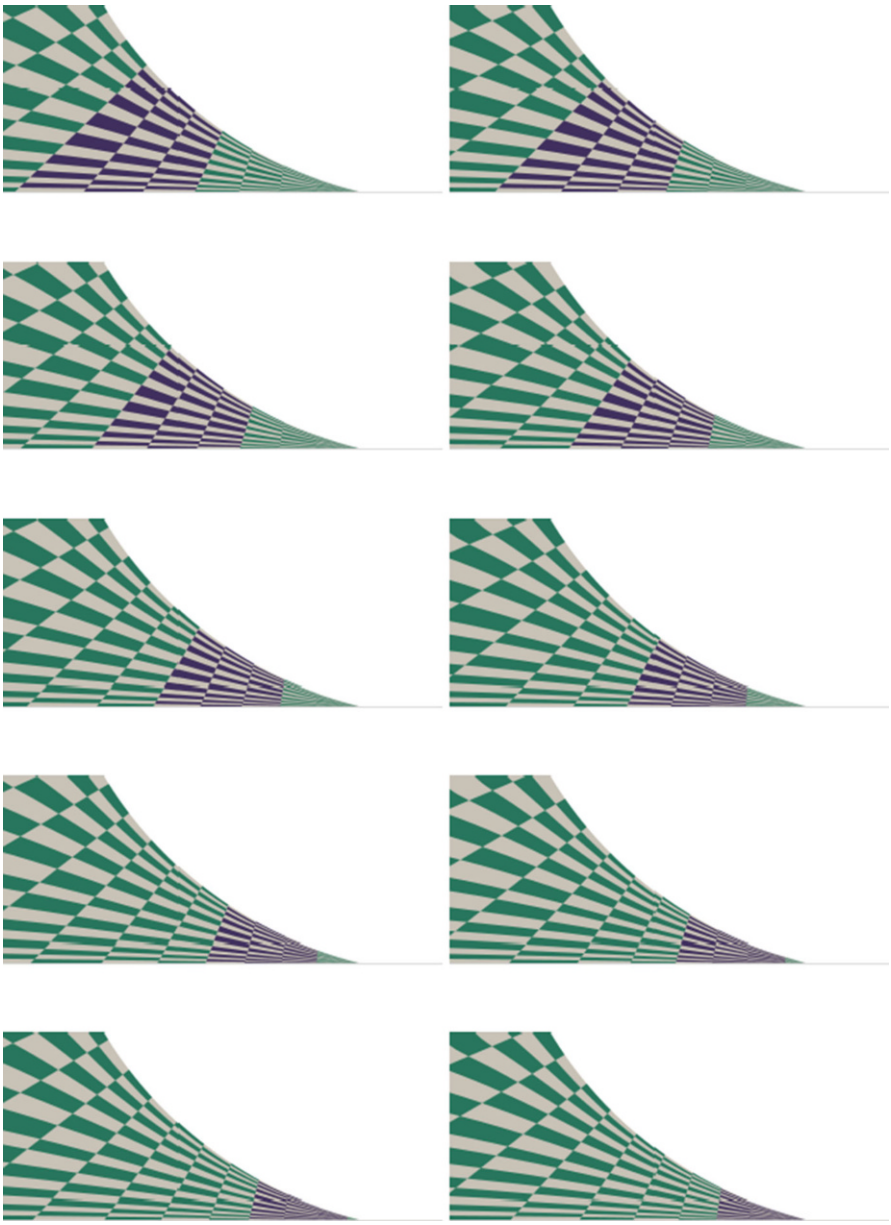


Fig. 9 A simple 2D model. Preliminary mesh near the contact area during a period that is mostly between the 6th and 7th frames in Fig. 8. The checkerboard coloring is for differentiating between the NURBS elements. A band of elements are colored *blue* to illustrate the mesh rotation

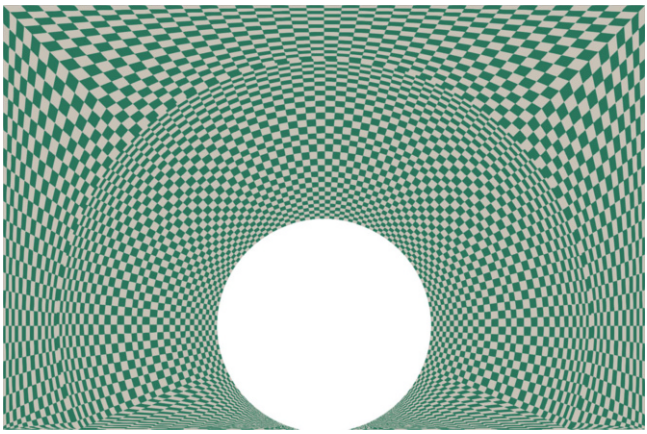


Fig. 10 A simple 2D model. Refined mesh. The number of control points and elements are 7992 and 7200. The checkerboard coloring is for differentiating between the NURBS elements

Overall, the results from the two computations are very comparable. Figure 13 shows the horizontal component of the flow velocity computed with the nonmoving-mesh (ST-SI-IGA) method, using the preliminary and refined meshes. The spinning surface generates a flow relative to the planar surface, creating boundary layers near the spinning and planar surfaces. The preliminary-mesh solution has just slightly more fluctuations than the refined-mesh solution, and we can see convergence. To compare the solutions obtained with the ST-SI-TC-IGA and nonmoving-mesh (ST-SI-IGA) methods, Figs. 14 and 15 show the horizontal component of the flow velocity computed with these two methods, using the preliminary and refined meshes. The solutions obtained with the two methods are in close agreement, indicating that the ST-SI-TC-IGA method can accurately represent the boundary layers in this class of flow problems, including the boundary layers in regions near the contact.

5 Tire Aerodynamics with an Actual Tire Geometry

We present a tire-aerodynamics computational analysis with an actual tire geometry. The tire has a prescribed motion, is in contact with the road, and has a prescribed deformation.

5.1 Problem Setup

The tire model is shown in Fig. 16. The diameter and width are 1.03 m and 260 mm. There are three longitudinal grooves, and a transverse groove for every 5°. The

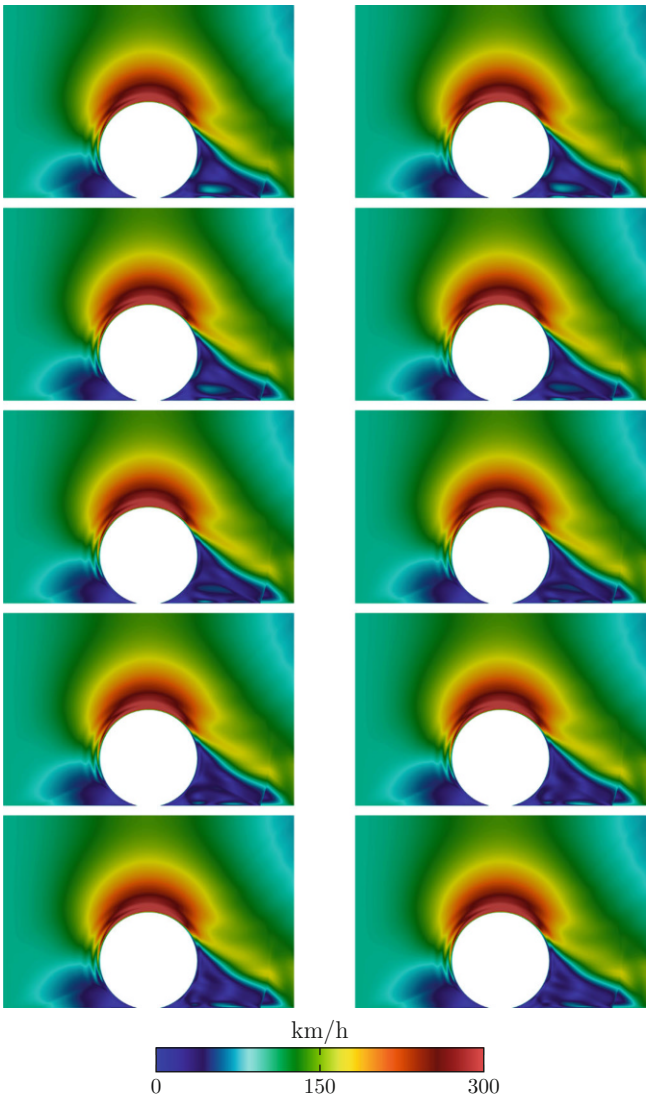


Fig. 11 A simple 2D model. Velocity magnitude from the preliminary-mesh computation with the nonmoving-mesh (ST-SI-IGA) method, at ten uniformly spaced instants during a full rotation

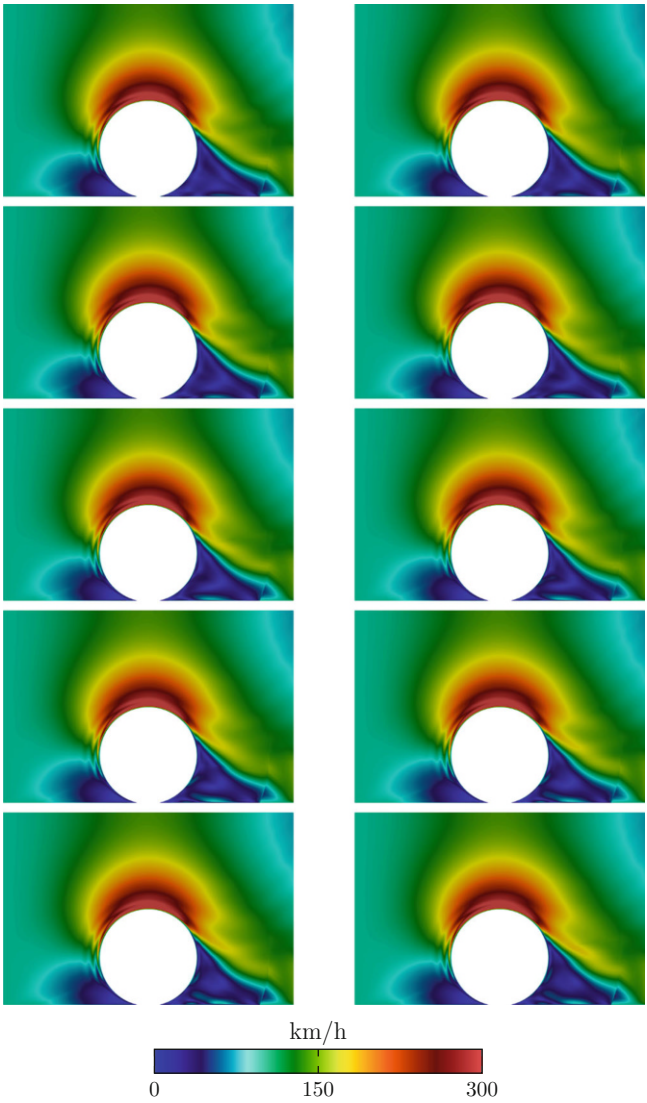


Fig. 12 A simple 2D model. Velocity magnitude from the preliminary-mesh computation with the ST-SI-TC-IGA, at the same instants as in Fig. 11

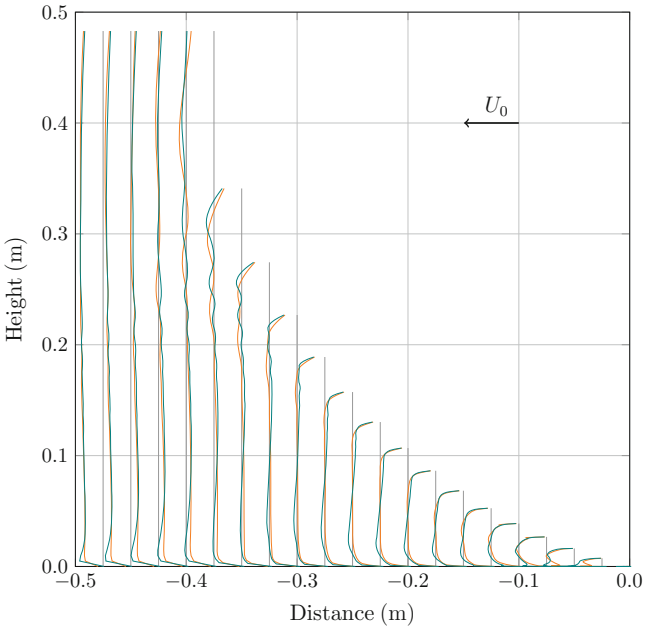


Fig. 13 A simple 2D model. Horizontal component of the flow velocity relative to the planar surface, displayed along vertical lines at different distances from the leftmost contact point. Computed with the nonmoving-mesh (ST-SI-IGA) method, using the preliminary (*orange*) and refined (*green*) meshes

depth and width of the grooves are 11.071 and 11.692 mm for the center groove, 10.974 and 7.177 mm for the side grooves, and 11.085 mm and 8.489 mm for the transverse grooves. Tire with the prescribed deformation is shown in Fig. 17. The tire deformation is represented in time based on the deformation at five instants of a 5° rotation, which was provided by the tire company. Figure 18 shows the tire deformation at those five instants. The deformation representation in time is with cubic NURBS basis functions and obtained by projection from the five-instant data. The projection is done with the ST-C [95]. The rotation speed corresponds to a linear speed of 103 km/h at the undeformed tire periphery. In this case, $U_0 = 100$ km/h. The density and kinematic viscosity of the air are $1.205\,\mathrm{kg/m^3}$ and $1.511\times10^{-5}\,\mathrm{m^2/s}$.

The computational domain is shown in Fig. 19. The domain size is 4.000 m and 5.489 m in width and height, and 8.000 m in the flow direction. The tire is placed at 2.000 m from the inflow boundary. The boundary conditions are 3D extensions of the conditions in the simple 2D model, with slip conditions on the boundary planes perpendicular to the tire axis.

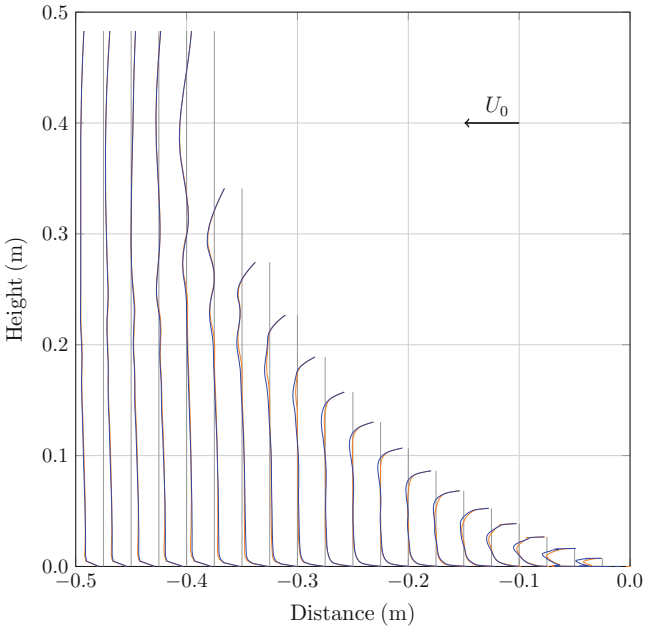


Fig. 14 A simple 2D model. Horizontal component of the flow velocity relative to the planar surface, displayed along vertical lines at different distances from the leftmost contact point. Computed with the nonmoving-mesh (ST-SI-IGA) (*orange*) and ST-SI-TC-IGA (*blue*) methods, using the preliminary mesh

5.2 Meshes

We use two different quadratic NURBS meshes: a preliminary mesh and a refined mesh. The number of control points and elements for the two meshes are given in Table 1. Figure 20 shows, for the two meshes, the refinement level near the tire surface. As can be discerned from the figure, the refined mesh has twice the resolution in the normal direction, and four times the resolution in the circumferential direction. In the axial direction, it has four times the resolution across the treads, and twice the resolution across the grooves.

5.3 Computational Conditions

In the computation with the preliminary mesh, the ST-SUPS is used, and in the computation with the refined mesh, the ST-VMS is used. The stabilization parameters are those given by Eqs. (2.4)–(2.8) in [5], without "τ_{SUGN4}" in the

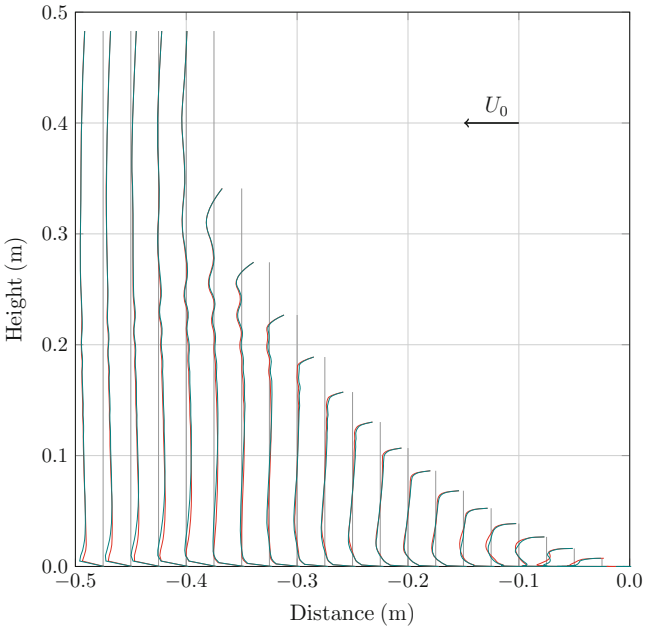


Fig. 15 A simple 2D model. Horizontal component of the flow velocity relative to the planar surface, displayed along vertical lines at different distances from the leftmost contact point. Computed with the nonmoving-mesh (ST-SI-IGA) (*green*) and ST-SI-TC-IGA (*red*) methods, using the refined mesh

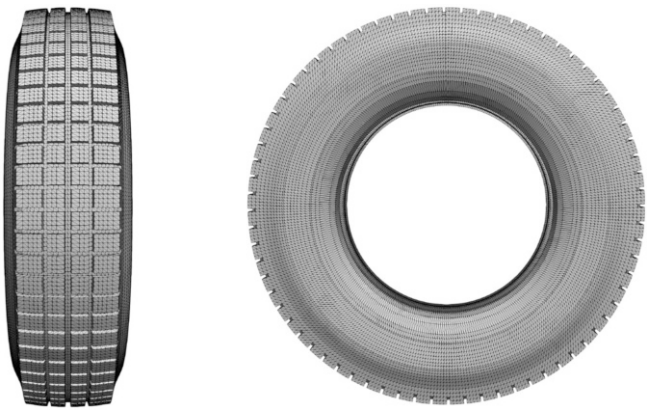


Fig. 16 Tire aerodynamics with an actual tire geometry. Tire model

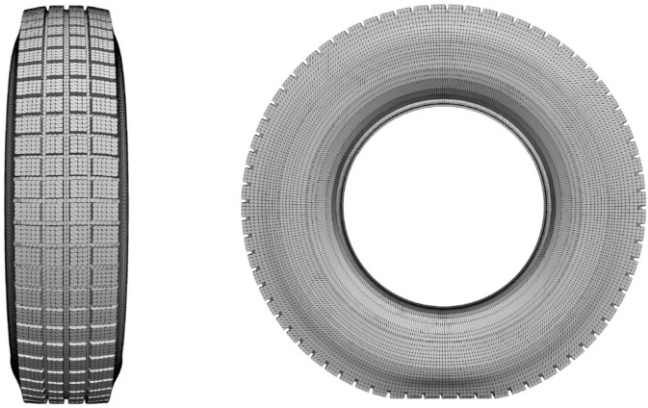


Fig. 17 Tire aerodynamics with an actual tire geometry. Deformed shape

computation with the ST-SUPS. In the computation with the preliminary mesh, there are 1440 time steps per rotation, which is equivalent to a time-step size of 7.85×10^{-5} s. In the computation with the refined mesh, the time-step size is reduced to one-fourth of the value used with the preliminary mesh, making it 1.96×10^{-5} s. The number of nonlinear iterations per time step is 3, and the number of GMRES iterations per nonlinear iteration is 300.

5.4 Results

Figures 21 and 22 show, for the two meshes, the velocity magnitude near the contact area. In the solution obtained with the preliminary mesh, the flow patterns are closer to the tire surface. Figures 23 and 24 show, for the two meshes, the isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude. The solution obtained with the refined mesh has a better resolution of the vortex structure. This confirms the importance of having a good method and high resolution in computational analysis of tire aerodynamics near contact areas.

6 Concluding Remarks

The ST-SI-TC-IGA, a new ST computational method, has enabled us to address the challenges faced in computational analysis of tire aerodynamics with actual geometry, road contact, and tire deformation. The core component of the ST-SI-TC-IGA is the ST-VMS, and the other key components are the ST-SI, ST-TC, and ST-IGA. The challenge created by the turbulent nature of the flow is addressed with

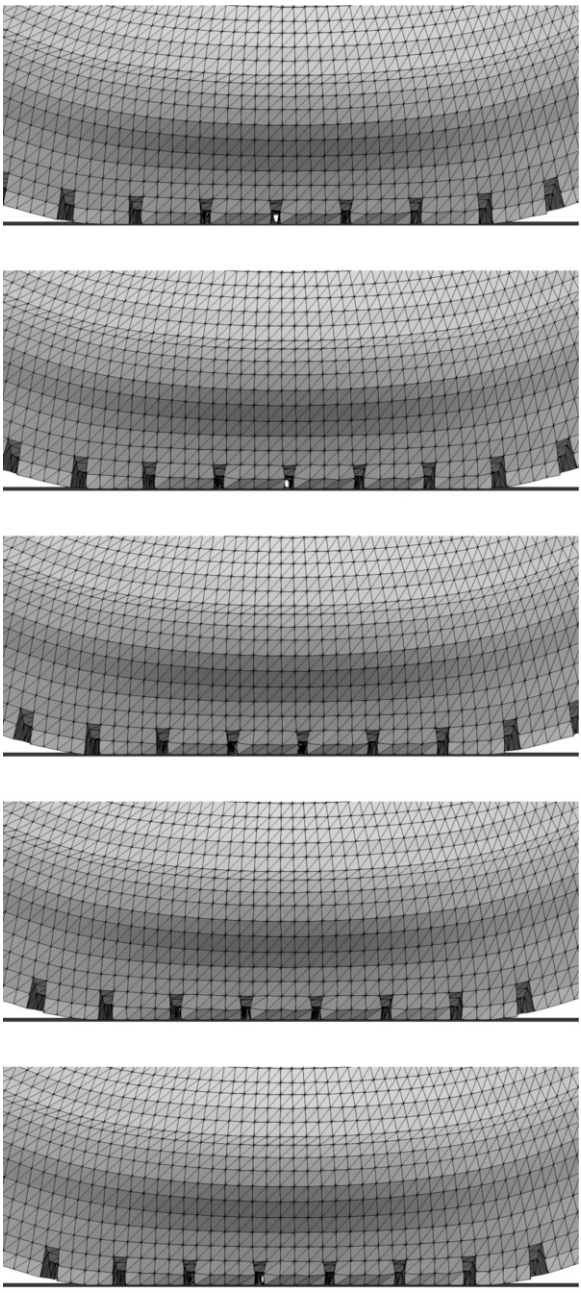


Fig. 18 Tire aerodynamics with an actual tire geometry. Tire deformation near the contact region at five instants of a 5° rotation, provided by the tire company

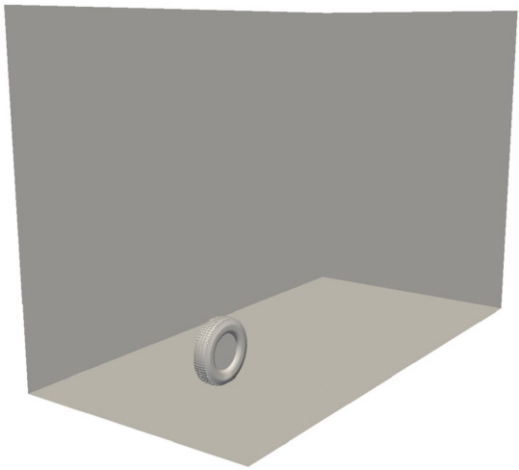


Fig. 19 Tire aerodynamics with an actual tire geometry. Computational domain

Table 1 Tire aerodynamics with an actual tire geometry

	Preliminary	Refined
nc	690,144	4,149,720
ne	376,560	2,921,552

Number of control points (*nc*) and element (*ne*) for the two quadratic NURBS meshes used in the computations

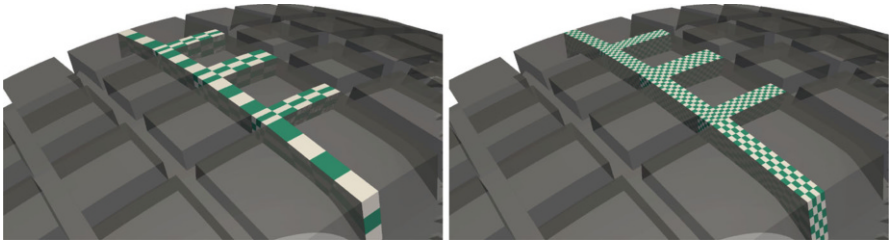


Fig. 20 Tire aerodynamics with an actual tire geometry. Refinement level near the tire surface for the preliminary (*left*) and refined (*right*) meshes. The checkerboard coloring is for differentiating between the NURBS elements

the VMS feature of the ST-VMS. The moving-mesh feature of the ST framework enables high-resolution computation near the moving air–tire interfaces. These two features are enhanced with the higher-order accuracy of the ST framework. With the ST-SI, the mesh covering the tire spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. This enables high-resolution representation of the boundary layers near the tire. With the ST-TC, we are able to do moving-mesh computations even with the TC created

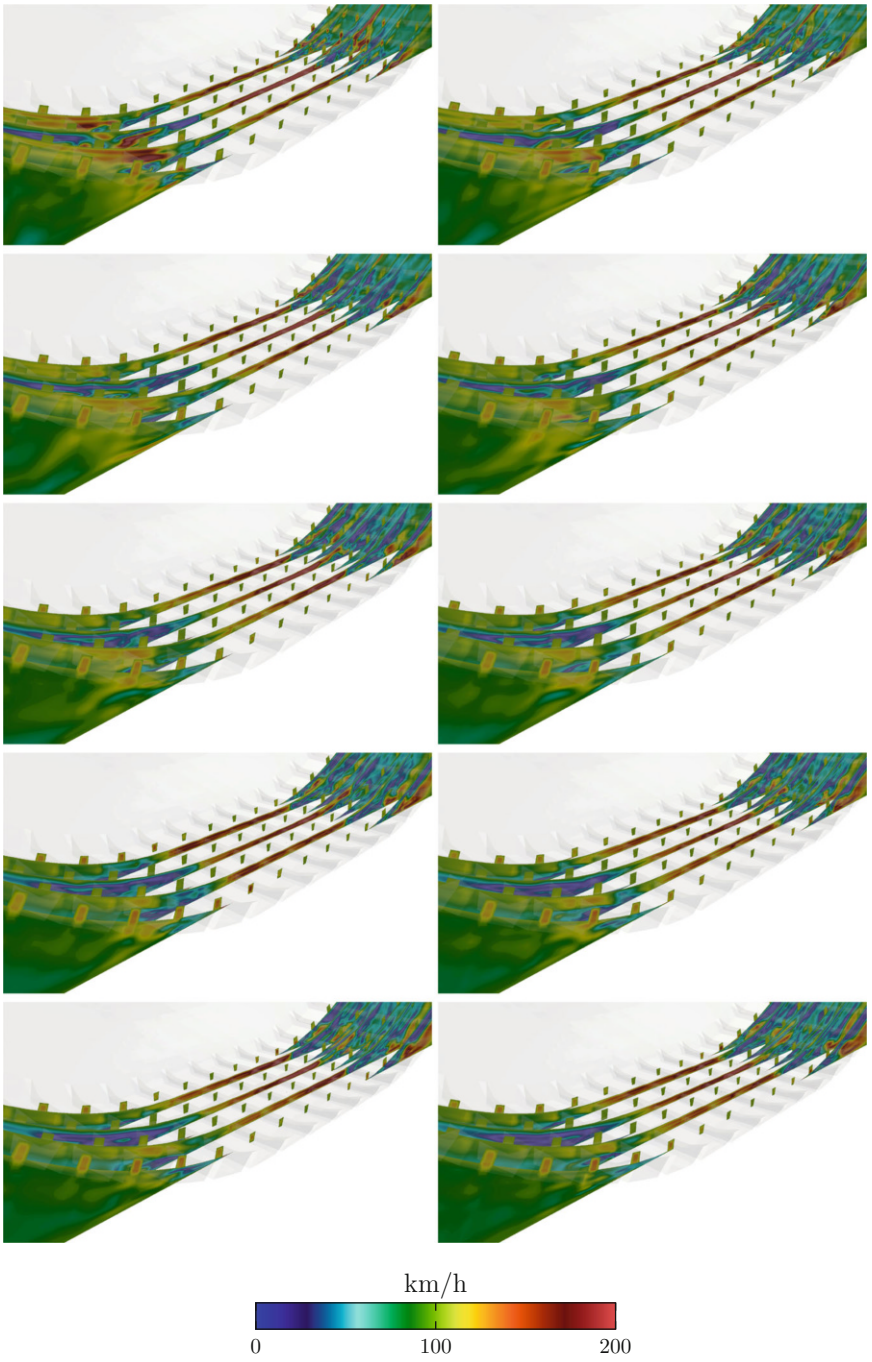


Fig. 21 Tire aerodynamics with an actual tire geometry. Computed with the preliminary mesh. Velocity magnitude near the contact area, displayed on planes perpendicular to the tire axis

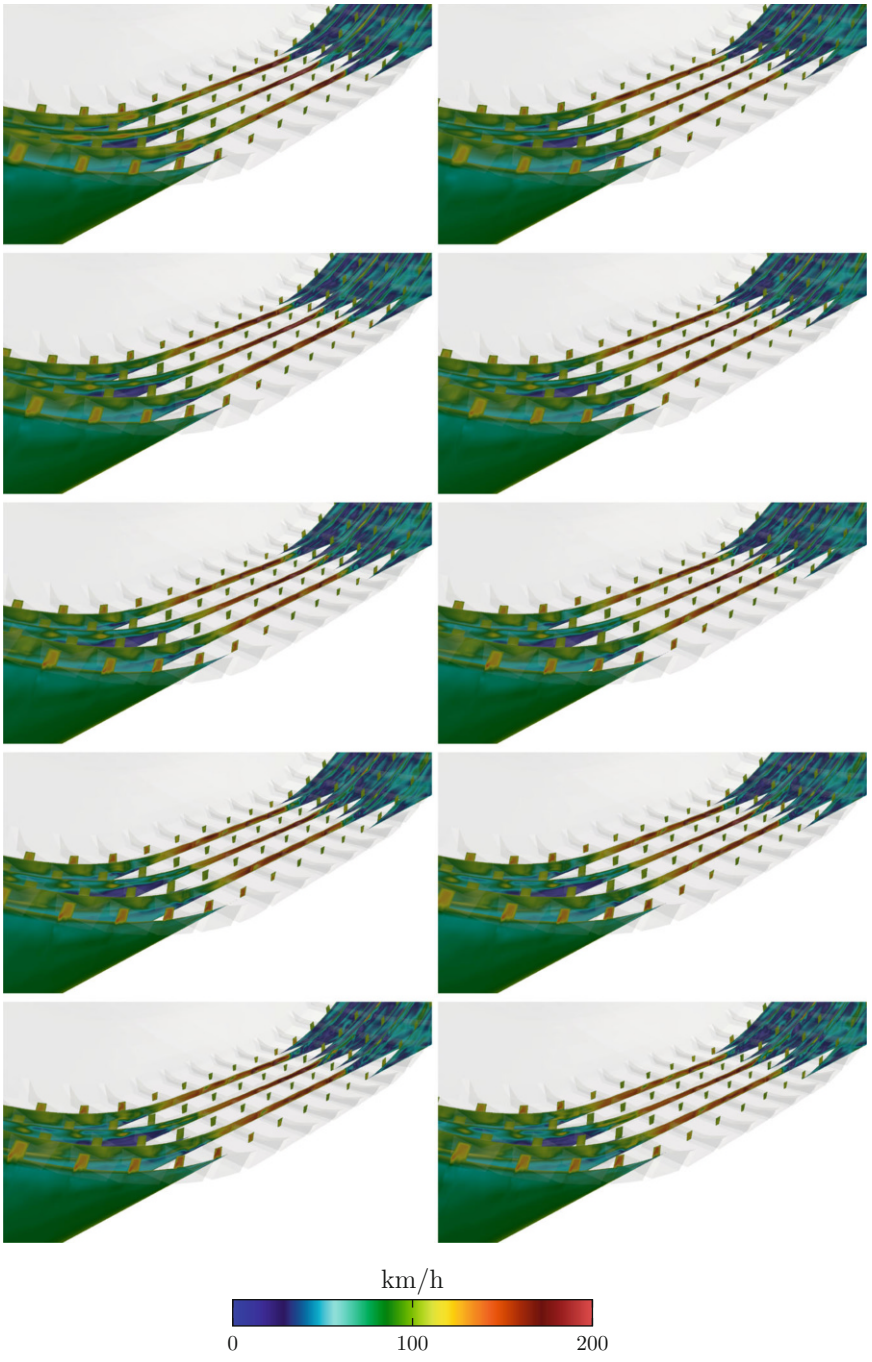


Fig. 22 Tire aerodynamics with an actual tire geometry. Computed with the refined mesh. Velocity magnitude near the contact area, displayed on planes perpendicular to the tire axis

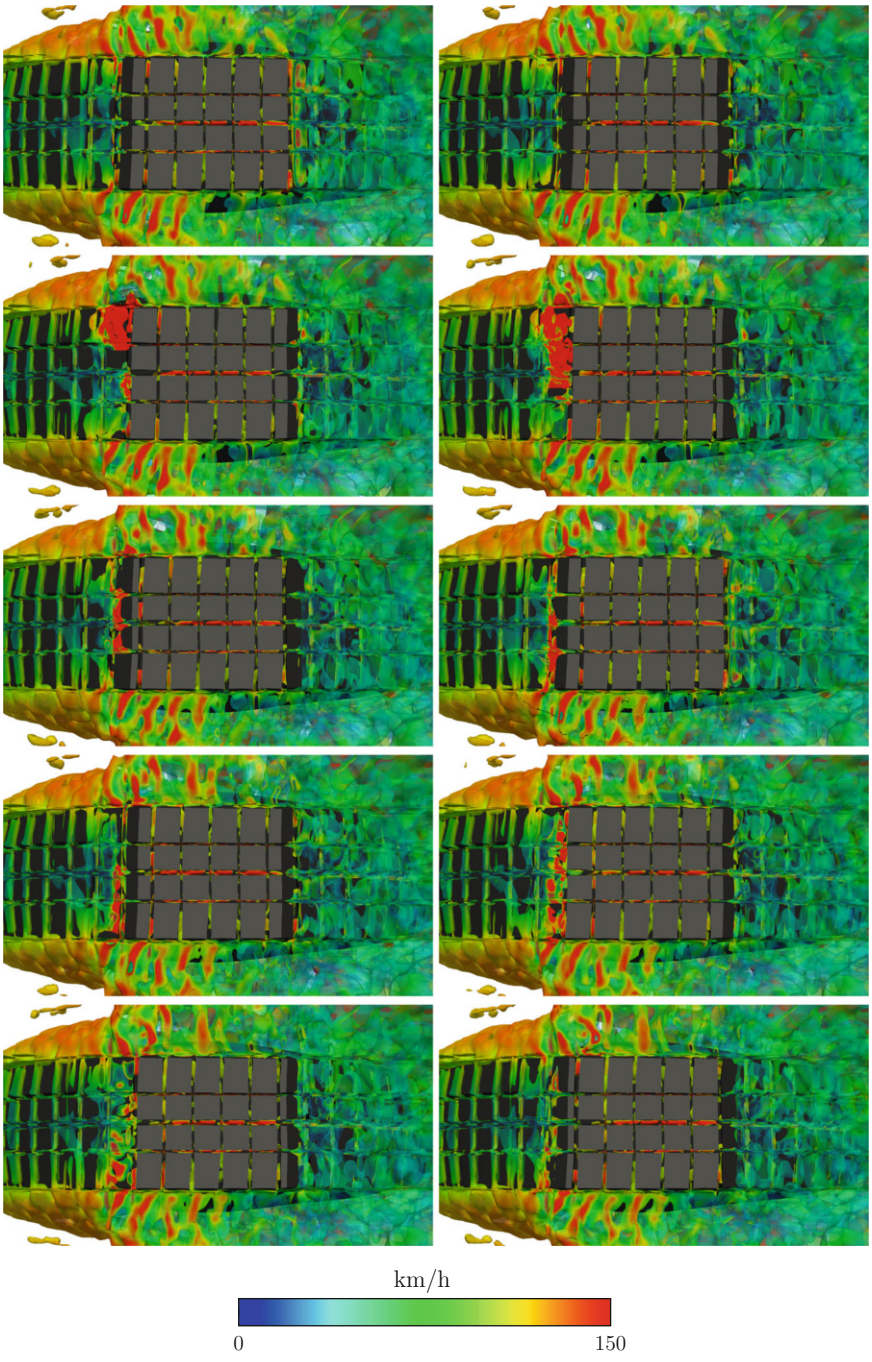


Fig. 23 Tire aerodynamics with an actual tire geometry. Computed with the preliminary mesh. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude, viewed from the bottom. The *gray* zones are the contact areas

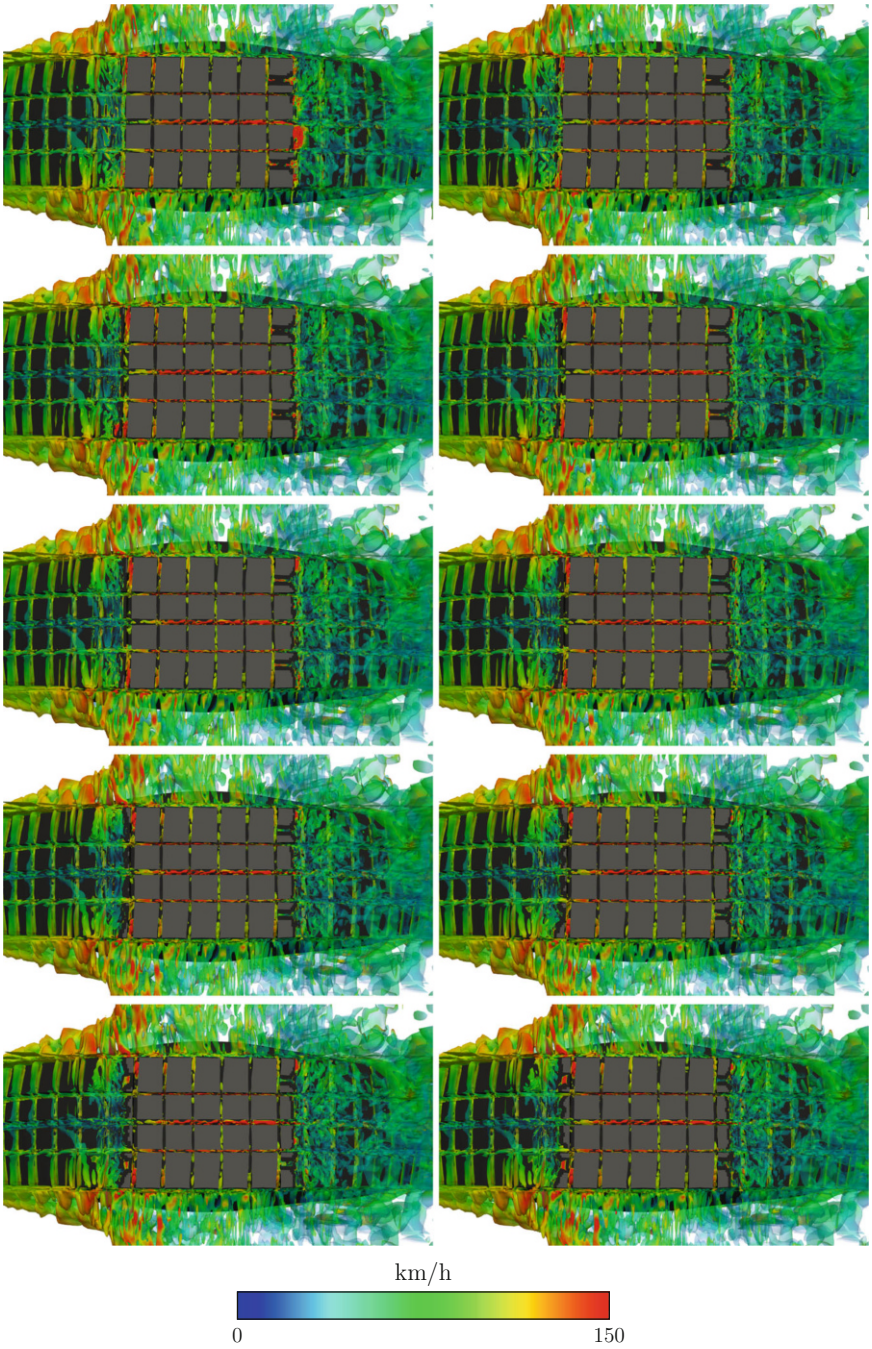


Fig. 24 Tire aerodynamics with an actual tire geometry. Computed with the refined mesh. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude, viewed from the bottom. The *gray* zones are the contact areas

by the contact between the tire and the road. This enables dealing with the contact while maintaining high-resolution representation near the tire. Integration of the ST-SI and ST-TC enables high-resolution representation even though parts of the SI are coinciding with the tire and road surfaces. It also enables dealing with the tire–road contact location change and contact sliding. Integration of the ST-IGA with the ST-SI and ST-TC not only enables a more accurate representation of the tire surfaces and increased accuracy in the flow solution, but also keeps the element density in the tire grooves and in the narrow spaces near the contact areas at a reasonable level. We presented computations with two models of flow around a rotating tire with road contact and prescribed deformation. One was a simple 2D model for verification purposes, and one was a 3D model with an actual tire geometry that included the longitudinal and transverse grooves. The 2D computations confirm the reliability of the moving-mesh and TC features of the ST-SI-TC-IGA. The 3D computations confirm the importance of having a good method and high resolution in computational analysis of tire aerodynamics near contact areas. Overall, the computations show the effectiveness of the ST-SI-TC-IGA in tire aerodynamics.

Acknowledgements This work was supported (first and second authors) in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from JSPS; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (SIP), "Innovative Combustion Technology" (Funding agency: JST); and Rice–Waseda research agreement. The computational method parts of the work were also supported (third author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University. The tire deformation used in Sect. 5 was provided by Bridgestone.

References

1. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods", *Computers & Fluids*, **158** (2017) 176–188, https://doi.org/10.1016/j.compfluid.2016.11.012.
2. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", *Computational Mechanics*, **48** (2011) 247–267, https://doi.org/10.1007/s00466-011-0571-z.
3. K. Takizawa and T.E. Tezduyar, "Space–time fluid–structure interaction methods", *Mathematical Models and Methods in Applied Sciences*, **22** (supp02) (2012) 1230001, https://doi.org/10.1142/S0218202512300013.
4. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, "Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, https://doi.org/10.1142/S0218202515400072.
5. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, "Space–time VMS method for flow computations with slip interfaces (ST-SI)", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, https://doi.org/10.1142/S0218202515400126.
6. K. Takizawa, T.E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi, "Computational thermo-fluid analysis of a disk brake", *Computational Mechanics*, **57** (2016) 965–977, https://doi.org/10.1007/s00466-016-1272-4.

7. 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, https://doi.org/10.1007/s00466-013-0935-7.
8. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, "Space–time fluid mechanics computation of heart valve models", *Computational Mechanics*, **54** (2014) 973–986, https://doi.org/10.1007/s00466-014-1046-9.
9. 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, https://doi.org/10.1115/1.4005073.
10. K. Takizawa, T.E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori, "Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA)", *Computers & Fluids*, **142** (2017) 15–20, https://doi.org/10.1016/j.compfluid.2016.02.021.
11. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44, https://doi.org/10.1016/S0065-2156(08)70153-4.
12. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, https://doi.org/10.1002/fld.505.
13. 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, https://doi.org/10.1002/fld.1430.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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, https://doi.org/10.1007/s11831-012-9071-3.
22. 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** (supp02) (2012) 1230002, https://doi.org/10.1142/S0218202512300025.
23. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
24. 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, https://doi.org/10.1142/S0218202513400010.

25. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, "New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2217–2226, https://doi.org/10.1142/S0218202515020029.
26. 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, https://doi.org/10.1016/S0045-7825(00)00204-8.
27. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", *Computers and Fluids*, **36** (2007) 12–26.
28. 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.
29. 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.
30. 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.
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. M.E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, A.L. Marsden, and M. of Congenital Hearts Alliance (MOCHA), "A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations", *Computational Mechanics*, **48** (2011) 277–291, https://doi.org/10.1007/s00466-011-0599-0.
33. 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, https://doi.org/10.1002/fld.2400.
34. 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.
35. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "High-performance computing of wind turbine aerodynamics using isogeometric analysis", *Computers and Fluids*, **49** (2011) 93–100.
36. Y. Bazilevs, M.-C. Hsu, and M.A. Scott, "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines", *Computer Methods in Applied Mechanics and Engineering*, **249–252** (2012) 28–41.
37. 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.
38. 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.
39. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre, "Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods", *Archives of Computational Methods in Engineering*, **21** (2014) 359–398, https://doi.org/10.1007/s11831-014-9119-7.
40. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines", *International Journal for Numerical Methods in Engineering*, **102** (2015) 766–783, https://doi.org/10.1002/nme.4738.
41. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Aerodynamic simulation of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2013) 021011, https://doi.org/10.1115/1.4024415.
42. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel, and J.O. Dabiri, "FSI modeling of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2014) 081006, https://doi.org/10.1115/1.4027466.

43. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs, "Computational free-surface fluid–structure interaction with application to floating offshore wind turbines", *Computers and Fluids*, **141** (2016) 155–174, https://doi.org/10.1016/j.compfluid.2016.03.008.
44. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S.M.I. Gohari, and S. Sarkar, "ALE–VMS formulation for stratified turbulent incompressible flows with applications", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2349–2375, https://doi.org/10.1142/S0218202515400114.
45. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades", *Journal of Applied Mechanics*, **83** (6) (2016) 061010.
46. 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.
47. 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.
48. 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.
49. 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.
50. 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.
51. 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.
52. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Fluid–structure interaction simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **52** (2013) 971–981, https://doi.org/10.1007/s00466-013-0858-3.
53. C.C. Long, M. Esmaily-Moghadam, A.L. Marsden, and Y. Bazilevs, "Computation of residence time in the simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **54** (2014) 911–919, https://doi.org/10.1007/s00466-013-0931-y.
54. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk", *Computational Mechanics*, **54** (2014) 921–932, https://doi.org/10.1007/s00466-013-0967-z.
55. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation", *Computational Mechanics*, **54** (2014) 1055–1071, https://doi.org/10.1007/s00466-014-1059-4.
56. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks, "Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models", *Computational Mechanics*, **55** (2015) 1211–1225, https://doi.org/10.1007/s00466-015-1166-x.
57. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 1005–1053.
58. 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.

59. 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.
60. C. Wang, M.C.H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs, "Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis", *Computers and Fluids*, **142** (2017) 3–14, https://doi.org/10.1016/j.compfluid.2015.12.004.
61. M.C.H. Wu, D. Kamensky, C. Wang, A.J. Herrema, F. Xu, M.S. Pigazzini, A. Verma, A.L. Marsden, Y. Bazilevs, and M.-C. Hsu, "Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear", *Computer Methods in Applied Mechanics and Engineering*, (2017), Published online. https://doi.org/doi:10.1016/j.cma.2016.09.032.
62. J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs, "Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines", *Computers and Fluids*, **158** (2017) 157–166, https://doi.org/10.1016/j.compfluid.2016.06.016.
63. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "Experimental and numerical FSI study of compliant hydrofoils", *Computational Mechanics*, **55** (2015) 1079–1090, https://doi.org/10.1007/s00466-014-1090-5.
64. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration", *Computers and Fluids*, **141** (2016) 201–211, https://doi.org/10.1016/j.compfluid.2015.07.013.
65. K. Takizawa and T.E. Tezduyar, "Computational methods for parachute fluid–structure interactions", *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, https://doi.org/10.1007/s11831-012-9070-4.
66. 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, https://doi.org/10.1007/s00466-012-0761-3.
67. 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, https://doi.org/10.1007/s00466-013-0880-5.
68. K. Takizawa, T.E. Tezduyar, C. Boswell, Y. Tsutsui, and K. Montel, "Special methods for aerodynamic-moment calculations from parachute FSI modeling", *Computational Mechanics*, **55** (2015) 1059–1069, https://doi.org/10.1007/s00466-014-1074-5.
69. 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, https://doi.org/10.1142/S0218202513400058.
70. K. Takizawa, T.E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel, "FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1203–1220, https://doi.org/10.1007/s00466-014-1052-y.
71. K. Takizawa, T.E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel, "Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1461–1476, https://doi.org/10.1007/s00466-014-1069-2.
72. K. Takizawa, T.E. Tezduyar, and R. Kolesar, "FSI modeling of the Orion spacecraft drogue parachutes", *Computational Mechanics*, **55** (2015) 1167–1179, https://doi.org/10.1007/s00466-014-1108-z.
73. 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, https://doi.org/10.1007/s00466-011-0589-2.
74. 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, https://doi.org/10.1007/s00466-011-0614-5.

75. 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, https://doi.org/10.1007/s00466-013-0888-x.
76. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, M.-C. Hsu, O. Øiseth, K.M. Mathisen, N. Kostov, and S. McIntyre, "Engineering analysis and design with ALE-VMS and space–time methods", *Archives of Computational Methods in Engineering*, **21** (2014) 481–508, https://doi.org/10.1007/s11831-014-9113-0.
77. K. Takizawa, "Computational engineering analysis with the new-generation space–time methods", *Computational Mechanics*, **54** (2014) 193–211, https://doi.org/10.1007/s00466-014-0999-z.
78. 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, https://doi.org/10.1007/s00466-012-0759-x.
79. 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, https://doi.org/10.1016/j.compfluid.2012.11.008.
80. 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, https://doi.org/10.1007/s00466-012-0758-y.
81. K. Takizawa, T.E. Tezduyar, and N. Kostov, "Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV", *Computational Mechanics*, **54** (2014) 213–233, https://doi.org/10.1007/s00466-014-0980-x.
82. K. Takizawa, T.E. Tezduyar, and A. Buscher, "Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping", *Computational Mechanics*, **55** (2015) 1131–1141, https://doi.org/10.1007/s00466-014-1095-0.
83. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt, "ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling", *Mathematical Models and Methods in Applied Sciences*, **24** (2014) 2437–2486, https://doi.org/10.1142/S0218202514500250.
84. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent", *Computational Mechanics*, **50** (2012) 675–686, https://doi.org/10.1007/s00466-012-0760-4.
85. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms", *Computational Mechanics*, **51** (2013) 1061–1073, https://doi.org/10.1007/s00466-012-0790-y.
86. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, and T. Ueda, "FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta", *Computational Mechanics*, **54** (2014) 1035–1045, https://doi.org/10.1007/s00466-014-1017-1.
87. K. Takizawa, D. Montes, S. McIntyre, and T.E. Tezduyar, "Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers", *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 223–248, https://doi.org/10.1142/s0218202513400022.
88. K. Takizawa, T.E. Tezduyar, and H. Hattori, "Computational analysis of flow-driven string dynamics in turbomachinery", *Computers & Fluids*, **142** (2017) 109–117, https://doi.org/10.1016/j.compfluid.2016.02.019.
89. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method", *Computers & Fluids*, **158** (2017) 189–200, https://doi.org/10.1016/j.compfluid.2017.04.017.
90. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "A general-purpose NURBS mesh generation method for complex geometries", to appear in a special volume to be published by Springer, 2018.
91. K. Takizawa, T.E. Tezduyar, S. Asada, and T. Kuraishi, "Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC)", *Computers & Fluids*, **141** (2016) 124–134, https://doi.org/10.1016/j.compfluid.2016.05.006.

92. K. Takizawa, T.E. Tezduyar, and T. Terahara, "Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)", *Computers & Fluids*, **141** (2016) 191–200, https://doi.org/10.1016/j.compfluid.2016.05.027.
93. K. Takizawa, T.E. Tezduyar, and T. Kanai, "Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity", *Mathematical Models and Methods in Applied Sciences*, **27** (2017) 771–806, https://doi.org/10.1142/S0218202517500166.
94. 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, https://doi.org/10.1016/0045-7825(94)00082-4.
95. K. Takizawa and T.E. Tezduyar, "Space–time computation techniques with continuous representation in time (ST-C)", *Computational Mechanics*, **53** (2014) 91–99, https://doi.org/10.1007/s00466-013-0895-y.
96. T.E. Tezduyar and D.K. Ganjoo, "Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion problems", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 49–71, https://doi.org/10.1016/0045-7825(86)90023-X.
97. G.J. Le Beau, S.E. Ray, S.K. Aliabadi, and T.E. Tezduyar, "SUPG finite element computation of compressible flows with the entropy and conservation variables formulations", *Computer Methods in Applied Mechanics and Engineering*, **104** (1993) 397–422, https://doi.org/10.1016/0045-7825(93)90033-T.
98. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, https://doi.org/10.1016/j.compfluid.2005.02.011.
99. 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, https://doi.org/10.1016/j.cma.2005.05.032.
100. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", *Computers & Fluids*, **36** (2007) 147–159, https://doi.org/10.1016/j.compfluid.2005.07.009.
101. 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, https://doi.org/10.1007/s00466-005-0025-6.
102. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, https://doi.org/10.1007/s00466-006-0056-7.
103. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364, https://doi.org/10.1007/s00466-006-0045-x.
104. 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, https://doi.org/10.1016/j.compfluid.2005.07.004.
105. 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, https://doi.org/10.1002/fld.1743.
106. 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, https://doi.org/10.1002/fld.1447.
107. 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, https://doi.org/10.1002/fld.1484.

108. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms", *Journal of Applied Mechanics*, **76** (2009) 021211, https://doi.org/10.1115/1.3062967.
109. 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, https://doi.org/10.1115/1.3057496.
110. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167, https://doi.org/10.1007/s00466-009-0441-0.
111. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "Improving stability of stabilized and multiscale formulations in flow simulations at small time steps", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840, https://doi.org/10.1016/j.cma.2009.06.019.
112. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270, https://doi.org/10.1002/fld.2451.
113. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique", *Journal of Applied Mechanics*, **79** (2012) 010910, https://doi.org/10.1115/1.4005060.
114. A. Corsini, F. Rispoli, A.G. Sheard, and T.E. Tezduyar, "Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations", *Computational Mechanics*, **50** (2012) 695–705, https://doi.org/10.1007/s00466-012-0789-4.
115. P.A. Kler, L.D. Dalcin, R.R. Paz, and T.E. Tezduyar, "SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems", *Computational Mechanics*, **51** (2013) 171–185, https://doi.org/10.1007/s00466-012-0712-z.
116. A. Corsini, F. Rispoli, A.G. Sheard, K. Takizawa, T.E. Tezduyar, and P. Venturini, "A variational multiscale method for particle-cloud tracking in turbomachinery flows", *Computational Mechanics*, **54** (2014) 1191–1202, https://doi.org/10.1007/s00466-014-1050-0.
117. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T.E. Tezduyar, "Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing", *Computational Mechanics*, **55** (2015) 1201–1209, https://doi.org/10.1007/s00466-015-1160-3.
118. Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM Journal of Scientific and Statistical Computing*, **7** (1986) 856–869.

Thermal Convection in the van der Waals Fluid

Ju Liu

Abstract In this work, the van der Waals fluid model, a diffuse-interface model for liquid–vapor two-phase flows, is numerically investigated. The thermodynamic properties of the van der Waals fluid are first studied. Dimensional analysis is performed to identify the control parameters for the system. An entropy-stable numerical scheme and isogeometric analysis are utilized to discretize the governing equations for numerical simulations. The steady state solution at low Rayleigh number is presented, demonstrating the capability of the model in describing liquid–vapor phase transitions. Next, two-dimensional nucleate and film boiling are simulated, showing the applicability of the model in different regimes of boiling. In the last, the heat transport property of the van der Waals model is numerically investigated. The scaling law for the Nusselt number with respect to the Rayleigh number in the van der Waals model is obtained by performing a suite of high-resolution simulations.

1 Introduction

The Rayleigh-Bénard convection describes the buoyancy driven flow confined by a hot bottom plate and a cold top plate. It is a canonical model for studying hydrodynamic stability [6] and thermally driven turbulence [2]. The bulk transport properties for the Rayleigh-Bénard convective flow is mysterious and still attracts a tremendous amount of scientific interests world-widely. Traditionally, the Oberbeck-Boussinesq (OB) approximation for buoyancy has been utilized as

J. Liu (✉)
Department of Pediatrics, Stanford University, Stanford, CA, USA
e-mail: liuju@stanford.edu

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_9

a standard mathematical model to investigate the Rayleigh-Bénard instability [8]. In the OB approximation, it is assumed that the fluid density varies linearly with the temperature field, the flow is incompressible, and the heat produced by internal friction is negligible. Based on the OB approximation, various scaling theories have been developed to account for the relationship between the bulk fluid transport properties (e.g., the Nusselt number Nu, the Reynolds number Re, etc.) and the strength of the buoyant force, which is measured by the Rayleigh number Ra. Theoretical, experimental, and numerical studies suggest that $\mathrm{Nu} \propto \mathrm{Ra}^{\gamma_{\mathrm{Nu}}}$ and $\mathrm{Re} \propto \mathrm{Ra}^{\gamma_{\mathrm{Re}}}$ [2]. Currently, the precise values of γ_{Nu} and γ_{Re} at different fluid regimes still remain under debate, and the existence of the theoretically predicted "ultimate state" at high Rayleigh number still awaits experimental and numerical confirmation [26].

Recent research activity has moved beyond the classical Oberbeck–Boussinesq model [29], and investigations of the non-Oberbeck–Boussinesq (NOB) effects have been carried out [1, 23]. In more practical situations, fluid properties can be temperature dependent [27]; fluids may experience compressibility effects or even phase transition. In the literature, numerical investigations of the convection in ideal fluids and super–critical fluids have been performed [1, 11, 28]. A recent research work has investigated the convection in multiphase fluids [29]. In [17, 23], researchers modified the classical OB model by introducing point sources in the balance equations to model the behavior of bubbles in boiling flows. This approach necessitates empirical knowledge of several physical coefficients related to the bubbles. The lattice Boltzmann model has been utilized in combination with non-ideal gas models to study heat convection in two-phase flows [5, 7, 25].

In recent years, phase-field models are introduced as an effective modeling technique for interfacial physics. It uses an order parameter to distinguish different phases and postulates that the interface has finite width and material properties transit across the interfacial region smoothly but sharply. Traditional interface-capturing and interface-tracking methods are designed based on geometrical information of existing interfaces. When dealing with problems with phase transition, those methods become intractable. The phase-field models enjoy solid thermodynamic foundations [19], and this property allows them to describe complicated phase transition phenomena without resorting to ad hoc modeling tricks. The first instantiation of the phase-field models emanates from the work of van der Waals [24] and Korteweg [16]. It is now commonly known as the van der Waals fluid theory or the Navier–Stokes–Korteweg equations. In recent years, the van der Waals model was systematically analyzed in rational mechanics framework and the choice of the constitutive relations was carefully made to guarantee fundamental thermomechanical principles [9, 20, 21]. Recently progress has been made in boiling simulation using the Navier–Stokes–Korteweg equations. Boiling is regarded to be highly challenging for numerical simulations [15, 17]. It involves several physical mechanisms in multiple spatial-to-temporal scales. Most of the mechanisms are still not well understood quantitatively. Due to its improved approximation property for real fluids and its simplicity in describing boiling phenomena, it is appealing to investigate the free convection for the van der Waals fluids.

In this work, I will show that the van der Waals fluid model gives an accurate description of gas and liquid phases for various single-component fluids over a wide range of temperature. The satisfaction of the Clausius–Clapeyron relation demonstrates the model's capability in describing phase transition phenomena. The van der Waals model is incorporated into the continuum mechanics equations, and the resulting governing equations constitute a set of partial differential equations, which involve a third-order differential operator. The solid thermodynamic foundation allows this set of partial differential equations to describe bubble generation, phase transition, and topology change of interfaces in a unified approach. The numerical method for solving these equations is based on a provably unconditionally stable, second-order accurate scheme [21]. Isogeometric analysis [13] is invoked to provide high-resolution spatial discretization. With the numerical method, I will first study the solutions of the Navier–Stokes–Korteweg equations at a small Rayleigh number. The steady state solution of the system at different volumetric averaged densities are solved and discussed. It will be shown that large volumetric averaged densities lead to stratified single-phase fluid as the steady state solution. In contrast, liquid–vapor two-phase fluid can be generated at low volumetric averaged densities. Next, two numerical simulations are performed to study the capability of the model in describing nucleate and film boiling. It will be shown that low fluid viscosity results in nucleate boiling while high fluid viscosity leads to film boiling. In the last, a suite of eighteen numerical simulations is performed to study the thermal convection property in the van der Waals fluid at volumetric averaged density 0.8.

2 Model

2.1 The van der Waals Model

In this work, a fixed, bounded, and connected domain $\Omega \subset \mathbb{R}^{n_d}$ is considered, where n_d represents the number of space dimensions. The time interval of interest is denoted as $(0, T)$, with $T > 0$. The Navier–Stokes–Korteweg equations are posed in the space-time domain $\Omega \times (0, T)$ as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\tau} - \nabla \cdot \boldsymbol{\varsigma} = \rho \mathbf{g}, \tag{2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)\mathbf{u} - (\boldsymbol{\tau} + \boldsymbol{\varsigma})\mathbf{u}) + \nabla \cdot \mathbf{q} + \nabla \cdot \boldsymbol{\Pi} = \rho \mathbf{g} \cdot \mathbf{u} + \rho r. \tag{3}$$

Equations (1)–(3) describe the balance of mass, linear momentum, and energy, respectively. In these equations, ρ is the fluid density, $\mathbf{u}$ is the velocity, p is the thermodynamic pressure, $\boldsymbol{\tau}$ is the viscous stress, $\boldsymbol{\varsigma}$ is the Korteweg stress, $\mathbf{g}$ is the

gravity, E is the total energy per unit mass, $\mathbf{q}$ is the heat flux, and $\boldsymbol{\Pi}$ represents the power expenditure due to phase transitions [12, 21]. The constitutive relations for the van der Waals fluid are given as follows.

$$\boldsymbol{\tau} = \bar{\mu}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T - \frac{2}{3}\nabla\cdot\mathbf{u}\mathbf{I}\right), \tag{4}$$

$$\boldsymbol{\varsigma} = \left(\lambda\rho\Delta\rho + \frac{\lambda}{2}|\nabla\rho|^2\right)\mathbf{I} - \lambda\nabla\rho\otimes\nabla\rho, \tag{5}$$

$$p = Rb\theta\frac{\rho}{b-\rho} - a\rho^2, \tag{6}$$

$$\mathbf{q} = -\kappa\nabla\theta, \tag{7}$$

$$\boldsymbol{\Pi} = \lambda\rho\nabla\cdot\mathbf{u}\nabla\rho, \tag{8}$$

$$E = \iota + \frac{1}{2}|\mathbf{u}|^2, \tag{9}$$

$$\iota = \iota_{loc} + \frac{\lambda}{2\rho}|\nabla\rho|^2, \tag{10}$$

$$\iota_{loc} = -a\rho + C_v\theta. \tag{11}$$

In the above, $\bar{\mu}$ is the dynamic viscosity, λ is the capillarity coefficient, R is the specific gas constant, a and b are fundamental fluid properties whose values for typical fluids can be found in [14], κ is the thermal conductivity, C_v is the specific heat capacity at constant volume, ι is the internal energy per unit mass, and ι_{loc} is the local part of the internal energy density.

Before proceeding further, let us introduce six additional thermodynamic quantities. First, the isobaric thermal expansion coefficient β is defined as

$$\beta := -\frac{1}{\rho}\frac{\partial\rho}{\partial\theta}. \tag{12}$$

Taking partial derivative with respect to θ at both sides of (6), one can readily obtain an explicit expression of β for the van der Waals fluid as

$$\beta = \frac{Rb(b-\rho)}{Rb^2\theta - 2a\rho(b-\rho)^2}. \tag{13}$$

Second, the heat capacity at constant pressure is defined as

$$C_p := \frac{\partial\iota_{loc}}{\partial\theta} - \frac{p}{\rho^2}\frac{\partial\rho}{\partial\theta}. \tag{14}$$

Invoking the constitutive relations (6) and (11), one has

$$C_p := C_v + \frac{Rb\theta}{b-\rho}\beta. \tag{15}$$

It is known that C_v can be expressed as

$$C_v = \varpi R, \tag{16}$$

wherein the non-dimensional number ϖ depends on the structure of the fluid molecules [18]. Hence one may further rewrite C_p as

$$C_p = (\varpi + \frac{b\theta\beta}{b-\rho})R = (\varpi + \chi)R, \tag{17}$$

$$\chi := \frac{b\theta\beta}{b-\rho} = \frac{Rb^2\theta}{Rb^2\theta - 2a\rho(b-\rho)^2}. \tag{18}$$

Third, the local electrochemical potential is defined as

$$\nu_{loc} := -2a\rho + R\theta \log\left(\frac{\rho}{b-\rho}\right) - C_v\theta\left(\log\left(\frac{\theta}{\theta_{\text{ref}}}\right) - 1\right) + \frac{Rb\theta}{b-\rho}, \tag{19}$$

wherein $\theta_{\text{ref}} > 0$ is the reference temperature. Fourth, the entropy for the van der Waals fluid is defined as

$$s := -R\log\left(\frac{\rho}{b-\rho}\right) + C_v \log\left(\frac{\theta}{\theta_{\text{ref}}}\right). \tag{20}$$

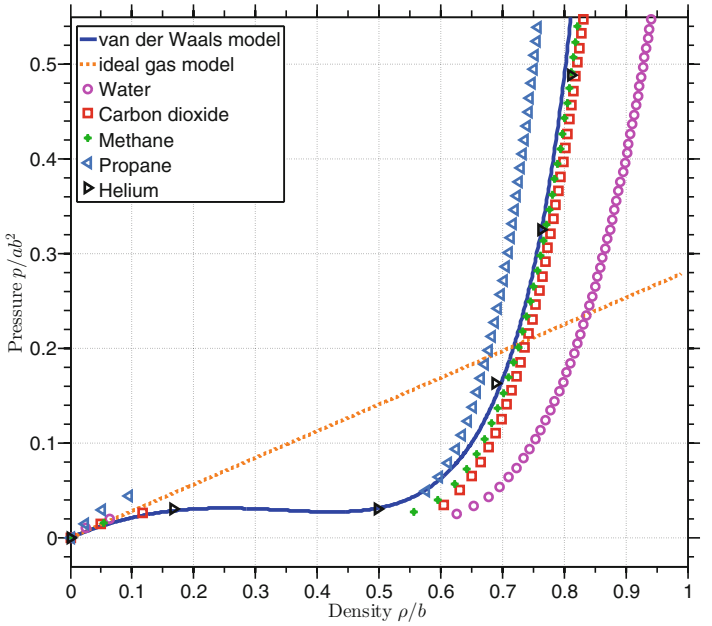
In the last, the kinematic viscosity $\bar{\nu}$ is defined as $\bar{\nu} := \bar{\mu}/\rho$. The thermal diffusivity α is defined as $\alpha := \kappa/C_p\rho$.

Remark 1 In the limit of $\rho \to 0$, the relations (13) and (15) lead to

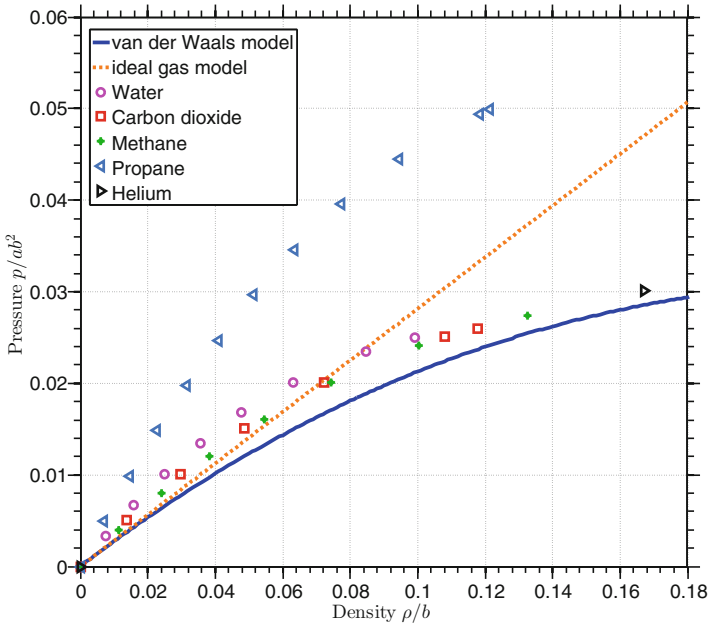
$$\beta \to \frac{1}{\theta}, \qquad C_p \to C_v + R.$$

These two relations recover the thermal expansion coefficient and the Mayer's relation for ideal gases [18].

The van der Waals fluid model is regarded as a good approximation for fluids in both vapor and liquid phases. In Fig. 1, the van der Waals equation of state (6) is plotted as a uni-variate function of density at 0.95 of the critical temperature. The ideal gas model and real fluid data [10] are plotted in the same figure for comparison purpose. It can be observed that the val der Waals model gives a very accurate description for various different fluids in both vapor and liquid phases. It is worth mentioning that the approximation is particularly excellent for helium, which is widely used as the working fluid in Rayleigh-Bénard experiments [4]. The good approximation attribute makes the van der Waals model an ideal candidate for studying the Rayleigh-Bénard convection for liquid–vapor two-phase fluids.



(a)



(b)

Fig. 1 (**a**) Comparison of the van der Waals model and the ideal gas model with real fluids at temperature $\theta = 0.95\theta_0$, $\theta_0 = 8ab/27R$. The data for water, carbon dioxide, methane, propane, and helium are obtained from [10] and scaled to dimensionless form. (**b**) gives a detailed view in the vapor phase

2.2 Dimensional Analysis

In this section, dimensional analysis is performed for the system of equations (1)–(3) using M_0, L_0, T_0, and θ_0 as the reference scales of mass, length, time, and temperature. If the reference scales are chosen as

$$\frac{M_0}{L_0^3} = b, \quad \frac{M_0}{L_0 T_0^2} = ab^2, \quad \theta_0 = \frac{8ab}{27R}, \tag{21}$$

and θ_{ref} is selected as $\theta_{\text{ref}} = \theta_0$, the dimensionless system can be written as

$$\frac{\partial \rho^*}{\partial t^*} + \nabla^* \cdot \left(\rho^* \mathbf{u}^*\right) = 0, \tag{22}$$

$$\frac{\partial (\rho^* \mathbf{u}^*)}{\partial t^*} + \nabla^* \cdot (\rho^* \mathbf{u}^* \otimes \mathbf{u}^*) + \nabla^* p^* - \nabla^* \cdot \boldsymbol{\tau}^* - \nabla^* \cdot \boldsymbol{\varsigma}^* = \rho^* \mathbf{g}^*, \tag{23}$$

$$\frac{\partial (\rho^* E^*)}{\partial t^*} + \nabla^* \cdot \left((\rho^* E^* + p^*)\mathbf{u}^* - (\boldsymbol{\tau}^* + \boldsymbol{\varsigma}^*)\mathbf{u}^*\right) + \nabla^* \cdot \mathbf{q}^* + \nabla^* \cdot \boldsymbol{\Pi}^* = \rho^* \mathbf{g}^* \cdot \mathbf{u}^* + \rho^* r^*, \tag{24}$$

wherein

$$p^* = \frac{8\theta^* \rho^*}{27(1-\rho^*)} - \rho^{*2}, \tag{25}$$

$$\boldsymbol{\tau}^* = \bar{\mu}^* \left(\nabla^* \mathbf{u}^* + \nabla^* \mathbf{u}^{*T} - \frac{2}{3} \nabla^* \cdot \mathbf{u}^* \mathbf{I}\right), \tag{26}$$

$$\boldsymbol{\varsigma}^* = \lambda^* \left(\left(\rho^* \Delta^* \rho^* + \frac{1}{2} |\nabla^* \rho^*|^2\right) \mathbf{I} - \nabla^* \rho^* \otimes \nabla^* \rho^*\right), \tag{27}$$

$$\mathbf{q}^* = -\kappa^* \nabla^* \theta^*, \tag{28}$$

$$\boldsymbol{\Pi}^* = \lambda^* \rho^* \nabla^* \cdot \mathbf{u}^* \nabla^* \rho^*, \tag{29}$$

$$\iota^* = \iota_{loc}^* + \frac{\lambda^*}{2\rho^*} |\nabla^* \rho^*|^2, \tag{30}$$

$$\iota_{loc}^* = -\rho^* + \frac{8\varpi}{27} \theta^*, \tag{31}$$

$$\nu_{loc}^* = -2\rho^* + \frac{8\theta^*}{27(1-\rho^*)} + \frac{8}{27} \theta^* \log\left(\frac{\rho^*}{1-\rho^*}\right) + \frac{8\varpi}{27} \theta^* \left(1 - \log(\theta^*)\right), \tag{32}$$

$$s^* = -\frac{8}{27} \log\left(\frac{\rho^*}{1-\rho^*}\right) + \frac{8\varpi}{27} \log\left(\theta^*\right), \tag{33}$$

$$\bar{\mu}^* = \frac{\bar{\mu}}{L_0 b\sqrt{ab}}, \tag{34}$$

$$\lambda^* = \frac{\lambda}{aL_0^2}, \tag{35}$$

$$\kappa^* = \kappa \frac{8}{27R(ab)^{1/2}bL_0}, \tag{36}$$

$$\mathbf{g}^* = \mathbf{g}\frac{L_0}{ab}. \tag{37}$$

The dimensionless isobaric thermal expansion coefficient β can be expressed as

$$\beta = \frac{c_\beta}{\theta_0}, \tag{38}$$

$$c_\beta = c_\beta(\rho^*, \theta^*) = \frac{4(1-\rho^*)}{4\theta^* - 27\rho^*(1-\rho^*)^2}. \tag{39}$$

The heat capacity at constant pressure can be expressed as

$$C_p = C_v + \chi R = (\varpi + \chi)R, \tag{40}$$

$$\chi = \chi(\rho^*, \theta^*) = \frac{c_\beta \theta^*}{1-\rho^*} = \frac{4\theta^*}{4\theta^* - 27\rho^*(1-\rho^*)^2}. \tag{41}$$

The non-dimensional kinematic viscosity and thermal diffusivity are given as

$$\bar{\nu}^* = \frac{\bar{\nu}}{L_0\sqrt{ab}}, \qquad \alpha^* = \alpha \frac{8\,(\varpi + \chi)}{27\sqrt{ab}L_0}. \tag{42}$$

At a given temperature, the equilibrium states of vapor and liquid can be found by equating the pressure and chemical potential [14, 21]. This solution procedure involves solving a system of two nonlinear algebraic equations. Table 1 shows the coexistent vapor and liquid densities and the corresponding values of c_β and χ at several different temperatures. For multiphase fluids, the Clausius–Clapeyron relation characterizes the energy released or absorbed during a phase transition process. It relates the latent heat $\Delta s\theta_0$ with the coexistence curve in the pressure-temperature diagram. In Fig. 2, the Clausius–Clapeyron relation for the van der Waals fluid at various temperature is illustrated, demonstrating the capability of the van der Waals fluid in modeling phase transition phenomena.

In this study, the reference length scale L_0 is chosen such that $|\mathbf{g}^*|$ is -0.025, wherein $|\cdot|$ denotes the l^2-norm of a vector. The rest three reference scales can be determined correspondingly from (21). Hence, the evolution of the system (22)–(24) is governed by four non-dimensional parameters $\bar{\mu}^*$, λ^*, ϖ, and κ^* (or equivalently, $\bar{\nu}^*$, λ^*, ϖ, and α^*), the initial and boundary conditions, and the geometry of the domain. In this work, I consider the fluid dynamics constrained by two parallel

Table 1 The vapor and liquid densities at the Maxwell states are given for various temperatures

θ^*	ρ_v^*	ρ_l^*	$c_\beta(\rho_v^*, \theta^*)$	$c_\beta(\rho_l^*, \theta^*)$	$\chi(\rho_v^*, \theta^*)$	$\chi(\rho_l^*, \theta^*)$
0.990	0.2682	0.4012	35.6607	31.5997	48.2414	52.2409
0.970	0.2228	0.4519	12.6242	10.2064	15.7562	18.0642
0.950	0.1930	0.4872	7.9449	6.0123	9.3528	11.1391
0.900	0.1419	0.5524	4.4078	2.9250	4.6231	5.8817
0.865	0.1161	0.5884	3.4974	2.1422	3.4226	4.5020
0.850	0.1066	0.6024	3.2397	1.9195	3.0822	4.1033
0.800	0.0799	0.6442	2.6789	1.4253	2.3292	3.2051
0.750	0.0591	0.6808	2.3701	1.1330	1.8892	2.6620
0.700	0.0427	0.7135	2.1957	0.9405	1.6055	2.2978

The values of the corresponding C_β and χ are also evaluated

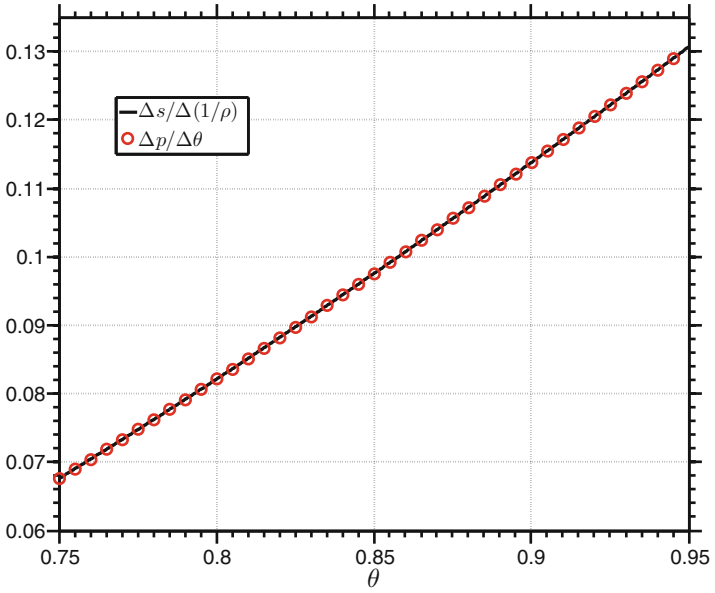


Fig. 2 Clausius–Clapeyron relation at different temperatures. At a fixed temperature θ_0, the liquid and vapor densities at the Maxwell state are denoted as $\rho_l^{\theta_0}$ and $\rho_v^{\theta_0}$. $\Delta s = s(\rho_v^{\theta_0}, \theta_0) - s(\rho_l^{\theta_0}, \theta_0)$, $\Delta(1/\rho) = 1/\rho_v^{\theta_0} - 1/\rho_l^{\theta_0}$. Let $p^{equ}(\theta)$ denote the equilibrium pressure at temperature θ, $\Delta p/\Delta\theta$ at θ_0 is evaluated using the central difference scheme: $p^{equ}(\theta_0 + \Delta\theta) - p^{equ}(\theta_0 - \Delta\theta)/(2\Delta\theta)$

plates. The gravity points in the direction opposite to the z-axis. The height between the two plates is denoted as $H = H^* L_0$; the length of the plates is denoted $D = D^* L_0$. The temperature on the bottom and top plates is denoted as θ_b and θ_t respectively, and the temperature difference is $\Delta\theta := \theta_b - \theta_t = \theta_0(\theta_b^* - \theta_t^*) = \theta_0 \Delta\theta^*$. The aspect ratio Λ characterizing the geometry of the domain is defined as

$$\Lambda := \frac{D}{H} = \frac{D^*}{H^*}. \tag{43}$$

The Rayleigh number is defined as

$$\mathrm{Ra} := \frac{g^* \Delta\theta^*}{\kappa^* \bar{\mu}^*} = \frac{g^* \Delta\theta^*}{\rho^{*2} \alpha^* \bar{\nu}^*}. \tag{44}$$

The Rayleigh number measures the relative strength of the buoyancy force in comparison with the resistant effect due to the viscosity and conductivity. The Prandtl number in the system is defined as

$$\mathrm{Pr} := \frac{\bar{\mu}^*}{\kappa^*} = \frac{\bar{\nu}^*}{\alpha^*}, \tag{45}$$

which measures the ratio of the momentum diffusivity to the thermal conductivity. In addition to the temperature difference $\Delta\theta^*$, the values of the temperature on the top and bottom boundaries will also affect the dynamics of the system since these values may change the liquid–vapor coexistence state (see Table 1). Hence, I introduce the arithmetic mean of the top and bottom temperature as one control parameter:

$$\theta_m := \frac{\theta_b + \theta_t}{2}. \tag{46}$$

The initial state of the total mass within the box is another control parameter for the system. It is described by the volumetric averaged density:

$$\rho_m := \frac{\int_\Omega \rho d\mathbf{x}}{\int_\Omega d\mathbf{x}}. \tag{47}$$

A major question people strive to answer is the dependence of the heat transport efficiency and the flow structure on the control parameters in the van der Waals fluid. The heat transport efficiency is described by the Nusselt number, which measures the relative strength of the total heat flux in comparison with the purely diffusive heat flux. The heat flux can be measured either over a fixed horizontal plane or in the whole body. The Nusselt number measured at a horizontal plane A is defined as

$$\begin{aligned} \mathrm{Nu}_{A,t} &:= \frac{\langle u_z \theta \rangle_{A,t} - \alpha \partial_z \langle \theta \rangle_{A,t}}{\alpha \Delta\theta H^{-1}} \\ &= \frac{8(\varpi + \chi)\rho^* \left\langle u_z^* \theta^* \right\rangle_{A,t} - 27\kappa^* \partial_{z^*} \langle \theta^* \rangle_{A,t}}{27\kappa^* \Delta\theta^* H^{*-1}}. \end{aligned} \tag{48}$$

Here, u_z is the velocity component in the z-direction; $\langle \cdot \rangle_{A,t}$ denotes an average operator over a horizontal plane A and over time for a quantity. If one average $\mathrm{Nu}_{A,t}$ in the z-direction, the volume-averaged Nusselt number is obtained as

$$\mathrm{Nu}_{V,t} := \frac{8(\varpi + \chi)\rho^* \left\langle u_z^* \theta^* \right\rangle_{V,t}}{27\kappa^* \Delta\theta^* H^{*-1}} + 1. \tag{49}$$

Here, $\langle\cdot\rangle_{V,t}$ represents an average operator over the volume and over time. Henceforth, I will restrict my discussion to the dimensionless form, and the superscript $*$ will be omitted for notational simplicity.

3 Numerical Methods

The numerical method for solving the governing equations (22)–(24) is based on a set of functional entropy variables. The mathematical entropy function associated with the van der Walls fluid is

$$H := -\rho s = \frac{8}{27}\rho \log\left(\frac{\rho}{1-\rho}\right) - \frac{8\varpi}{27}\rho \log(\theta).$$

For three-dimensional problems, the vector of conservation variables is

$$\boldsymbol{U}^T = [U_1, U_2, U_3, U_4, U_5] := [\rho, \rho u_1, \rho u_2, \rho u_3, \rho E].$$

The entropy variables $\boldsymbol{V}^T = [V_1, V_2, V_3, V_4, V_5]$ are defined as the functional derivatives of H with respect to $\boldsymbol{U}$:

$$V_i[\delta v_i] = \frac{\delta H}{\delta U_i}[\delta v_i], \quad i = 1, \ldots, 5,$$

wherein $\delta \boldsymbol{v}^T = [\delta v_1, \delta v_2, \delta v_3, \delta v_4, \delta v_5]$ are the test functions. The entropy variables $\boldsymbol{V}$ can be written explicitly as

$$V_1[\delta v_1] = \frac{1}{\theta}\left(\nu_{loc} - \frac{|\boldsymbol{u}|^2}{2}\right)\delta v_1 + \lambda\frac{1}{\theta}\nabla\rho \cdot \nabla\delta v_1,$$

$$V_i[\delta v_i] = \frac{u_{i-1}}{\theta}\delta v_i, \quad i = 2, 3, 4, \quad V_5[\delta v_5] = -\frac{1}{\theta}\delta v_5.$$

The definition of the entropy variable V_1 involves a non-local differential operator. Inspired from this fact, a new independent variable V is introduced as

$$V := \frac{1}{\theta}\left(\nu_{loc} - \frac{|\mathbf{u}|^2}{2}\right) - \lambda\nabla \cdot \left(\frac{\nabla\rho}{\theta}\right).$$

The fundamental thermodynamic relation between p and ν_{loc} allows us to express p in terms of V as

$$p = \rho V\theta - \rho\Psi_{loc} + \frac{\rho|\mathbf{u}|^2}{2} + \lambda\rho\theta\nabla\cdot\left(\frac{\nabla\rho}{\theta}\right). \tag{50}$$

Making use of the relation (50), the original strong-form problem (22)–(24) can be rewritten as

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0, \tag{51}$$

$$\begin{aligned}\frac{\partial(\rho\mathbf{u})}{\partial t} &+ \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) + \nabla\left(\rho V\theta + \frac{\rho|\mathbf{u}|^2}{2} + \lambda\rho\theta\nabla\cdot\left(\frac{\nabla\rho}{\theta}\right)\right)\\ &- \left(V\theta + \frac{|\mathbf{u}|^2}{2} + \lambda\theta\nabla\cdot\left(\frac{\nabla\rho}{\theta}\right)\right)\nabla\rho - H\nabla\theta - \nabla\cdot\boldsymbol{\tau} - \nabla\cdot\boldsymbol{\varsigma} = \rho\mathbf{b},\end{aligned} \tag{52}$$

$$\begin{aligned}\frac{\partial(\rho E)}{\partial t} &+ \nabla\cdot\left(\left(\rho V\theta - \theta H + \lambda|\nabla\rho|^2 + \rho|\mathbf{u}|^2 + \lambda\rho\theta\nabla\cdot\left(\frac{\nabla\rho}{\theta}\right)\right)\mathbf{u}\right)\\ &- \nabla\cdot((\boldsymbol{\tau}+\boldsymbol{\varsigma})\,\mathbf{u}) + \nabla\cdot\mathbf{q} + \nabla\cdot\boldsymbol{\Pi} = \rho\mathbf{b}\cdot\mathbf{u} + \rho r,\end{aligned} \tag{53}$$

$$V = \frac{1}{\theta}\left(\nu_{loc} - \frac{|\mathbf{u}|^2}{2}\right) - \lambda\nabla\cdot\left(\frac{\nabla\rho}{\theta}\right). \tag{54}$$

The new strong-form problem (51)–(54) is an equivalent statement of the original Navier–Stokes–Korteweg equations (22)–(24). Based on this new strong-form problem, the numerical scheme can be constructed. Let the time interval $(0, T)$ be divided into N_{ts} subintervals (t_n, t_{n+1}), $n = 0, \cdots, N_{ts}-1$, of size $\Delta t_n = t_{n+1} - t_n$. I use the notation

$$\mathbf{Y}_n^h := \left[\rho_n^h, \frac{u_{1,n}^h}{\theta_n^h}, \frac{u_{2,n}^h}{\theta_n^h}, \frac{u_{3,n}^h}{\theta_n^h}, \frac{-1}{\theta_n^h}, V_n^h\right]^T$$

to represent the fully discrete solutions at the time level n. I define the jump of density, linear momentum, and total energy over each time step as

$$\begin{aligned}[\![\rho_n^h]\!] &:= \rho_{n+1}^h - \rho_n^h, \quad [\![\rho_n^h\mathbf{u}_n^h]\!] := \rho_{n+1}^h\mathbf{u}_{n+1}^h - \rho_n^h\mathbf{u}_n^h,\\ \left[\rho_n^h E(\rho_n^h, \mathbf{u}_n^h, \theta_n^h)\right] &:= (\rho\Psi_{loc})(\rho_{n+\frac{1}{2}}^h, \theta_{n+1}^h) - (\rho\Psi_{loc})(\rho_{n+\frac{1}{2}}^h, \theta_n^h)\\ &\quad + (\rho\Psi_{loc})(\rho_{n+1}^h, \theta_{n+\frac{1}{2}}^h) - (\rho\Psi_{loc})(\rho_n^h, \theta_{n+\frac{1}{2}}^h)\\ &\quad - \theta_{n+\frac{1}{2}}^h\left(H(\rho_{n+1}^h, \theta_{n+1}^h) - H(\rho_n^h, \theta_n^h)\right)\\ &\quad - \frac{\theta_{n+1}^h - \theta_n^h}{2}\left(H(\rho_{n+\frac{1}{2}}^h, \theta_{n+1}^h) + H(\rho_{n+\frac{1}{2}}^h, \theta_n^h)\right)\end{aligned}$$

$$+\frac{(\theta^h_{n+1}-\theta^h_n)^3}{12}\frac{\partial^2 H}{\partial\theta^2}(\rho^h_{n+\frac{1}{2}},\theta^h_{n+1})$$
$$+\frac{1}{2}\left(\rho^h_{n+1}|\mathbf{u}^h_{n+1}|^2-\rho^h_n|\mathbf{u}^h_n|^2\right)+\lambda\left(|\nabla\rho^h_{n+1}|^2-|\nabla\rho^h_n|^2\right).$$

With the jump operators defined above, the fully discrete scheme can be stated as follows. In each time step, given $\mathbf{Y}^h_n$ and the time step Δt_n, find $\mathbf{Y}^h_{n+1}$ such that for all $w^h_1\in\mathscr{V}^h$, $\mathbf{w}^h=(w^h_2;w^h_3;w^h_4)^T\in\left(\mathscr{V}^h\right)^3$, $w^h_5\in\mathscr{V}^h$, and $w^h_6\in\mathscr{V}^h$,

$$\mathbf{B}^M(w^h_1;\mathbf{Y}^h_{n+1}):=\left(w^h_1,\frac{[\![\rho^h_n]\!]}{\Delta t_n}\right)_\Omega-\left(\nabla w^h_1,\rho^h_{n+\frac{1}{2}}\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega=0, \tag{55}$$

$$\begin{aligned}\mathbf{B}^U(\mathbf{w}^h;\mathbf{Y}^h_{n+1}):=&\left(\mathbf{w}^h,\frac{[\![\rho^h_n\mathbf{u}^h_n]\!]}{\Delta t_n}\right)_\Omega-\left(\nabla\mathbf{w}^h,\rho^h_{n+\frac{1}{2}}\mathbf{u}^h_{n+\frac{1}{2}}\otimes\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega\\
&-\left(\nabla\cdot\mathbf{w}^h,\rho^h_{n+\frac{1}{2}}V^h_{n+\frac{1}{2}}\theta^h_{n+\frac{1}{2}}+\frac{1}{2}\rho^h_{n+\frac{1}{2}}|\mathbf{u}^h_{n+\frac{1}{2}}|^2+\lambda\rho^h_{n+\frac{1}{2}}\theta^h_{n+\frac{1}{2}}\nabla\cdot\left(\frac{\nabla\rho^h_{n+\frac{1}{2}}}{\theta^h_{n+\frac{1}{2}}}\right)\right)_\Omega\\
&-\left(\mathbf{w}^h,\left(V^h_{n+\frac{1}{2}}\theta^h_{n+\frac{1}{2}}+\frac{|\mathbf{u}^h_{n+\frac{1}{2}}|^2}{2}+\lambda\theta^h_{n+\frac{1}{2}}\nabla\cdot\left(\frac{\nabla\rho^h_{n+\frac{1}{2}}}{\theta^h_{n+\frac{1}{2}}}\right)\right)\nabla\rho^h_{n+\frac{1}{2}}\right)_\Omega\\
&-\left(\mathbf{w}^h,H^h_{n+\frac{1}{2}}\nabla\theta^h_{n+\frac{1}{2}}\right)_\Omega+\left(\nabla\mathbf{w}^h,\boldsymbol{\tau}^h_{n+\frac{1}{2}}+\boldsymbol{\varsigma}^h_{n+\frac{1}{2}}\right)_\Omega-\left(\mathbf{w}^h,\rho^h_{n+\frac{1}{2}}\mathbf{b}\right)_\Omega=\mathbf{0},\end{aligned} \tag{56}$$

$$\begin{aligned}\mathbf{B}^E(w^h_5;\mathbf{Y}^h_{n+1}):=&\left(w^h_5,\frac{\left[\rho^h_nE(\rho^h_n,\mathbf{u}^h_n,\theta^h_n)\right]}{\Delta t_n}\right)_\Omega-\left(\nabla w^h_5,\left(\rho^h_{n+\frac{1}{2}}V^h_{n+\frac{1}{2}}\theta^h_{n+\frac{1}{2}}\right.\right.\\
&\left.\left.-\theta^h_{n+\frac{1}{2}}H^h_{n+\frac{1}{2}}+\frac{\lambda|\nabla\rho^h_{n+\frac{1}{2}}|^2}{2}+\lambda\rho^h_{n+\frac{1}{2}}\theta^h_{n+\frac{1}{2}}\nabla\cdot\left(\frac{\nabla\rho^h_{n+\frac{1}{2}}}{\theta^h_{n+\frac{1}{2}}}\right)+\rho^h_{n+\frac{1}{2}}|\mathbf{u}^h_{n+\frac{1}{2}}|^2\right)\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega\\
&+\left(\nabla w^h_5,\boldsymbol{\tau}^h_{n+\frac{1}{2}}\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega+\left(\nabla w^h_5,\boldsymbol{\varsigma}^h_{n+\frac{1}{2}}\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega-\left(\nabla w^h_5,\mathbf{q}^h_{n+\frac{1}{2}}+\boldsymbol{\Pi}^h_{n+\frac{1}{2}}\right)_\Omega\\
&-\left(w^h_5,\rho^h_{n+\frac{1}{2}}\mathbf{b}\cdot\mathbf{u}^h_{n+\frac{1}{2}}\right)_\Omega-\left(w^h_5,\rho^h_{n+\frac{1}{2}}r\right)_\Omega=0,\end{aligned} \tag{57}$$

$$\mathbf{B}^A(w^h_6;\mathbf{Y}^h_{n+1}):=\left(w^h_6,V^h_{n+\frac{1}{2}}-\frac{1}{2\theta^h_{n+\frac{1}{2}}}\left(\left(\nu_{loc}(\rho^h_n,\theta^h_{n+\frac{1}{2}})+\nu_{loc}(\rho^h_{n+1},\theta^h_{n+\frac{1}{2}})\right)\right.\right.$$

$$+\frac{[\![\rho_n^h]\!]^2}{12}\frac{\partial^2 \nu_{loc}}{\partial \rho^2}(\rho_n^h,\theta^h_{n+\frac{1}{2}})\Big)-\frac{\mathbf{u}_n^h\cdot\mathbf{u}_{n+1}^h}{2\theta^h_{n+\frac{1}{2}}}\Big)_{\Omega}-\left(\nabla w_6^h,\frac{\lambda\nabla\rho^h_{n+\frac{1}{2}}}{\theta^h_{n+\frac{1}{2}}}\right)_{\Omega}=0. \tag{58}$$

The main results of the fully discrete scheme (55)–(58) are stated in the following two theorems.

Theorem 1 *The solutions of the fully discrete scheme (55)–(58) satisfy*

$$\int_{\Omega}\Bigg(\frac{H(\rho^h_{n+1},\theta^h_{n+1})-H(\rho^h_n,\theta^h_n)}{\Delta t_n}+\nabla\cdot\left(H(\rho^h_{n+\frac{1}{2}},\theta^h_{n+\frac{1}{2}})\mathbf{u}^h_{n+\frac{1}{2}}\right)-\nabla\cdot\left(\frac{\mathbf{q}^h_{n+\frac{1}{2}}}{\theta^h_{n+\frac{1}{2}}}\right)$$
$$+\frac{\rho^h_{n+\frac{1}{2}}r}{\theta^h_{n+\frac{1}{2}}}\Bigg)dV_{\mathbf{x}}=-\int_{\Omega}\frac{1}{\theta^h_{n+\frac{1}{2}}}\boldsymbol{\tau}^h_{n+\frac{1}{2}}:\nabla\mathbf{u}^h_{n+\frac{1}{2}}dV_{\mathbf{x}}-\int_{\Omega}\frac{\kappa|\nabla\theta^h_{n+\frac{1}{2}}|^2}{\left(\theta^h_{n+\frac{1}{2}}\right)^2}dV_{\mathbf{x}}$$
$$-\int_{\Omega}\frac{1}{\theta^h_{n+\frac{1}{2}}\Delta t_n}\left(\frac{[\![\rho_n^h]\!]^4}{24}\frac{\partial^3\nu_{loc}}{\partial\rho^3}(\rho^h_{n+\xi_1},\theta^h_{n+\frac{1}{2}})-\frac{[\![\theta_n^h]\!]^4}{24}\frac{\partial^3 H}{\partial\theta^3}(\rho^h_{n+\frac{1}{2}},\theta^h_{n+\xi_2})\right)dV_{\mathbf{x}}$$
$$\le 0.$$

Theorem 2 *The local truncation error in time* $\boldsymbol{\Theta}(t)=\left(\Theta_\rho(t);\boldsymbol{\Theta}^T_{\mathbf{u}}(t);\Theta_E(t)\right)^T$ *is bounded by* $|\boldsymbol{\Theta}(t_n)|\le K\Delta t_n^2\mathbf{1}_5$ *for all* $t_n\in[0,T]$*, where* K *is a constant independent of* Δt_n *and* $\mathbf{1}_5=(1;1;1;1;1)^T$.

The above two theorems are proven in [21]. Theorem 1 states that the numerical method is unconditionally stable in entropy. Theorem 2 states that the temporal scheme is second-order accurate. The numerical scheme is implemented based on the PETSc package [3], and code verification has been performed a thorough comparison with manufactured solutions and "overkill" solutions [21]. In this study, C^1-continuous quadratic B-splines are employed to define $\mathscr{V}^h$ as well as the computational domain. Consequently, this approach may be considered as the isogeometric analysis method [13]. In all simulations, I fix $\varpi=3$, $\Lambda=2$. On boundary surfaces, ninety-degree contact angle boundary condition $\nabla\rho\cdot\mathbf{n}=0$ is imposed for the density variable; no-slip boundary condition is imposed for the velocity; Dirichlet boundary condition is imposed for the temperature on the top and bottom surfaces, and the adiabatic condition is imposed on the vertical boundary surfaces.

4 Results

4.1 Solution at Small Rayleigh Number

When the Rayleigh number is below the onset of instability, the system will evolve towards a steady state. The density and temperature profiles can be found by setting all time derivatives and **u** as zero in (22)–(24):

$$\nabla p - \nabla \cdot \boldsymbol{\varsigma} = \rho \mathbf{g}, \tag{59}$$

$$\nabla \cdot \mathbf{q} = 0. \tag{60}$$

In this example, the material moduli are chosen as $\bar{\mu} = 9.156 \times 10^{-3}$, $\kappa = 1.175 \times 10^{-2}$, $\lambda = 9.0 \times 10^{-6}$, $\theta_m = 0.865$, and $\Delta\theta = 0.17$. Notice that, instead of the thermal diffusivity α, the thermal conductivity κ is fixed as a constant in this example. Hence, the Eq. (60) is a linear Laplace's equation for the temperature field, and it can be solved analytically. The analytic steady state temperature profile is $\theta = \theta_t + \Delta\theta z/H$. Obtaining an analytic solution for the steady state density profile is non-trivial since it involves solving a third-order partial differential equation (59). The initial density is homogeneous and $\rho_m = 0.8$, 0.7, 0.6, and 0.5, respectively. The density profiles of the steady state solutions are illustrated in Fig. 3. All the

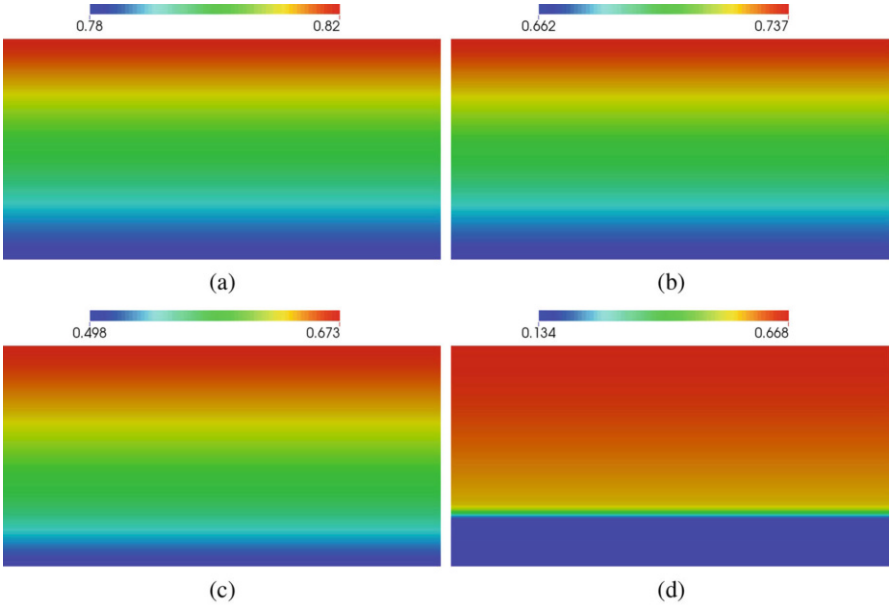


Fig. 3 The density profile of the steady state solution for (**a**) $\rho_m = 0.8$, (**b**) $\rho_m = 0.7$, (**c**) $\rho_m = 0.6$, and (**d**) $\rho_m = 0.5$

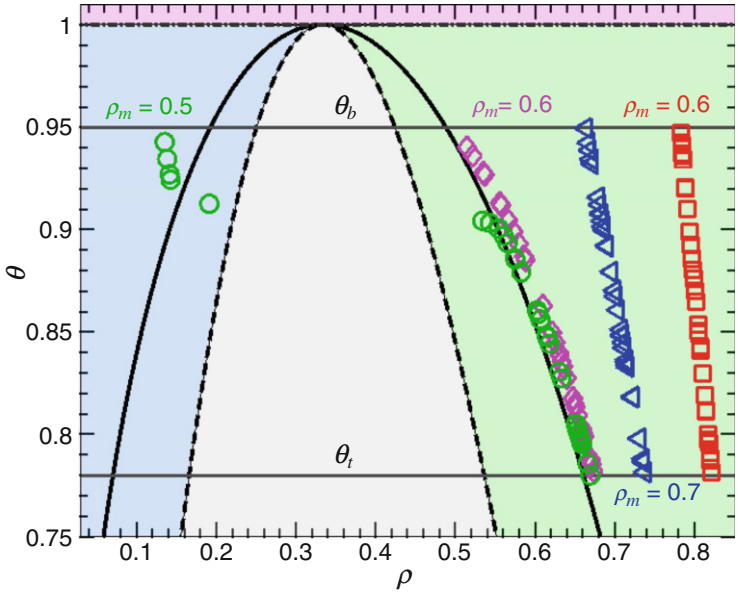


Fig. 4 The black solid line is the coexistence line in the ρ-θ phase space; the black dashed line is the spinodal line; the solid grey horizontal lines correspond to the constant top and bottom temperatures. The blue shaded area represents the vapor state; the green shaded area represents the liquid state; the red shaded area represents the supercritical fluid state; the grey shaded area corresponds to the unstable elliptic region [14]. A scattered plot of the temperature and density values is superimposed. Their values are sampled at random spatial positions of the steady state solutions for $\rho_m = 0.8$ (red squares), $\rho_m = 0.7$ (blue triangles), $\rho_m = 0.6$ (magenta diamonds), and $\rho_m = 0.5$ (green circles)

steady state solutions show a stratification pattern with low-density fluid in the bottom and high-density fluid in the top. The difference between the four solutions can be better illustrated by sampling the density and temperature of the solutions at random spatial locations (Fig. 4). For $\rho_m = 0.8$, 0.7, and 0.6, all the sampled particles fall into the liquid state (the green shaded area in Fig. 4). Hence, the corresponding solutions shown in Fig. 3a–c are all pure liquid with stratification. Unlike classical fluid stratification where the low-density fluid is above the high-density fluid, the Rayleigh number is small in these cases. The low Rayleigh number implies the relative strength of the gravity is small, and hence the fluid stratification is mainly driven by the temperature gradient. The linear temperature profile leads to a steady state pattern with light fluid in the bottom and dense fluid in the top. For $\rho_m = 0.5$, Fig. 4 shows that a fraction of the fluid transits to the vapor state. Consequently, the result shown in Fig. 3d is different from the results in Fig. 3a–c. The steady state solution for $\rho_m = 0.5$ consists of separated liquid and vapor states and a thin transitional layer. The vapor state is located at the bottom of the domain and the liquid state is located on top of the vapor fluid. This configuration is similar to the steady state liquid–vapor two-phase solution obtained under zero gravity [22].

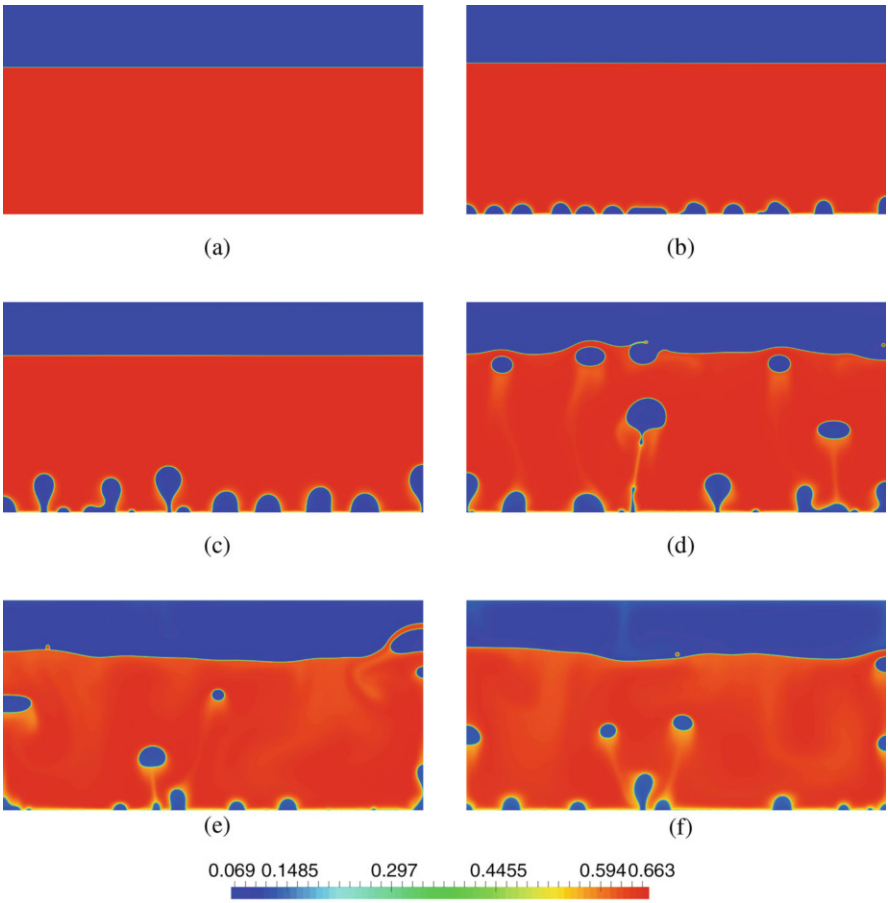


Fig. 5 Two-dimensional nucleate boiling simulation: Density profiles at (**a**) $t = 0.0$, (**b**) $t = 6.25$, (**c**) $t = 17.50$, (**d**) $t = 37.50$, (**e**) $t = 75.0$, and (**f**) $t = 100.0$

4.2 Nucleate and Film Boiling

In this section, I numerically study the capability of the van der Waals fluid model in modeling different regimes of boiling. In the first simulation, parameters are chosen as $\bar{\nu} = 1.150 \times 10^{-4}$, $\alpha = 1.725 \times 10^{-5}$, $\lambda = 1.190 \times 10^{-7}$, $\theta_m = 0.8625$, $\Delta\theta = 0.175$, and $\rho_m = 0.2424$. The spatial domain is discretized by 2048×1024 quadratic B-splines and the time integration is performed up to $T = 100.0$ with a fixed time step size $\Delta t = 5.0 \times 10^{-4}$. In Fig. 5, snapshots of the density profiles are illustrated at different time steps. The initial condition of this simulation represents the liquid fluid at the bottom and the vapor fluid at the top. A static free interface is located along $z = 0.35$, and the initial temperature is 0.775. During the initial times, random small vapor bubbles are generated at the heated bottom surface and

rise upward. At about $t = 37.50$, first a few bubbles reach the free surface, and in the meantime, there are more bubbles generated from the bottom surface. The coalescence of the vapor bubble with the free surface leads to surface waves. At $t = 37.50$ and $t = 75.0$, one can observe the surface waves. At time $t = 100.0$, there are tiny liquid droplets generated over the free surface as a result of the breakage of the liquid film.

In the next example, the kinematic viscosity is chosen as 4.600×10^{-4}, which is four times larger than that of the previous example. The increase in viscosity leads to slower dynamics of the fluid motion, and consequently, the numerical integration is performed up to $T = 500.0$. All the other parameters are identical to those in the previous case. In Fig. 6, snapshots of the temperature at different time steps are depicted. It can be seen that during the initial times, there is a thin film generated at

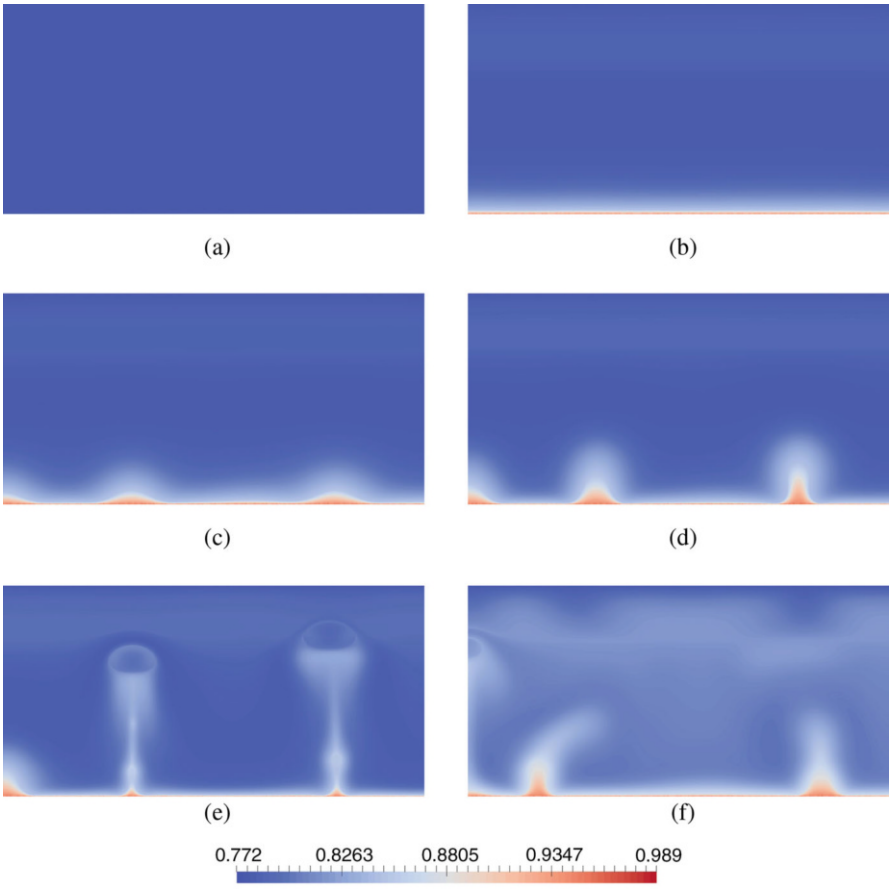


Fig. 6 Two-dimensional film boiling simulation: Temperature profiles at (**a**) $t = 0.0$, (**b**) $t = 100.0$, (**c**) $t = 175.0$, (**d**) $t = 200.0$, (**e**) $t = 225.0$, and (**f**) $t = 500.0$

the bottom heated surface. As time evolves, the film becomes unstable and three bubbles are formed. The bubbles gradually get detached from the film and rise upward carrying heat away from the thin film. At the final state, there is a mixing pattern of the temperature field driven by the free convection.

4.3 The Nusselt Number Scaling

In this section, the flow structure of the free convection in the van der Waals fluid is explored by relating the Nusselt number with the Rayleigh number Ra at a fixed Prandtl number Pr. In this suite of simulations, the volume averaged density ρ_m is chosen as 0.8. The viscosity coefficient $\bar{\mu}$ and the conductivity κ are progressively reduced so that the Rayleigh number ranges from 10 to 1.0×10^9 while the Prandtl number is maintained at 1.0. For Ra $\leq 10^8$, two different meshes are used for the same set of parameters to guarantee converged results for the Nusselt number. As is revealed in Fig. 4, the fluid in this simulation remains in the liquid phase, and the fluid density is stratified due to the temperature variation and the gravity force. In Figs. 7 and 8, the density and the temperature are depicted at time $t = 5000.0$ for different Rayleigh numbers. At smaller Rayleigh numbers, there are two convective rolls formed in the domain. For Rayleigh number 1.0×10^8, the rolls become unstable and the symmetry is broken, as is shown in Fig. 8d. The simulation results are reported in Table 2. In this set of simulations, the value of γ_{Nu} is about 0.22 by least square fitting. In [28], the value of γ_{Nu} for ideal gas is reported to be

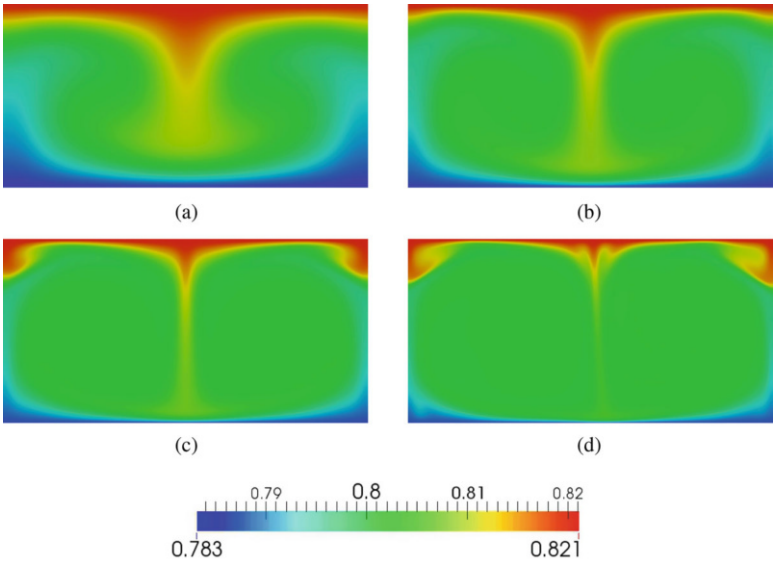


Fig. 7 The density profiles of the solutions at time step $t = 5000.0$: (**a**) Ra $= 1.0 \times 10^5$, (**b**) Ra $= 1.0 \times 10^6$, (**c**) Ra $= 1.0 \times 10^7$, and (**d**) Ra $= 1.0 \times 10^8$

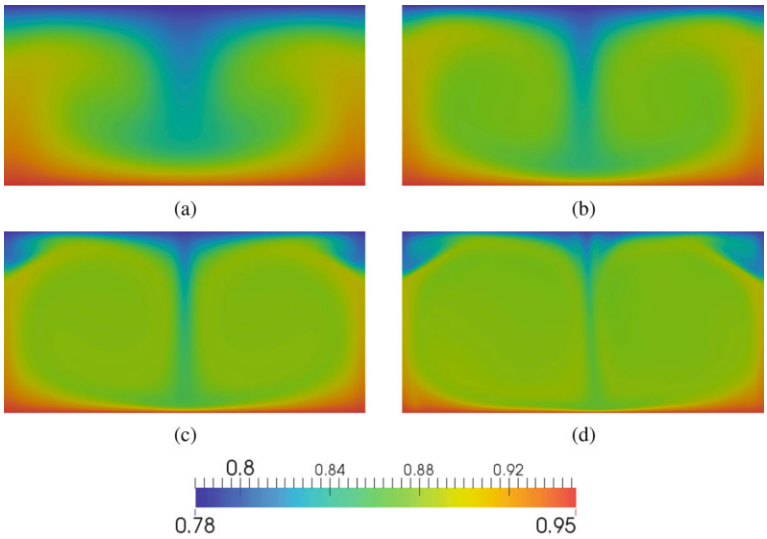


Fig. 8 The temperature profiles of the solutions at time step $t = 5000.0$: (**a**) $\mathrm{Ra} = 1.0 \times 10^5$, (**b**) $\mathrm{Ra} = 1.0 \times 10^6$, (**c**) $\mathrm{Ra} = 1.0 \times 10^7$, and (**d**) $\mathrm{Ra} = 1.0 \times 10^8$

Table 2 Summary of the simulation results with $\lambda = 9.0 \times 10^{-6}$, $\rho_m = 0.8$, $\theta_b = 0.95$, and $\theta_t = 0.78$

$N_x \times N_z$	$\bar{\mu}$	κ	$\mathrm{Ra}(\rho_m, \theta_m)$	$\mathrm{Pr}(\rho_m, \theta_m)$	$\mathrm{Nu}_{V,t}$
256×128	9.156×10^{-3}	1.175×10^{-2}	1.0×10^1	1.0	1.000
512×256	9.156×10^{-3}	1.175×10^{-2}	1.0×10^1	1.0	1.000
256×128	9.156×10^{-5}	1.175×10^{-4}	1.0×10^5	1.0	2.921
512×256	9.156×10^{-5}	1.175×10^{-4}	1.0×10^5	1.0	2.924
256×128	4.095×10^{-5}	5.257×10^{-5}	5.0×10^5	1.0	4.608
512×256	4.095×10^{-5}	5.257×10^{-5}	5.0×10^5	1.0	4.619
256×128	2.895×10^{-5}	3.717×10^{-5}	1.0×10^6	1.0	5.532
512×256	2.895×10^{-5}	3.717×10^{-5}	1.0×10^6	1.0	5.546
256×128	1.295×10^{-5}	1.662×10^{-5}	5.0×10^6	1.0	8.032
512×256	1.295×10^{-5}	1.662×10^{-5}	5.0×10^6	1.0	8.032
256×128	9.156×10^{-6}	1.175×10^{-5}	1.0×10^7	1.0	9.236
512×256	9.156×10^{-6}	1.175×10^{-5}	1.0×10^7	1.0	9.311
256×128	4.095×10^{-6}	5.257×10^{-6}	5.0×10^7	1.0	12.187
512×256	4.095×10^{-6}	5.257×10^{-6}	5.0×10^7	1.0	12.582
256×128	2.895×10^{-6}	3.717×10^{-6}	1.0×10^8	1.0	14.264
512×256	2.895×10^{-6}	3.717×10^{-6}	1.0×10^8	1.0	14.653
512×256	1.295×10^{-6}	1.662×10^{-6}	5.0×10^8	1.0	19.952
512×256	9.156×10^{-7}	1.175×10^{-6}	1.0×10^9	1.0	22.305

For the averaged density ρ_m and the arithmetic mean temperature θ_m, $c_\beta(\rho_m, \theta_m) = 0.3082$, $\chi(\rho_m, \theta_m) = 1.3328$. Consequently, $\mathrm{Ra}(\rho_m, \theta_m) = 1.0762 \times 10^{-3}/(\bar{\mu}\kappa)$ and $\mathrm{Pr}(\rho_m, \theta_m) = 1.2838\bar{\mu}/\kappa$

0.265; in [17], the value of γ_{Nu} for bubbling flows is given to be between $1/5$ and $1/3$. This suggests that the numerically measured γ_{Nu} in the van der Waals fluid conforms to reported values.

5 Conclusion

In this work, the Rayleigh-Bénard free convection in the van der Waals fluid is numerically investigated. Dimensional analysis is performed for the governing equations, and the control parameters for the convection problem are identified. The provably entropy stable algorithm and isogeometric analysis provide a reliable high-resolution numerical method for studying the free convection problem. The numerical results demonstrate the capability of the numerical model in describing boiling in different regimes. A suite of two-dimensional numerical simulations is also performed as an investigation of the heat transport property of the van der Waals fluid. The preliminary results indicate that this work provides a suitable framework to study the heat transport property for nucleate and film boiling. As the future work, the Nusselt number scaling law will be further investigated under different choices of the control parameters.

References

1. G. Accary, P. Bontoux, and B. Zappoli. Turbulent Rayleigh-Bénard convection in a near-critical fluid by three-dimensional direct numerical simulation. *Journal of Fluid Mechanics*, 619:127–145, 2008.
2. G. Ahlers, S. Grossmann, and D. Lohse. Heat transfer and large scale dynamics in turbulent Rayleigh-Bénard convection. *Reviews of Modern Physics*, 81:503–537, 2009.
3. S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In E. Arge, A.M. Bruaset, and H.P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
4. R.P. Behringer. Rayleigh-Bénard convection and turbulence in liquid helium. *Reviews of Modern Physics*, 57:657, 1985.
5. L. Biferale, P. Perlekar, M. Sbragaglia, and F. Toschi. Convection in multiphase fluid flows using lattice boltzmann methods. *Physical Review Letters*, 108:104502, 2012.
6. S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Dover, 1981.
7. Q. Chang and J.I.D. Alexander. Application of the lattice Boltzmann method to two-phase Rayleigh-Benard convection with a deformable interface. *Journal of Computational Physics*, 212:473–489, 2006.
8. P.G. Drazin and W.H. Reid. *Hydrodynamic Stability*. Cambridge University Press, 1981.
9. J.E. Dunn and J. Serrin. On the thermomechanics of interstitial working. *Archive for Rational Mechanics and Analysis*, 88:95–133, 1985.
10. M.O. McLinden E.W. Lemmon and D.G. Friend. *Thermophysical Properties of Fluid Systems* in NIST Chemistry WebBook, NIST Standard Reference Database Number 69. National Institute of Standards and Technology, Gaithersburg MD. (retrieved February 11, 2016).
11. A. Furukawa and A. Onuki. Convective heat transport in compressible fluids. *Physical Review E*, 66:016302, 2002.

12. M.E. Gurtin, E. Fried, and L. Anand. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, 2009.
13. 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*, 194:4135–4195, 2005.
14. D.C. Johnston. *Advances in Thermodynamics of the van der Waals Fluid*. Morgan & Claypool Publishers, 2014.
15. D. Juric and G. Tryggvason. Computations of boiling flows. *International Journal of Multiphase Flow*, 24:387–410, 1998.
16. D.J. Korteweg. Sur la forme que prennent les équations du mouvement des fluides si l'on tient compte des forces capillaires causées par des variations de densité considérables mais continues et sur la théorie de la capillarité dans l'hypothèse d'une variation continue de la densité. *Arch. Néerl.*, 6(2):1–24, 1901.
17. R. Lakkaraju, R.J.A.M. Stevens, P. Oresta, R. Verzicco, D. Lohse, and A. Prosperetti. Heat transport in bubbling turbulent convection. *Proceedings of the National Academy of Sciences of the United States of America*, 110:9237–9242, 2013.
18. L.D. Landau and E.M. Lifshitz. *Fluid mechanics*, volume 6 of *Course of Theoretical Physics*. Butterworth-Heinemann, 1987.
19. J. Liu. *Thermodynamically Consistent Modeling and Simulation of Multiphase Flows*. PhD thesis, The University of Texas at Austin, 2014.
20. J. Liu, H. Gomez, J.A. Evans, T.J.R. Hughes, and C.M. Landis. Functional Entropy Variables: A New Methodology for Deriving Thermodynamically Consistent Algorithms for Complex Fluids, with Particular Reference to the Isothermal Navier-Stokes-Korteweg Equations. *Journal of Computational Physics*, 248:47–86, 2013.
21. J. Liu, C.M. Landis, H. Gomez, and T.J.R. Hughes. Liquid-Vapor Phase Transition: Thermomechanical Theory, Entropy Stable Numerical Formulation, and Boiling Simulations. *Computer Methods in Applied Mechanics and Engineering*, 297:476–553, 2015.
22. A. Onuki. Dynamic van der Waals theory. *Physical Review E*, 75:036304, 2007.
23. P. Oresta, R. Verzicco, D. Lohse, and A. Prosperetti. Heat transfer mechanisms in bubbly Rayleigh-Bénard convection. *Physical Review E*, 80:026304, 2009.
24. J.S. Rowlinson. Translation of J.D. van der Waals' "The thermodynamic theory of capillarity under the hypothesis of a continuous variation of density". *Journal of Statistical Physics*, 20:200–244, 1979.
25. X. Shan. Simulation of Rayleigh-Bénard convection using a lattice Boltzmann method. *Physical Review E*, 55:2780, 1997.
26. J Sommeria. Turbulence: The elusive 'ultimate state' of thermal convection. *Nature*, 398:294–295, 1999.
27. P.J. Tackley. Effects of strongly variable viscosity on three-dimensional compressible convection in planetary mantles. *Journal of Geophysical Research*, 101:3311–3332, 1996.
28. A. Tilgner. Convection in an ideal gas at high Rayleigh numbers. *Physical Review E*, 84:026323, 2011.
29. K.Q. Xia. Current trends and future directions in turbulent thermal convection. *Theoretical and Applied Mechanics Letters*, 3:052001, 2013.

A General-Purpose NURBS Mesh Generation Method for Complex Geometries

Yuto Otoguro, Kenji Takizawa, and Tayfun E. Tezduyar

Abstract Spatial discretization with NURBS meshes is increasingly being used in computational analysis, including computational flow analysis with complex geometries. In flow analysis, compared to standard discretization methods, isogeometric discretization provides more accurate representation of the solid surfaces and increased accuracy in the flow solution. The Space-Time Computational Analysis (STCA), where the core method is the ST Variational Multiscale method, is increasingly relying on the ST Isogeometric Analysis (ST-IGA) as one of its key components, quite often also with IGA basis functions in time. The ST Slip Interface (ST-SI) and ST Topology Change methods are two other key components of the STCA, and complementary nature of all these ST methods makes the STCA powerful and practical. To make the ST-IGA use, and in a wider context the IGA use, even more practical in computational flow analysis with complex geometries, NURBS volume mesh generation needs to be easier and more automated. To that end, we present a general-purpose NURBS mesh generation method. The method is based on multi-block-structured mesh generation with existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces to the extent they are suitable for accurate and robust fluid mechanics computations. It is expected to retain the refinement distribution and element quality of the multi-block-structured mesh that we start with. The flexibility of discretization with the general-purpose mesh generation is supplemented with the ST-SI method, which allows, without loss of accuracy, C^{-1} continuity between NURBS patches and thus removes the matching requirement between the patches. We present mesh-quality performance studies for 2D and 3D meshes, including

Y. Otoguro (✉) · K. Takizawa
Department of Modern Mechanical Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: Yuto.Otoguro@tafsm.org; Kenji.Takizawa@tafsm.org

T. E. Tezduyar
Mechanical Engineering, Rice University, Houston, TX, USA

Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan
e-mail: tezduyar@tafsm.org

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_10

those for complex models, and test computation for a turbocharger turbine and exhaust manifold. These demonstrate that the general-purpose mesh generation method proposed makes the IGA use in computational flow analysis even more practical.

1 Introduction

This chapter is a revised and expanded version of a recent journal article [1]. The attractiveness of NURBS meshes in spatial discretization [2–5] led to their increasing use in computational analysis, including computational flow analysis with complex geometries. In flow analysis, compared to standard discretization methods, isogeometric discretization provides more accurate representation of the solid surfaces and increased accuracy in the flow solution. Because of that, the Space-Time Computational Analysis (STCA), which has been evolving since its inception in 1990 (see [1, 6–24]), has been increasingly relying on the ST Isogeometric Analysis (ST-IGA) [9, 11, 22] as one of its key methods, quite often also with IGA basis functions in time. The ST Variational Multiscale (ST-VMS) method [9, 10, 19] is the core method of the STCA in fluid mechanics, fluid–structure interactions (FSI), and flows with moving boundaries and interfaces (MBI). The ST Slip Interface (ST-SI) [20, 21] and ST Topology Change (ST-TC) [15, 17] methods are two other key methods.

1.1 ST-VMS and ST-SUPS

The ST-VMS method is the VMS version of the Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [6–8]. The DSD/SST method was introduced to function as a moving-mesh method for computation of flows with MBI, including FSI. Moving the fluid mechanics mesh to track (i.e., follow) a fluid–solid interface enables mesh-resolution control near the interface and, consequently, accurate flow representation there. The stabilization components of the DSD/SST method are the Streamline-Upwind/Petrov-Galerkin (SUPG) [25] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [6] stabilizations, and for that the method is now also called "ST-SUPS." The VMS components of the ST-VMS method are from the residual-based VMS (RBVMS) method [26–29]. The ST-VMS method has two more stabilization terms beyond those the ST-SUPS method has, and these additional terms give the method better turbulence modeling features. This significantly added to the scope and accuracy of the ST-SUPS method. Both methods, because of their ST accuracy features (see [9, 10]), are desirable also in computations that do not involve any MBI.

The Arbitrary Lagrangian–Eulerian (ALE) method, with the ALE finite element method introduced in 1981 [30], is an older and more commonly used moving-mesh

method. The ALE-VMS method [4, 31–35], which is the VMS version of the ALE method, was introduced after the ST-SUPS [6] and ALE-SUPS [36] methods and preceded the ST-VMS method [9]. The ALE-VMS and RBVMS methods are often used with special methods, such as those for weakly enforced no-slip boundary condition [37–39], "sliding interfaces" [5, 40] and backflow stabilization [41]. They have been successfully applied to different types of FSI, MBI, and fluid mechanics problems. The classes of problems include wind-turbine aerodynamics and FSI [42–49], more specifically, vertical-axis wind turbines [50, 51], floating wind turbines [52], wind turbines in atmospheric boundary layers [53], and fatigue-damage in wind-turbine blades [54], patient-specific cardiovascular fluid mechanics and FSI [3, 4, 55–59], biomedical-device FSI [60–65], ship hydrodynamics with free-surface flow and fluid–object interaction [66, 67], hydrodynamics and FSI of a hydraulic arresting gear [68, 69], hydrodynamics of tidal-stream turbines with free-surface flow [70], and bioinspired FSI for marine propulsion [71, 72].

The ST-SUPS and ST-VMS methods have also been successfully applied to different classes of FSI, MBI, and fluid mechanics problems. The classes of problems include spacecraft parachute FSI [13, 33, 73–79], wind-turbine aerodynamics [14, 16, 20, 33, 42, 48, 80–82], flapping-wing aerodynamics [11, 12, 15, 16, 18, 33, 82–85], cardiovascular fluid mechanics [15–17, 24, 82, 86–89], spacecraft aerodynamics [13, 90], thermo-fluid analysis of ground vehicles and their tires [19], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics in turbomachinery [91], flow analysis of turbocharger turbines [1, 22], flow around tires with road contact and deformation [23], ram-air parachutes [92], and compressible-flow parachute aerodynamics [93].

1.2 ST-TC

The ST-TC method [15, 17] was introduced for moving-mesh computation of flow problems with TC, such as contact between solid surfaces. It made moving-mesh computations possible even when there is an actual contact between solid surfaces or other TC. It can handle an actual TC while maintaining high-resolution boundary layer representation near solid surfaces. This enabled successful moving-mesh computation of heart valve flows [17, 24], wing clapping [18], and flow around a rotating tire with road contact and prescribed deformation [23].

1.3 ST-SI

The ST-SI method was introduced in [20], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS method when we have spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning solid surface spins with it, retaining the high-resolution representation

of the boundary layers. The SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the flow field. The starting point in the development of the ST-SI method was the version of the ALE-VMS method for "sliding interfaces" [5, 40]. In the ST-SI method, interface terms similar to those in the ALE-VMS version are added to the ST-VMS formulation to account for the compatibility conditions for the velocity and stress. An ST-SI version where the SI is between fluid and solid domains with weakly enforced Dirichlet boundary conditions for the fluid was also presented in [20]. The ST-SI method introduced in [21] for the coupled incompressible-flow and thermal-transport equations retain the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been successfully applied to aerodynamic analysis of vertical-axis wind turbines [20], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics in turbomachinery [91], flow analysis of turbocharger turbines [1, 22], flow around tires with road contact and deformation [23], aerodynamic analysis of ram-air parachutes [92], and flow analysis of heart valves [24].

The ST-SI methods have some additional good features. The integration of the ST-SI and ST-TC methods enables dealing with contact location change and contact sliding. This was used in the flow analysis around a tire with road contact and deformation [23] and in the flow analysis of heart valves [24]. The SI provides mesh generation flexibility in a general context by accurately connecting nonmatching meshes. This feature was used in the flow analysis of a heart valve [24] and a turbocharger turbine [1, 22]. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy (see [1]). In another version of the ST-SI method presented in [20], the SI is between a thin porous structure and the fluid on its two sides. With this, the fabric porosity is dealt with in a fashion consistent with how the standard two-sided SIs are dealt with and how the Dirichlet conditions are enforced weakly. Furthermore, this version of the ST-SI method enables handling thin structures that have T-junctions. This method has been successfully used in incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [92].

The compressible-flow ST-SI methods were introduced in [93], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [93]. These, together with the compressible-flow ST SUPG method [94], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled successful ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [93]. The computations were in the context of finite element discretization.

1.4 ST-IGA

The ST-IGA was introduced in [9]. It is the integration of the ST methods with isogeometric discretization. First computations with the ST-VMS method and ST-IGA were reported in [9] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. The stability and accuracy analysis given [9] for the advection equation showed that using higher-order basis functions in time would be essential in getting full benefit out of using higher-order basis functions in space.

In the early stages of the ST-IGA, the emphasis was on IGA basis functions in time. As pointed out in [9, 10] and demonstrated in [11, 12, 83], higher-order NURBS basis functions in time provide a more accurate representation of the motion of the solid surfaces and a mesh motion consistent with that. They also provide more efficiency in temporal representation of the motion and deformation of the volume meshes, and better efficiency in remeshing. That is how the ST/NURBS Mesh Update Method (STNMUM) was introduced and demonstrated in [11, 12, 83]. The name "STNMUM" was given in [14]. The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS basis functions in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly, and a "secondary mapping" [9–11, 33] enables also specifying a constant angular velocity for invariant speeds along the paths. The ST framework and NURBS in time also enable, with the "ST-C" method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [19, 21, 91, 95]. The STNMUM and desirable features of the ST-IGA with IGA basis functions in time have been demonstrated in many 3D computations. The classes of problems solved are flapping-wing aerodynamics [11, 12, 15, 16, 18, 33, 82–85], separation aerodynamics of spacecraft [13], wind-turbine aerodynamics [14, 16, 20, 48, 82], thermo-fluid analysis of ground vehicles and their tires [19], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics [91], and flow analysis of turbocharger turbines [1, 22].

The ST-IGA with IGA basis functions in space have been utilized in ST computational flow analysis of a turbocharger turbine [22], ram-air parachute [92], and a heart valve [24]. The turbocharger turbine analysis was based on the integration of the ST-SI method and ST-IGA. The IGA basis functions were used in the spatial discretization of the fluid mechanics equations and also in the temporal representation of the rotor and spinning-mesh motion. That enabled accurate representation of the turbine surfaces and rotor motion and increased accuracy in the flow solution. The ram-air parachute analysis was based on the integration of the ST-IGA, the ST-SI version that weakly enforces the Dirichlet conditions, and the ST-SI version that accounts for the porosity of a thin structure. The ST-IGA with IGA basis functions in space enabled, with relatively few number of unknowns, accurate representation of the parafoil geometry and increased accuracy in the flow solution. The volume mesh needed to be generated both inside and outside the parafoil. Mesh

generation inside was challenging near the trailing edge because of the narrowing space. Using IGA basis functions addressed that challenge and still kept the element density near the trailing edge at a reasonable level. The heart valve analysis was based on the integration of the ST-SI and ST-TC methods and ST-IGA. The "ST-SI-TC-IGA," beyond enabling a more accurate representation of the surfaces and increased accuracy in the flow solution, kept the element density in the narrow spaces near the contact areas at a reasonable level. When solid surfaces come into contact, the elements between the surface and the SI collapse. Before the elements collapse, the boundaries could be curved and rather complex, and the narrow spaces might have high-aspect-ratio elements. With NURBS elements, it was possible to deal with such adverse conditions rather effectively.

1.5 General-Purpose NURBS Mesh Generation

Complementary nature of the ST-SI and ST-TC methods and the ST-IGA makes the STCA powerful and practical. To make the ST-IGA use, and in a wider context the IGA use, even more practical in computational flow analysis with complex geometries, NURBS volume mesh generation needs to be easier and more automated. To that end, a general-purpose NURBS mesh generation method was introduced in [1]. The method is based on multi-block-structured mesh generation with existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces. The recovery of the original surfaces is to the extent they are suitable for accurate and robust fluid mechanics computations. The method is expected to retain the refinement distribution and element quality of the multi-block-structured mesh that we start with. Because there are ample good techniques and software for generating multi-block-structured meshes, the method makes general-purpose mesh generation relatively easy. The integration of the ST-SI method and ST-IGA allows, without loss of accuracy, C^{-1} continuity between NURBS patches and thus removes the matching requirement between the patches. We present mesh-quality performance studies for 2D and 3D meshes, including those for complex models. We also present test computation for a turbocharger turbine and exhaust manifold. The performance studies and test computation demonstrate that the general-purpose NURBS mesh generation method makes the IGA use in fluid mechanics computations even more practical.

We describe the general-purpose NURBS mesh generation method in Sect. 2. The mesh-quality performance studies are reported in Sect. 3. The ST-VMS and ST-SI methods are given in Sect. 4. The test computations for the turbine and manifold are presented in Sect. 5, and the concluding remarks are given in Sect. 6.

2 NURBS Mesh Generation Techniques

First we provide a brief introduction to the basic concepts and terminologies of NURBS meshes. We do that with figures. Figure 1 shows, as an example, a NURBS control mesh for a stator blade of a turbocharger turbine, depicting the NURBS patches and control points. Figure 2 shows the corresponding physical mesh, depicting the physical patches and elements. The relationship between the control and physical meshes is given by NURBS basis functions, defined on each patch by the knot vectors of the patch, with each space direction having its own knot vector.

2.1 Basic Technique

We start with a multi-block-structured mesh. Such grids are very common in finite difference and finite volume computations. The mesh quality can be measured in terms of the grid point distribution, grid line orthogonality, element aspect ratio, and the adjacent-element-length ratio. Generating high-quality meshes requires a number of skills and experience, but there are ample good techniques and software for generating multi-block-structured meshes. We assume that the multi-block-structured mesh we start with is the outcome of such techniques or software and is of high quality. The mesh consists of trilinear elements. We see each block as a precursor to a NURBS patch.

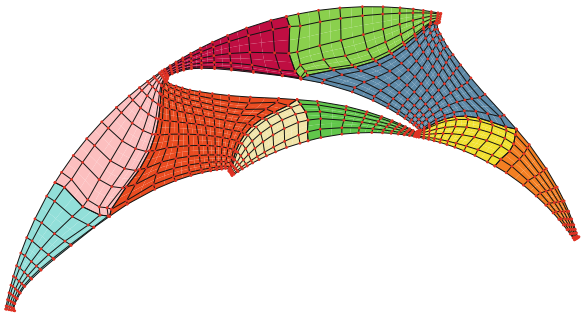


Fig. 1 Cross-section of a NURBS control mesh around a stator blade of a turbocharger turbine. The *red* circles represent the control points, and the different-colored areas are the NURBS patches

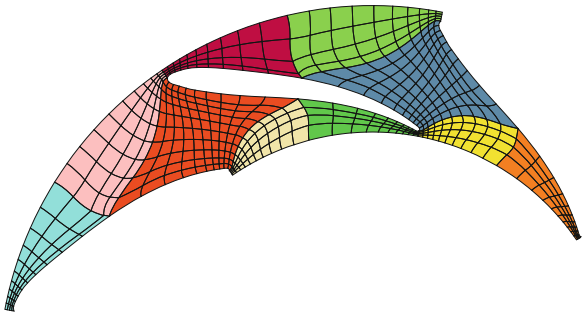


Fig. 2 Cross-sectional physical mesh corresponding to the control mesh in Fig. 1. The lines represent the element boundaries

Fig. 3 A schematic 1D finite element mesh, over the domain $[0, L]$

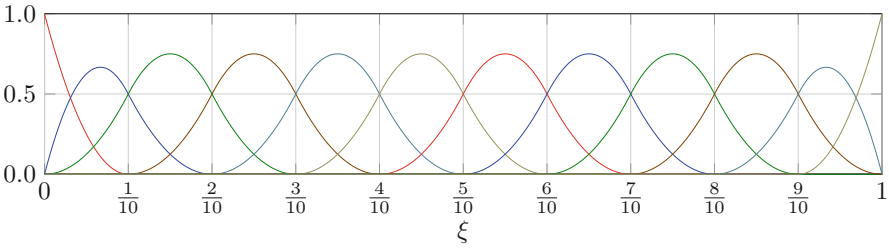


Fig. 4 NURBS basis functions in UPS corresponding to the finite element mesh in Fig. 3

The second step involves a number of projections. For those projections, we define a common parametric space between each block and the corresponding patch. We choose the parametric space to be $\boldsymbol{\xi} \in [0, 1]^3$. Then the knot vector of the patch for a given direction becomes

$$\boldsymbol{\Xi} = \{\underbrace{0, 0, \ldots 0}_{p+1}, \ldots, \underbrace{1, \ldots, 1}_{p+1}\}, \tag{1}$$

where p is the polynomial order for the NURBS basis functions used in that direction. We propose two methods to determine the knot values. In describing those methods, we use a schematic 1D finite element mesh, over the domain $[0, L]$, shown in Fig. 3, which we view as corresponding to one of the directions.

2.1.1 Uniform Parametric Spacing (UPS)

In this method, in each direction, the knot spacing $\Delta\xi$ is calculated from the number of elements in that direction. Supposing that there are n_{el} elements, the spacing is calculated as $\Delta\xi = 1/n_{\mathrm{el}}$, and the knot vector becomes

$$\boldsymbol{\Xi} = \left\{ \underbrace{0, 0, \ldots 0}_{p+1}, \frac{1}{n_{\mathrm{el}}}, \frac{2}{n_{\mathrm{el}}}, \ldots, \frac{n_{\mathrm{el}} - 1}{n_{\mathrm{el}}}, \underbrace{1, \ldots, 1}_{p+1} \right\}. \tag{2}$$

Figure 4 shows the resulting shape functions, N_A, where A is the basis function index, and Fig. 5 shows the absolute value of the physical-space derivative of the basis functions.

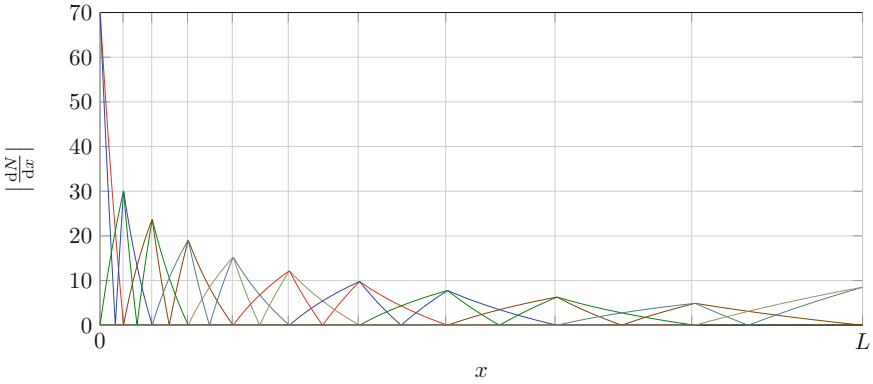


Fig. 5 Absolute value of the physical-space derivative of the basis functions in UPS

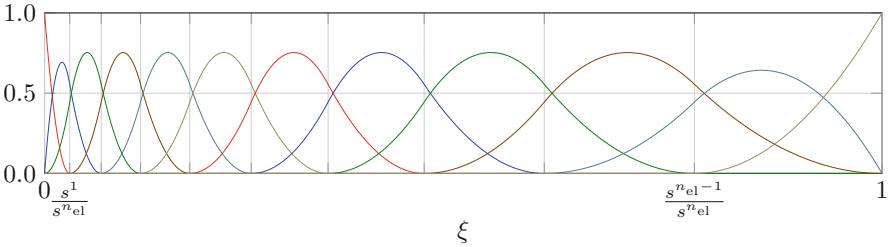


Fig. 6 NURBS basis functions in PSPPS corresponding to the finite element mesh in Fig. 3

2.1.2 Parametric Spacing Proportional to Physical Spacing (PSPPS)

In this method, we first define in each direction an "element length" h^e for element e, and from that we define s^e:

$$s^e = \sum_{k=1}^{e} h^k. \tag{3}$$

With that, the knot vector is defined as

$$\boldsymbol{\Xi} = \left\{ \underbrace{0, 0, \ldots 0}_{p+1}, \frac{s^1}{s^{n_{el}}}, \frac{s^2}{s^{n_{el}}}, \ldots, \frac{s^{n_{el}-1}}{s^{n_{el}}}, \underbrace{1, \ldots, 1}_{p+1} \right\}. \tag{4}$$

Figure 6 shows the resulting basis functions, and Fig. 7 shows the absolute value of the physical-space derivative of the basis functions.

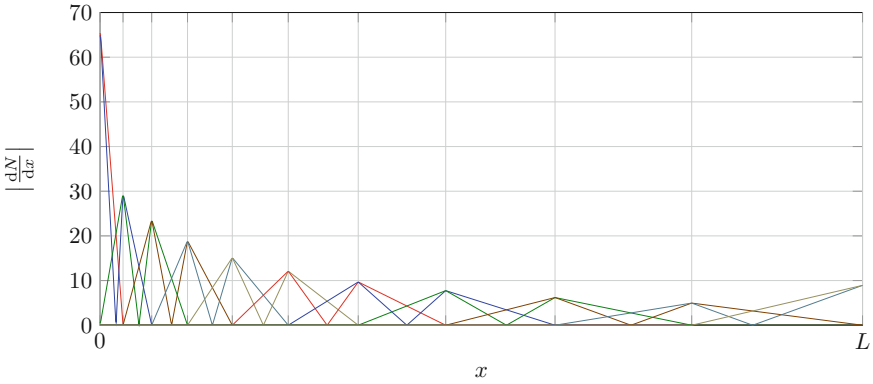


Fig. 7 Absolute value of the physical-space derivative of the basis functions in PSPPS

Remark 1 Since the knot vectors will be defined in each path in a multi-dimensional context, to obtain the element length in each direction, we need to do some averaging in the other two directions. For example, we can do the averaging for the element length in the parametric direction 1 as

$$h_1^i = \frac{1}{(n_{\mathrm{el}})_3\,(n_{\mathrm{el}})_2} \sum_{j=k}^{(n_{\mathrm{el}})_3} \sum_{j=1}^{(n_{\mathrm{el}})_2} h_1^{i,j,k}, \tag{5}$$

where $h_1^{i,j,k}$ is the element length for the (i, j, k) element of the structured mesh.

2.1.3 Projections

For each patch, the projections are done hierarchically. First the control points at the corners of the patch are set to the same locations as the corner grid points of the block. Then the edges between the corners are projected by using the common parametric space:

$$\int_0^1 \mathbf{w}^h \cdot \left(\mathbf{x}^h - \boldsymbol{\chi}^h\right) \mathrm{d}\xi = 0. \tag{6}$$

Here $\boldsymbol{\chi}^h$ is the position representation in the block, $\mathbf{x}^h$ is the position representation in the patch, and $\mathbf{w}^h$ is the test function of the projection. The parametric coordinate ξ used in the integration is a general representation for all edges. Next the surfaces between the edges are projected:

$$\int_0^1 \int_0^1 \mathbf{w}^h \cdot \left(\mathbf{x}^h - \boldsymbol{\chi}^h\right) \mathrm{d}\xi_1 \mathrm{d}\xi_2 = 0. \tag{7}$$

Here, the combination of the parametric coordinates ξ_1 and ξ_2 is a general representation for all surfaces. The projection sequence is completed with the projection of the volumes between the surfaces:

$$\int_0^1 \int_0^1 \int_0^1 \mathbf{w}^h \cdot \left(\mathbf{x}^h - \boldsymbol{\chi}^h\right) \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 = 0. \tag{8}$$

2.1.4 Merging the Patches

For two adjacent blocks, the projections described in Sect. 2.1.3 result in the same control point positions over the surface they share, provided that the two blocks have the same knot vectors over the shared surface. This happens automatically in UPS. To make it happen in PSPPS, the requirement is taken into account while doing the element length averaging. The control point variables are declared to be the same between the adjacent patches over the shared surface, which results in C^0 continuity for the basis function across the surface.

Remark 2 Alternatively, as proposed and used in [22, 24], we can declare the control point variables over the shared surface to be separate, resulting in C^{-1} continuity, and connect them with the ST-SI method.

2.2 Element Reduction

We know from our experience that, to generate good quality NURBS meshes with the method proposed, in many cases we may need more elements in the finite element mesh than the target number of NURBS elements. Therefore we apply a reduction factor r to the number of elements in the block, which we can possibly be different for each direction in the block. However, if we use different factors in different directions, two adjacent blocks must have the same reduction factor in each of the two directions of the surface they share. To make the number of elements an integer, after applying the factor r, we raise the value to the next integer. After the element reduction, we use the modified versions of the UPS and PSPPS methods, which we describe in the following two subsections.

2.2.1 Element-Reduced UPS

We just define the knot vectors with uniform spacing based on the reduced number of elements. Figure 8 shows the resulting basis functions in element-reduced UPS corresponding to the finite element mesh in Fig. 3.

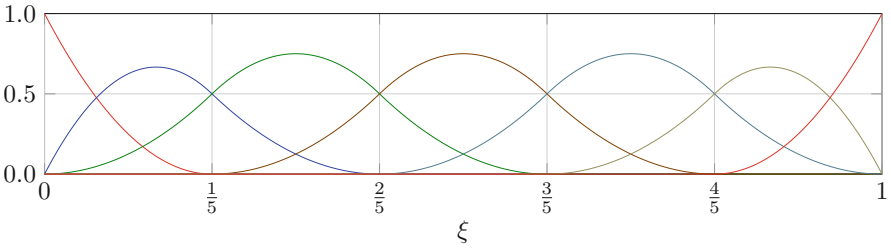


Fig. 8 NURBS basis functions in element-reduced UPS ($r = 0.5$) corresponding to the finite element mesh in Fig. 3

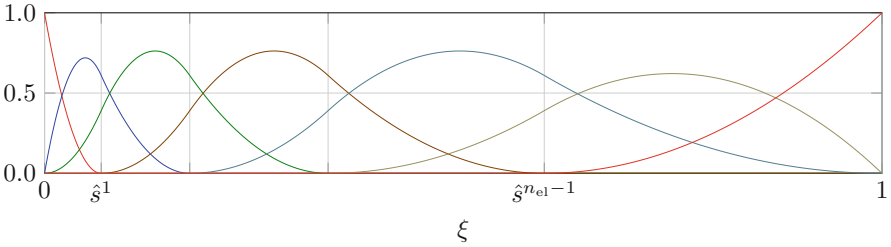


Fig. 9 NURBS basis functions in element-reduced PSPPS corresponding to the finite element mesh in Fig. 3

2.2.2 Element-Reduced PSPPS

We first define a 1D element spacing function $h(\xi)$, integrate it as

$$s(\xi) = \int_0^{\xi} h(\theta)\mathrm{d}\theta, \tag{9}$$

and normalize the outcome as $\hat{s}(\xi) = s(\xi)/s(1)$. Then, the knot values are defined as

$$\hat{s}^A = \hat{s}\left(\frac{A}{n_{\mathrm{el}}}\right), \tag{10}$$

where $A = 0, 1, \ldots, n_{\mathrm{el}}$. Figure 9 shows the resulting basis functions in element-reduced PSPPS corresponding to the finite element mesh in Fig. 3. In the multi-dimensional context, we use the integration version of Eq. (5).

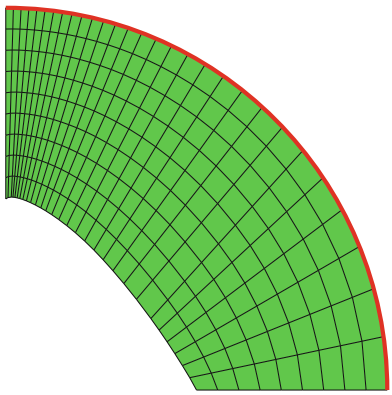


Fig. 10 A finite element mesh that includes a boundary that is an arc (*red* curve) in the exact representation

2.3 Techniques for Recovering the Exact Surfaces

In the NURBS mesh generation method proposed, we can recover the exact surfaces, instead of just relying on the surfaces represented by the finite element mesh.

2.3.1 Special Technique for Arc Surfaces

Suppose we have a finite element mesh with a boundary that is an arc in the exact representation (see Fig. 10). We first represent the arc by three control points and the corresponding quadratic NURBS basis functions. The arc template is shown in Fig. 11, and the corresponding knot vector and weights are

$$\boldsymbol{\xi} = \{0, 0, 0, 1, 1, 1\} \tag{11}$$

and

$$w^1 = 1, \qquad w^2 = \cos q, \qquad w^3 = 1. \tag{12}$$

For details, see [10, 33]. Figure 12 shows the NURBS representation of the arc, superimposed on the finite element mesh. Then we insert to the knot vector given by Eq. (11) the knots of the NURBS mesh we started with. From that, we obtain the new control points and weights representing the boundary. The new mesh can be seen in Fig. 13.

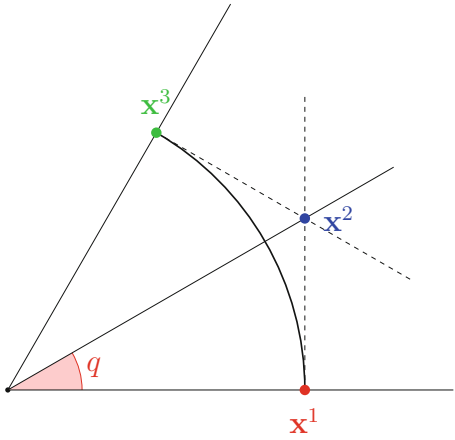


Fig. 11 The arc template with three control points representing an arc with angle $2q$

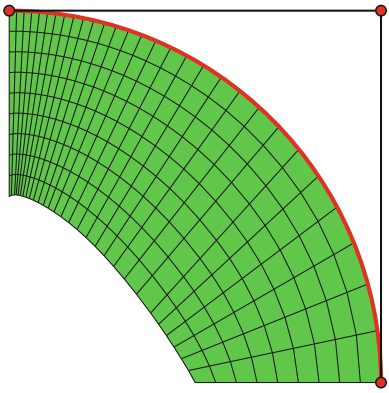


Fig. 12 NURBS representation of the arc, superimposed on the finite element mesh, where the three *red* circles are the control points

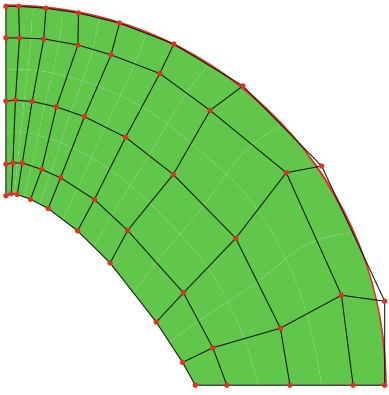


Fig. 13 NURBS control mesh with the exact representation of the arc boundary

2.3.2 General Technique for CAD Surfaces

If the CAD surface basis function space is a subset of our NURBS mesh basis function space, we can recover the exact surface by knot insertion to the CAD space to obtain our NURBS space. If not, the best we can do is to project the CAD surface to our NURBS space.

3 Mesh-Quality Performance

In the context of 2D and 3D examples, we evaluate the mesh-quality performance of the NURBS mesh generation method proposed. The quality measures are the grid line orthogonality and adjacent-element-length ratio.

3.1 2D Mesh

Figure 14 shows the finite element mesh. Figure 15 shows the quadratic NURBS mesh corresponding to the finite element mesh in Fig. 14. Figure 16 shows the distribution of the grid line orthogonality measure for the finite element and NURBS meshes, and Table 1 shows the adjacent-element-length ratio for the two meshes.

Figure 17 shows the element-reduced quadratic NURBS mesh ($r = 0.5$) corresponding to the finite element mesh in Fig. 14. Figure 18 shows the distribution of the grid line orthogonality measure for the finite element and element-reduced NURBS meshes, and Table 2 shows the adjacent-element-length ratio for the two meshes.

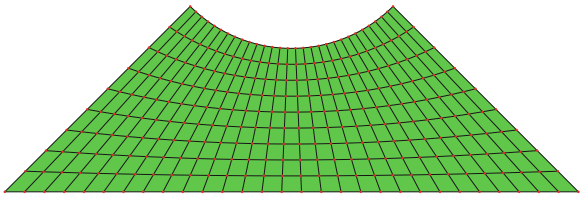


Fig. 14 2D mesh. Finite element mesh

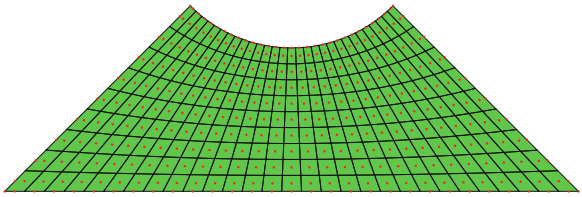


Fig. 15 2D mesh. Control points and physical elements of the quadratic NURBS mesh corresponding to the finite element mesh in Fig. 14

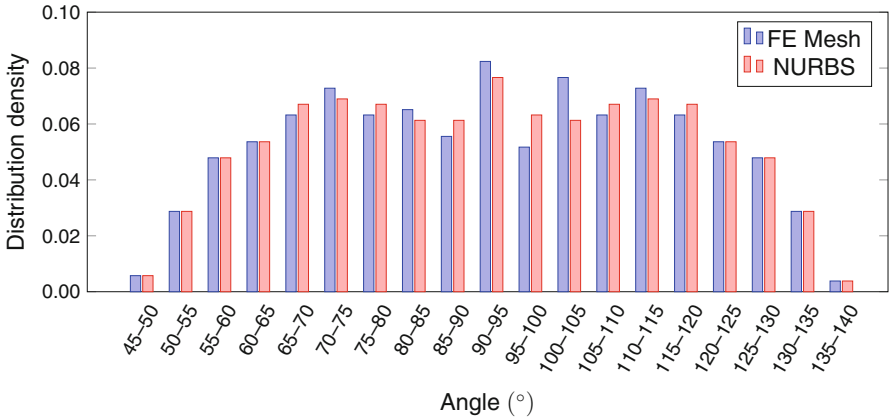


Fig. 16 2D mesh. Distribution of the grid line orthogonality measure for the finite element and NURBS meshes

Table 1 2D mesh

	FE mesh	NURBS patch
Maximum value	1.46	1.46
Standard deviation	0.0654	0.0654

Adjacent-element-length ratio for the finite element and NURBS meshes

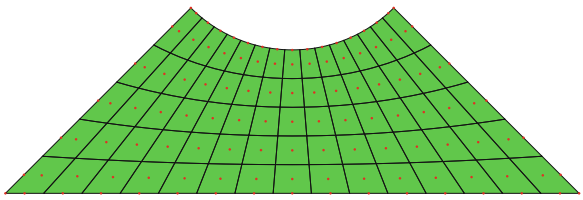


Fig. 17 2D mesh. Control points and physical elements of the element-reduced quadratic NURBS mesh (r = 0.5) corresponding to the finite element mesh in Fig. 14

3.2 Aorta and Branches

This is an example for internal flows. The model we use for the aorta and branches is shown in Fig. 19. Figure 20 shows the quadratic NURBS control mesh. Figure 21 shows the distribution of the grid line orthogonality measure for the finite element and NURBS meshes, and Table 3 shows the adjacent-element-length ratio for the two meshes.

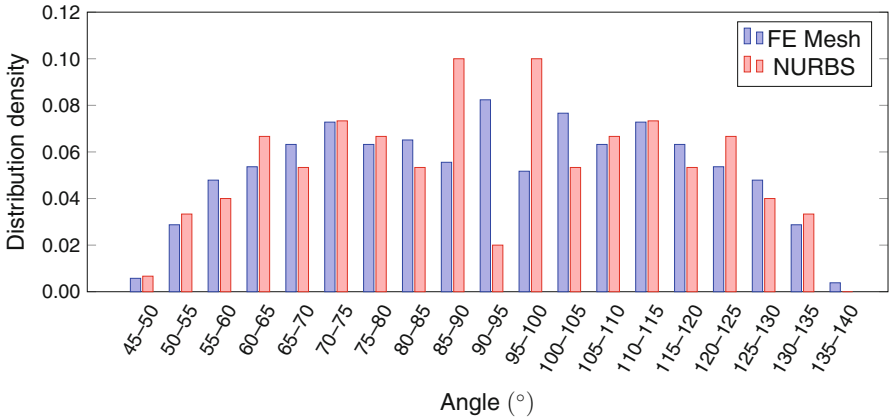


Fig. 18 2D mesh. Distribution of the grid line orthogonality measure for the finite element and element-reduced NURBS meshes

Table 2 2D mesh

	FE mesh	NURBS patch
Maximum value	1.46	2.37
Standard deviation	0.0654	0.1152

Adjacent-element-length ratio for the finite element and element-reduced NURBS meshes

3.3 Disk-Gap-Band Parachute

This is an example for external flows. The model for the disk-gap-band parachute is shown in Fig. 22. Figure 23 shows the quadratic NURBS control mesh. Figure 24 shows the distribution of the grid line orthogonality measure for the finite element and NURBS meshes, and Table 4 shows the adjacent-element-length ratio for the two meshes.

4 ST-VMS and ST-SI Methods

We include from [20, 23] the ST-VMS and ST-SI methods.

Fig. 19 Aorta and branches. Model

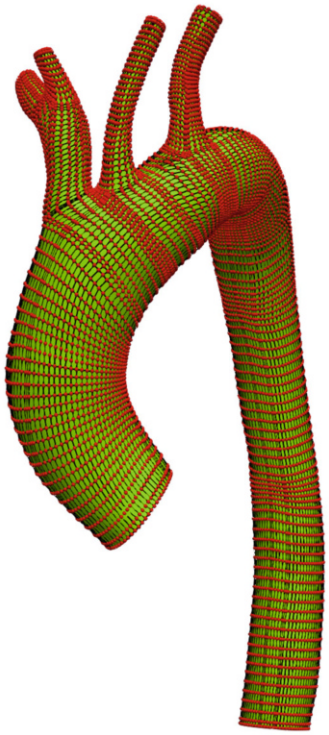


Fig. 20 Aorta and branches. Quadratic NURBS control mesh

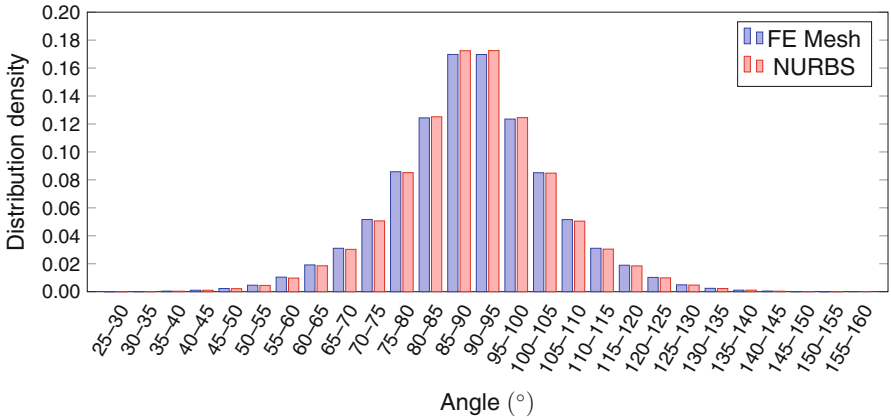


Fig. 21 Aorta and branches. Distribution of the grid line orthogonality measure for the finite element and NURBS meshes

Table 3 Aorta and branches

	FE mesh	NURBS patch
Maximum value	8.83	8.98
Standard deviation	0.2145	0.1677

Adjacent-element-length ratio for the finite element and NURBS meshes

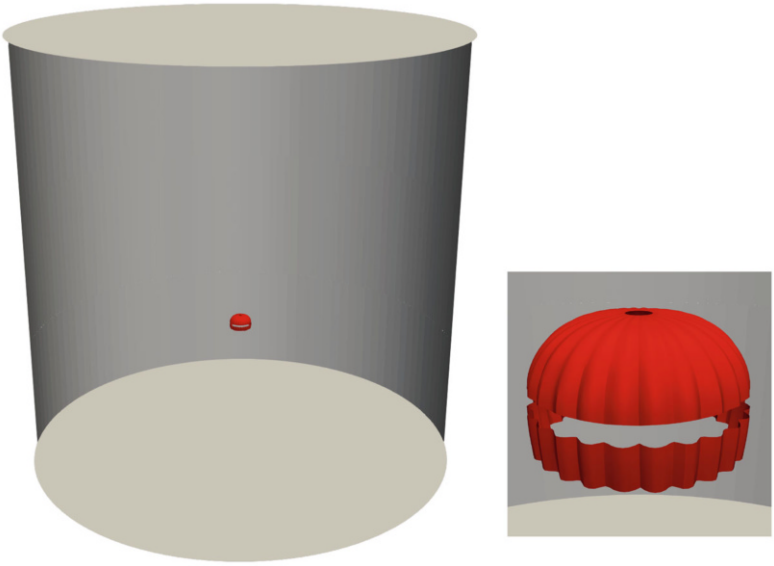


Fig. 22 Disk-gap-band parachute. Model

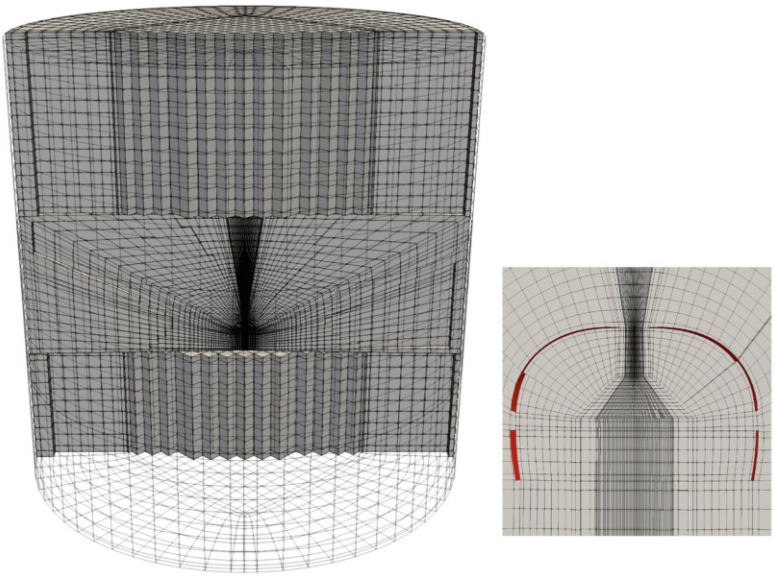


Fig. 23 Disk-gap-band parachute. Quadratic NURBS control mesh

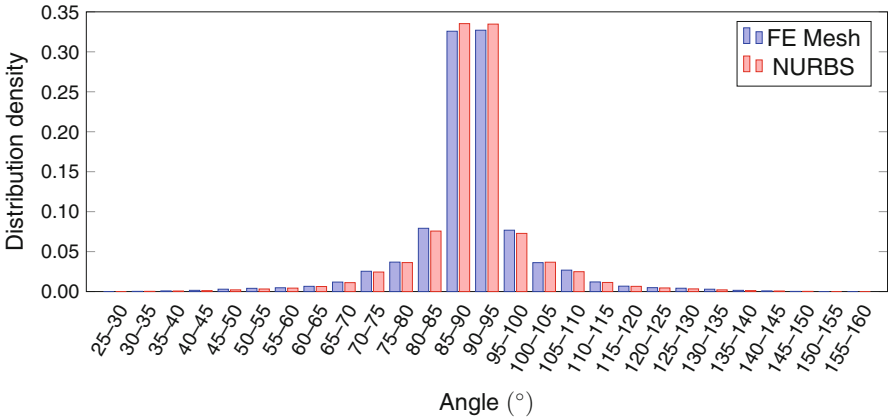


Fig. 24 Disk-gap-band parachute. Distribution of the grid line orthogonality measure for the finite element and NURBS meshes

Table 4 Disk-gap-band parachute

	FE mesh	NURBS patch
Maximum value	4.24	3.88
Standard deviation	0.1477	0.1644

Adjacent-element-length ratio for the finite element and NURBS meshes

The ST-VMS method is given as

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&\quad - \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) + \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&\quad + \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}Q \\
&\quad - \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \tau_\mathrm{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \mathrm{d}Q \\
&\quad - \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}^2}{\rho} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q = 0, \qquad (13)
\end{aligned}$$

where

$$\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \qquad (14)$$

$$r_\mathrm{C}(\mathbf{u}^h) = \boldsymbol{\nabla} \cdot \mathbf{u}^h \qquad (15)$$

are the residuals of the momentum equation and incompressibility constraint. Here, ρ, $\mathbf{u}$, p, $\mathbf{f}$, $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$, and $\mathbf{h}$ are the density, velocity, pressure, external force, stress tensor, strain rate tensor, and the traction specified at the boundary. The test functions associated with the velocity and pressure are $\mathbf{w}$ and q. A superscript "h" indicates that the function is coming from a finite-dimensional space. The symbol Q_n represents the ST slice between time levels n and $n+1$, $(P_n)_\mathrm{h}$ is the part of the lateral boundary of that slice associated with the traction boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and n_el is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level. See [7, 8, 14, 19, 20] for the definitions used here for the stabilization parameters τ_SUPS and ν_LSIC. For more ways of calculating the stabilization parameters in finite element computation of flow problems, see [96–117].

Remark 3 The ST-SUPS method can be obtained from the ST-VMS method by dropping the eighth and ninth integrations.

In the ST-SI method, labels "Side A" and "Side B" represent the two sides of the SI. We add boundary terms to Eq. (13). The boundary terms are first added separately for the two sides, using test functions $\mathbf{w}_{\mathrm{A}}^h$ and q_{A}^h and $\mathbf{w}_{\mathrm{B}}^h$ and q_{B}^h. Putting them together, the complete set of terms added becomes

$$
\begin{aligned}
&- \int_{(P_n)_{\mathrm{SI}}} \left(q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} - q_{\mathrm{A}}^h \mathbf{n}_{\mathrm{A}} \right) \cdot \frac{1}{2} \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2} \left(\left(\mathscr{F}_{\mathrm{B}}^h - \left| \mathscr{F}_{\mathrm{B}}^h \right| \right) \mathbf{u}_{\mathrm{B}}^h - \left(\mathscr{F}_{\mathrm{B}}^h - \left| \mathscr{F}_{\mathrm{B}}^h \right| \right) \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{A}}^h \cdot \frac{1}{2} \left(\left(\mathscr{F}_{\mathrm{A}}^h - \left| \mathscr{F}_{\mathrm{A}}^h \right| \right) \mathbf{u}_{\mathrm{A}}^h - \left(\mathscr{F}_{\mathrm{A}}^h - \left| \mathscr{F}_{\mathrm{A}}^h \right| \right) \mathbf{u}_{\mathrm{B}}^h \right) \mathrm{d}P \\
&+ \int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{n}_{\mathrm{B}} \cdot \mathbf{w}_{\mathrm{B}}^h + \mathbf{n}_{\mathrm{A}} \cdot \mathbf{w}_{\mathrm{A}}^h \right) \frac{1}{2} \left(p_{\mathrm{B}}^h + p_{\mathrm{A}}^h \right) \mathrm{d}P \\
&- \int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h \right) \cdot \left(\hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu \left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^h) + \boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^h) \right) \right) \mathrm{d}P \\
&- \gamma_{\mathrm{ACI}} \int_{(P_n)_{\mathrm{SI}}} \hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu \left(\boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{B}}^h \right) + \boldsymbol{\varepsilon} \left(\mathbf{w}_{\mathrm{A}}^h \right) \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P \\
&+ \int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h \right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h \right) \mathrm{d}P, \qquad (16)
\end{aligned}
$$

where

$$\mathscr{F}_{\mathrm{B}}^h = \mathbf{n}_{\mathrm{B}} \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{v}_{\mathrm{B}}^h \right), \qquad (17)$$

$$\mathscr{F}_{\mathrm{A}}^h = \mathbf{n}_{\mathrm{A}} \cdot \left(\mathbf{u}_{\mathrm{A}}^h - \mathbf{v}_{\mathrm{A}}^h \right), \qquad (18)$$

$$h = \frac{h_{\mathrm{B}} + h_{\mathrm{A}}}{2}, \qquad (19)$$

$$h_{\mathrm{B}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{B}} \cdot \nabla N_a^{\alpha} \right| \right)^{-1} \text{(for Side B)}, \qquad (20)$$

$$h_{\mathrm{A}} = 2 \left(\sum_{\alpha=1}^{n_{\mathrm{ent}}} \sum_{a=1}^{n_{\mathrm{ens}}} \left| \mathbf{n}_{\mathrm{A}} \cdot \nabla N_a^{\alpha} \right| \right)^{-1} \text{(for Side A)}, \qquad (21)$$

$$\hat{\mathbf{n}}_{\mathrm{B}} = \frac{\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}}{\| \mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}} \|}. \qquad (22)$$

Here, $(P_n)_{\mathrm{SI}}$ is the SI in the ST domain, $\mathbf{n}$ is the unit normal vector, $\mathbf{v}$ is the mesh velocity, n_{ens} and n_{ent} are the number of spatial and temporal element nodes, N_a^{α} is the basis function associated with spatial and temporal nodes a and α, $\gamma_{\mathrm{ACI}} = 1$,

and C is a nondimensional constant. For our element length definition, we typically set $C = 1$. In the computations reported in this article, $C = 2$.

A number of remarks were provided in [20] to explain the added terms and to comment on related interpretations. We refer the reader interested in those details to [20].

Remark 4 A coefficient γ_{ACI} was added in [23] to the sixth integration so that we have the option of using $\gamma_{\text{ACI}} = -1$. This option was added, in [93], also in the context of compressible flows. Using $\gamma_{\text{ACI}} = 1$ in a discontinuous Galerkin method was introduced in the symmetric interior penalty Galerkin method [118], and using $\gamma_{\text{ACI}} = -1$ was introduced in the nonsymmetric interior penalty Galerkin method [119]. Stabilized methods based on both $\gamma_{\text{ACI}} = 1$ and -1 were reported in [37] in the context of the advection–diffusion equation. In the computations reported in this article, we set $\gamma_{\text{ACI}} = 1$.

5 Test Computation with a Turbocharger Turbine and Exhaust Manifold

5.1 Mesh

The model we use is for a four-cylinder engine and is shown in Fig. 25. Figure 26 shows the quadratic NURBS control mesh. There are 427,371 control points and 285,519 elements. Table 5 shows the number of patches in different parts of the mesh. The thickness of the first layer of elements near the solid surfaces is given in Table 6. Figure 27 shows the four SIs of the mesh. Two of the SIs have an actual slip, and the other two are just for mesh generation purpose and connect nonmatching meshes.

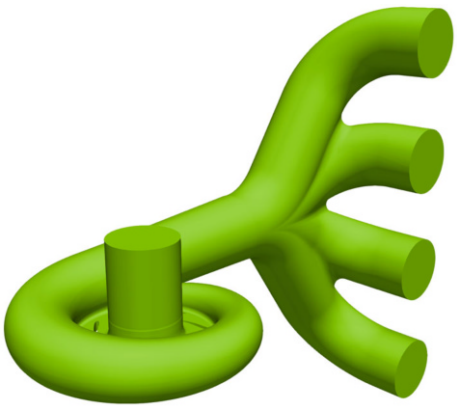


Fig. 25 Turbocharger turbine and exhaust manifold. Model

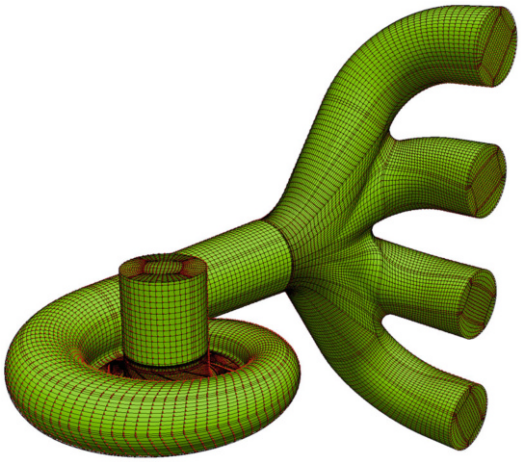


Fig. 26 Turbocharger turbine and exhaust manifold. Quadratic NURBS control mesh

Table 5 Turbocharger turbine and exhaust manifold

Part	Count
Manifold	59
Volute	36
Stator	40
Rotor	160
Outlet pipe	5

Number of patches

Table 6 Turbocharger turbine and exhaust manifold

Part	Range (mm)	
Manifold	0.31	0.98
Volute	0.16	0.58
Stator	0.022	0.20
Rotor	0.037	0.15

Thickness of the first layer of elements near the solid surfaces

5.2 Problem Setup

The rotor diameter is 15 mm and the rotor speed is 220,000 rpm, which translates to a rotation period of $T = 2.7272 \times 10^{-4}$ s. The engine speed is 2000 rpm. The gas density and kinematic viscosity at the manifold inlets are $0.65\,\mathrm{kg/m^3}$ and $5.8 \times 10^{-5}\,\mathrm{m^2/s}$. The flow rate at each manifold inlet is shown in Fig. 28.

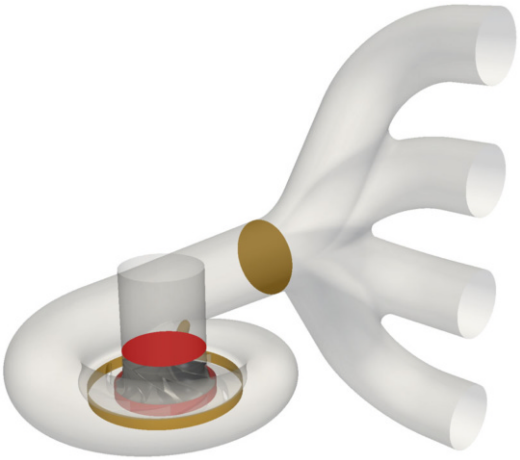


Fig. 27 Turbocharger turbine and exhaust manifold. The SIs of the mesh. The red SIs have an actual slip, and the brown SIs are just for mesh generation purpose and connect nonmatching meshes

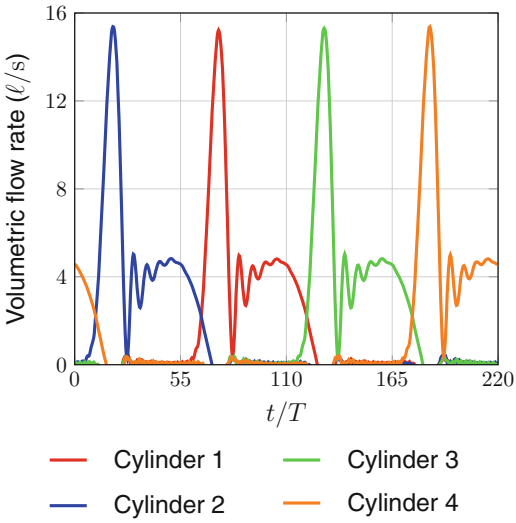


Fig. 28 Turbocharger turbine and exhaust manifold. Volumetric flow rate at each manifold inlet

5.3 Computational Conditions

In temporal representation of the mesh rotation, we again use quadratic NURBS basis functions. There are 90 time steps per rotation, which is equivalent to a time-step size of 3.03×10^{-6} s. The number of nonlinear iterations per time step is 3, and the number of GMRES iterations per nonlinear iteration is 500. The first two nonlinear iterations are based on the ST-SUPS method, and the last iteration is based on the ST-VMS method. We use the stabilization parameters given in [20], which mostly originate from those given in [7, 8, 19].

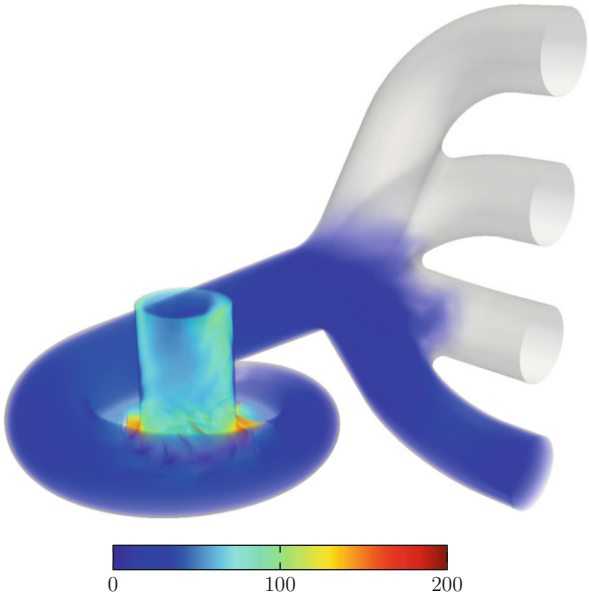


Fig. 29 Turbocharger turbine and exhaust manifold. Velocity magnitude (m/s) at the end of the 4th rotation

5.4 Results

Figure 29 shows the velocity magnitude at the end of the 4th rotation. Figures 30 and 31 show, from the end of the 3rd rotation to the end of the 9th rotation, the volumetric flow rate for the turbine inlet and the turbine efficiency. The efficiency is defined as

$$\eta = \frac{P}{K}, \tag{23}$$

where P is the power extracted from the turbine,

$$K = \frac{1}{T}\int_0^T \int_{\Gamma_{\mathrm{INF}}} (-\mathbf{n}\cdot\mathbf{u})\left(\frac{1}{2}\rho\|\mathbf{u}\|^2 + p - p_{\mathrm{OUTF}}\right) \mathrm{d}\Gamma \mathrm{d}t, \tag{24}$$

Γ_{INF} is the inflow boundary, and p_{OUTF} is the pressure at the outflow boundary.

6 Concluding Remarks

We have presented a general-purpose NURBS mesh generation method for STCA in fluid mechanics with complex geometries. The components of the STCA are the ST-VMS method, which serves as the core method, ST-SI and ST-TC methods, ST-

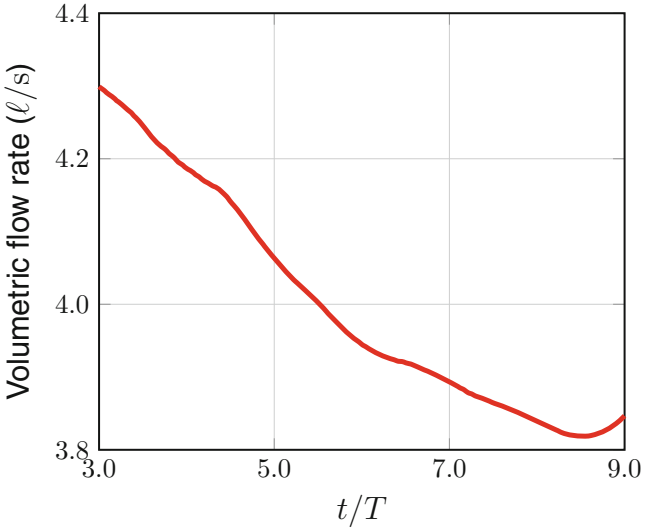


Fig. 30 Turbocharger turbine and exhaust manifold. Volumetric flow rate for the turbine inlet

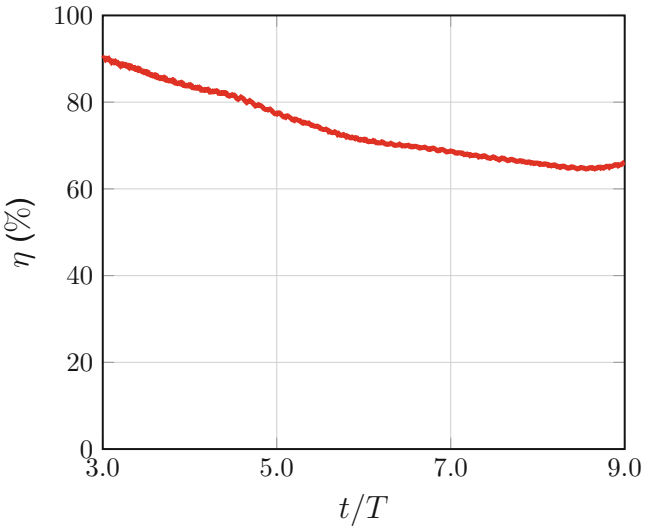


Fig. 31 Turbocharger turbine and exhaust manifold. Turbine efficiency

IGA, and their integration. The ST-IGA, with IGA basis functions in space and time, significantly increases the scope and accuracy of the STCA. The ST-IGA with IGA basis functions in space enables more accurate representation of the surfaces and increased accuracy in the flow solution. It also provides, with relatively few number of elements, reliable discretization in parts of a fluid mechanics domain with

narrowing space, such as near the trailing edge inside a parafoil. The integration of the ST-SI method and ST-IGA enables accurate representation in the presence of spinning solid surfaces, such as turbine rotors. The integration of the ST-IGA, the ST-SI version that weakly enforces the Dirichlet conditions, and the ST-SI version that accounts for the porosity of a thin structure enables accurate flow computations with thin porous structures, such as a parachute. The integration of the ST-SI and ST-TC methods and ST-IGA is the ST-SI-TC-IGA. Beyond enabling a more accurate representation of the surfaces and increased accuracy in the flow solution, the ST-SI-TC-IGA keeps the element density in the narrow spaces near the contact areas at a reasonable level. Considering all these desirable features, the general-purpose NURBS mesh generation method makes the ST-IGA use, and in a wider context the IGA use, even more practical in fluid mechanics computations. The method is based on multi-block-structured mesh generation with existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces. The recovery of the original surfaces is to the extent they are suitable for accurate and robust fluid mechanics computations. The method is expected to retain the refinement distribution and element quality of the multi-block-structured mesh that we start with. Because there are ample good techniques and software for generating multi-block-structured meshes, the method makes general-purpose mesh generation relatively easy. The integration of the ST-SI method and ST-IGA allows, without loss of accuracy, C^{-1} continuity between NURBS patches and thus removes the matching requirement between the patches. Mesh-quality performance studies for 2D and 3D meshes, including those for complex models, and test computation for a turbocharger turbine and exhaust manifold demonstrate that the general-purpose mesh generation method makes the IGA use in fluid mechanics computations even more practical.

Acknowledgements This work was supported (first and second authors) in part by JST-CREST; Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (SIP), "Innovative Combustion Technology" (Funding agency: JST); and Rice–Waseda research agreement. This work was also supported (third author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University.

References

1. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, "Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method", *Computers & Fluids*, **158** (2017) 189–200, https://doi.org/10.1016/j.compfluid.2017.04.017.
2. 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.

3. 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.
4. 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.
5. 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.
6. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44, https://doi.org/10.1016/S0065-2156(08)70153-4.
7. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, https://doi.org/10.1002/fld.505.
8. 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, https://doi.org/10.1002/fld.1430.
9. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", *Computational Mechanics*, **48** (2011) 247–267, https://doi.org/10.1007/s00466-011-0571-z.
10. K. Takizawa and T.E. Tezduyar, "Space–time fluid–structure interaction methods", *Mathematical Models and Methods in Applied Sciences*, **22** (supp02) (2012) 1230001, https://doi.org/10.1142/S0218202512300013.
11. 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, https://doi.org/10.1115/1.4005073.
12. 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, https://doi.org/10.1007/s00466-012-0759-x.
13. 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, https://doi.org/10.1142/S0218202513400058.
14. 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, https://doi.org/10.1007/s00466-013-0888-x.
15. 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, https://doi.org/10.1007/s00466-013-0935-7.
16. K. Takizawa, "Computational engineering analysis with the new-generation space–time methods", *Computational Mechanics*, **54** (2014) 193–211, https://doi.org/10.1007/s00466-014-0999-z.
17. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, "Space–time fluid mechanics computation of heart valve models", *Computational Mechanics*, **54** (2014) 973–986, https://doi.org/10.1007/s00466-014-1046-9.
18. K. Takizawa, T.E. Tezduyar, and A. Buscher, "Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping", *Computational Mechanics*, **55** (2015) 1131–1141, https://doi.org/10.1007/s00466-014-1095-0.
19. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, "Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, https://doi.org/10.1142/S0218202515400072.

20. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, "Space–time VMS method for flow computations with slip interfaces (ST-SI)", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, https://doi.org/10.1142/S0218202515400126.
21. K. Takizawa, T.E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi, "Computational thermo-fluid analysis of a disk brake", *Computational Mechanics*, **57** (2016) 965–977, https://doi.org/10.1007/s00466-016-1272-4.
22. K. Takizawa, T.E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori, "Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA)", *Computers & Fluids*, **142** (2017) 15–20, https://doi.org/10.1016/j.compfluid.2016.02.021.
23. K. Takizawa, T.E. Tezduyar, S. Asada, and T. Kuraishi, "Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC)", *Computers & Fluids*, **141** (2016) 124–134, https://doi.org/10.1016/j.compfluid.2016.05.006.
24. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, "Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods", *Computers & Fluids*, **158** (2017) 176–188, https://doi.org/10.1016/j.compfluid.2016.11.012.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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, https://doi.org/10.1007/s11831-012-9071-3.
32. 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** (supp02) (2012) 1230002, https://doi.org/10.1142/S0218202512300025.
33. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
34. 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, https://doi.org/10.1142/S0218202513400010.
35. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, "New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2217–2226, https://doi.org/10.1142/S0218202515020029.

36. 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, https://doi.org/10.1016/S0045-7825(00)00204-8.
37. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", *Computers and Fluids*, **36** (2007) 12–26.
38. 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.
39. 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.
40. M.-C. Hsu and Y. Bazilevs, "Fluid–structure interaction modeling of wind turbines: simulating the full machine", *Computational Mechanics*, **50** (2012) 821–833.
41. M.E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, A.L. Marsden, and M. of Congenital Hearts Alliance (MOCHA), "A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations", *Computational Mechanics*, **48** (2011) 277–291, https://doi.org/10.1007/s00466-011-0599-0.
42. 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, https://doi.org/10.1002/fld.2400.
43. 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.
44. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "High-performance computing of wind turbine aerodynamics using isogeometric analysis", *Computers and Fluids*, **49** (2011) 93–100.
45. Y. Bazilevs, M.-C. Hsu, and M.A. Scott, "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines", *Computer Methods in Applied Mechanics and Engineering*, **249–252** (2012) 28–41.
46. 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.
47. 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.
48. Y. Bazilevs, K. Takizawa, T.E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre, "Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods", *Archives of Computational Methods in Engineering*, **21** (2014) 359–398, https://doi.org/10.1007/s11831-014-9119-7.
49. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines", *International Journal for Numerical Methods in Engineering*, **102** (2015) 766–783, https://doi.org/10.1002/nme.4738.
50. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Aerodynamic simulation of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2013) 021011, https://doi.org/10.1115/1.4024415.
51. Y. Bazilevs, A. Korobenko, X. Deng, J. Yan, M. Kinzel, and J.O. Dabiri, "FSI modeling of vertical-axis wind turbines", *Journal of Applied Mechanics*, **81** (2014) 081006, https://doi.org/10.1115/1.4027466.

52. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs, "Computational free-surface fluid–structure interaction with application to floating offshore wind turbines", *Computers and Fluids*, **141** (2016) 155–174, https://doi.org/10.1016/j.compfluid.2016.03.008.
53. Y. Bazilevs, A. Korobenko, J. Yan, A. Pal, S.M.I. Gohari, and S. Sarkar, "ALE–VMS formulation for stratified turbulent incompressible flows with applications", *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2349–2375, https://doi.org/10.1142/S0218202515400114.
54. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, "FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades", *Journal of Applied Mechanics*, **83** (6) (2016) 061010.
55. 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.
56. 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.
57. 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.
58. 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.
59. 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.
60. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Fluid–structure interaction simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **52** (2013) 971–981, https://doi.org/10.1007/s00466-013-0858-3.
61. C.C. Long, M. Esmaily-Moghadam, A.L. Marsden, and Y. Bazilevs, "Computation of residence time in the simulation of pulsatile ventricular assist devices", *Computational Mechanics*, **54** (2014) 911–919, https://doi.org/10.1007/s00466-013-0931-y.
62. C.C. Long, A.L. Marsden, and Y. Bazilevs, "Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk", *Computational Mechanics*, **54** (2014) 921–932, https://doi.org/10.1007/s00466-013-0967-z.
63. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation", *Computational Mechanics*, **54** (2014) 1055–1071, https://doi.org/10.1007/s00466-014-1059-4.
64. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks, "Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models", *Computational Mechanics*, **55** (2015) 1211–1225, https://doi.org/10.1007/s00466-015-1166-x.
65. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 1005–1053.
66. 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.

67. 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.
68. C. Wang, M.C.H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs, "Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis", *Computers and Fluids*, **142** (2017) 3–14, https://doi.org/10.1016/j.compfluid.2015.12.004.
69. M.C.H. Wu, D. Kamensky, C. Wang, A.J. Herrema, F. Xu, M.S. Pigazzini, A. Verma, A.L. Marsden, Y. Bazilevs, and M.-C. Hsu, "Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear", *Computer Methods in Applied Mechanics and Engineering*, (2017), Published online. https://doi.org/10.1016/j.cma.2016.09.032.
70. J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs, "Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines", *Computers and Fluids*, **158** (2017) 157–166, https://doi.org/10.1016/j.compfluid.2016.06.016.
71. B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "Experimental and numerical FSI study of compliant hydrofoils", *Computational Mechanics*, **55** (2015) 1079–1090, https://doi.org/10.1007/s00466-014-1090-5.
72. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration", *Computers and Fluids*, **141** (2016) 201–211, https://doi.org/10.1016/j.compfluid.2015.07.013.
73. K. Takizawa and T.E. Tezduyar, "Computational methods for parachute fluid–structure interactions", *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, https://doi.org/10.1007/s11831-012-9070-4.
74. 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, https://doi.org/10.1007/s00466-012-0761-3.
75. 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, https://doi.org/10.1007/s00466-013-0880-5.
76. K. Takizawa, T.E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel, "FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1203–1220, https://doi.org/10.1007/s00466-014-1052-y.
77. K. Takizawa, T.E. Tezduyar, R. Kolesar, C. Boswell, T. Kanai, and K. Montel, "Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes", *Computational Mechanics*, **54** (2014) 1461–1476, https://doi.org/10.1007/s00466-014-1069-2.
78. K. Takizawa, T.E. Tezduyar, C. Boswell, Y. Tsutsui, and K. Montel, "Special methods for aerodynamic-moment calculations from parachute FSI modeling", *Computational Mechanics*, **55** (2015) 1059–1069, https://doi.org/10.1007/s00466-014-1074-5.
79. K. Takizawa, T.E. Tezduyar, and R. Kolesar, "FSI modeling of the Orion spacecraft drogue parachutes", *Computational Mechanics*, **55** (2015) 1167–1179, https://doi.org/10.1007/s00466-014-1108-z.
80. 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, https://doi.org/10.1007/s00466-011-0589-2.
81. 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, https://doi.org/10.1007/s00466-011-0614-5.

82. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, M.-C. Hsu, O. Øiseth, K.M. Mathisen, N. Kostov, and S. McIntyre, "Engineering analysis and design with ALE-VMS and space–time methods", *Archives of Computational Methods in Engineering*, **21** (2014) 481–508, https://doi.org/10.1007/s11831-014-9113-0.
83. 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, https://doi.org/10.1007/s00466-012-0758-y.
84. 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, https://doi.org/10.1016/j.compfluid.2012.11.008.
85. K. Takizawa, T.E. Tezduyar, and N. Kostov, "Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV", *Computational Mechanics*, **54** (2014) 213–233, https://doi.org/10.1007/s00466-014-0980-x.
86. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent", *Computational Mechanics*, **50** (2012) 675–686, https://doi.org/10.1007/s00466-012-0760-4.
87. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, "Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms", *Computational Mechanics*, **51** (2013) 1061–1073, https://doi.org/10.1007/s00466-012-0790-y.
88. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt, "ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling", *Mathematical Models and Methods in Applied Sciences*, **24** (2014) 2437–2486, https://doi.org/10.1142/S0218202514500250.
89. H. Suito, K. Takizawa, V.Q.H. Huynh, D. Sze, and T. Ueda, "FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta", *Computational Mechanics*, **54** (2014) 1035–1045, https://doi.org/10.1007/s00466-014-1017-1.
90. K. Takizawa, D. Montes, S. McIntyre, and T.E. Tezduyar, "Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers", *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 223–248, https://doi.org/10.1142/s0218202513400022.
91. K. Takizawa, T.E. Tezduyar, and H. Hattori, "Computational analysis of flow-driven string dynamics in turbomachinery", *Computers & Fluids*, **142** (2017) 109–117, https://doi.org/10.1016/j.compfluid.2016.02.019.
92. K. Takizawa, T.E. Tezduyar, and T. Terahara, "Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)", *Computers & Fluids*, **141** (2016) 191–200, https://doi.org/10.1016/j.compfluid.2016.05.027.
93. K. Takizawa, T.E. Tezduyar, and T. Kanai, "Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity", *Mathematical Models and Methods in Applied Sciences*, **27** (2017) 771–806, https://doi.org/10.1142/S0218202517500166.
94. 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, https://doi.org/10.1016/0045-7825(94)00082-4.
95. K. Takizawa and T.E. Tezduyar, "Space–time computation techniques with continuous representation in time (ST-C)", *Computational Mechanics*, **53** (2014) 91–99, https://doi.org/10.1007/s00466-013-0895-y.
96. T.E. Tezduyar and D.K. Ganjoo, "Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion problems", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 49–71, https://doi.org/10.1016/0045-7825(86)90023-X.

97. G.J. Le Beau, S.E. Ray, S.K. Aliabadi, and T.E. Tezduyar, "SUPG finite element computation of compressible flows with the entropy and conservation variables formulations", *Computer Methods in Applied Mechanics and Engineering*, **104** (1993) 397–422, https://doi.org/10.1016/0045-7825(93)90033-T.
98. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, https://doi.org/10.1016/j.compfluid.2005.02.011.
99. 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, https://doi.org/10.1016/j.cma.2005.05.032.
100. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", *Computers & Fluids*, **36** (2007) 147–159, https://doi.org/10.1016/j.compfluid.2005.07.009.
101. 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, https://doi.org/10.1007/s00466-005-0025-6.
102. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, https://doi.org/10.1007/s00466-006-0056-7.
103. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364, https://doi.org/10.1007/s00466-006-0045-x.
104. 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, https://doi.org/10.1016/j.compfluid.2005.07.004.
105. 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, https://doi.org/10.1002/fld.1743.
106. 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, https://doi.org/10.1002/fld.1447.
107. 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, https://doi.org/10.1002/fld.1484.
108. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms", *Journal of Applied Mechanics*, **76** (2009) 021211, https://doi.org/10.1115/1.3062967.
109. 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, https://doi.org/10.1115/1.3057496.
110. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167, https://doi.org/10.1007/s00466-009-0441-0.
111. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "Improving stability of stabilized and multiscale formulations in flow simulations at small time steps", *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840, https://doi.org/10.1016/j.cma.2009.06.019.
112. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270, https://doi.org/10.1002/fld.2451.

113. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique", *Journal of Applied Mechanics*, **79** (2012) 010910, https://doi.org/10.1115/1.4005060.
114. A. Corsini, F. Rispoli, A.G. Sheard, and T.E. Tezduyar, "Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations", *Computational Mechanics*, **50** (2012) 695–705, https://doi.org/10.1007/s00466-012-0789-4.
115. P.A. Kler, L.D. Dalcin, R.R. Paz, and T.E. Tezduyar, "SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems", *Computational Mechanics*, **51** (2013) 171–185, https://doi.org/10.1007/s00466-012-0712-z.
116. A. Corsini, F. Rispoli, A.G. Sheard, K. Takizawa, T.E. Tezduyar, and P. Venturini, "A variational multiscale method for particle-cloud tracking in turbomachinery flows", *Computational Mechanics*, **54** (2014) 1191–1202, https://doi.org/10.1007/s00466-014-1050-0.
117. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T.E. Tezduyar, "Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing", *Computational Mechanics*, **55** (2015) 1201–1209, https://doi.org/10.1007/s00466-015-1160-3.
118. M.F. Wheeler, "An elliptic collocation-finite element method with interior penalties", *SIAM Journal on Numerical Analysis*, **15** (1978) 152–161.
119. P. Houston, C. Schwab, and E. Suli, "Discontinuous hp-finite element methods for advection-diffusion reaction problems", *SIAM Journal on Numerical Analysis*, **39** (2002) 2133–2163.

Interface-Reproducing Capturing (IRC) Technique for Fluid-Structure Interaction: Methods and Applications

Tomohiro Sawada

Abstract How to enhance the interface-capturing (IC) computation of fluid–structure interaction (FSI) is a long-standing issue for IC approaches. This chapter introduces approaches based on an extended finite element method (XFEM) and a Lagrange multiplier (LM) method, as well as our contribution to the problem. The XFEM-based approach develops a framework for an interface-reproducing capturing (IRC) method whose spatial functions are locally equivalent to those of interface-tracking (IT) methods. The XFEM enriches the velocity and pressure function spaces of the local flow around the interface. This enrichment reproduces requisite discontinuities at the interface. Simultaneously, the LM method imposes continuity on the fluid and structure to couple them, and thus the fluid captures the interface. This chapter gives an overview, describes the methods and solution techniques, and shows verifications and applications, focusing mainly on computing the fluid–thin-structure interaction (FTSI). The verifications reveal how continuity and discontinuity at the interface affect the FSI computation and why the IRC method is effective. Applications to flow-induced flutter of flexible thin objects show the ability of the proposed method to take on the challenge of computing complex FSI problems. Applications to flows past fixed objects show its ability to compute simple problems with ease. The IRC method therefore has two aspects and potentials. Open issues mentioned in this chapter indicate that there is still much room for advancing the IC method.

T. Sawada (✉)
Integrated Macroscopic Simulation Team, Research Center for Computational Design of Advanced Functional Materials, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki, Japan
e-mail: tomohiro-sawada@aist.go.jp

T. E. Tezduyar (ed.), *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-319-96469-0_11

1 Introduction

Numerical methods for simulating fluid–structure interactions (FSI) [1–96] draw a great deal of interest from various areas of computation and application. One reason for this interest is that FSI methods and related techniques offer a basis for continuum mechanics-based modeling and computation of various phenomena. The method of handling material and spatial, namely Lagrangian and Eulerian specifications of the computational FSI field simultaneously or alternatively, which has been studied and advanced in the FSI area, is universal for any subjects. This universality has been enhanced by interface-tracking (IT) [4–18, 20–53] and interface-capturing (IC) [19, 28, 32–34, 37, 54–73, 75–96] approaches.

The IT approach generally uses interface-fitted coordinates and function spaces to track the interface between a fluid and a structure, as represented by arbitrary Lagrangian–Eulerian (ALE) [1–3] FSI methods [4–18, 47, 52, 53] and stabilized space-time (SST) FSI methods [20–53]. In particular, the SST-based approach [20–53] has advanced many essential techniques such as interface-matching techniques, mesh-moving techniques, and solution techniques and stabilization techniques [97–109] that are universal for all FSI methods, and it has facilitated various advanced applications. Interface tracking is the standard way to verify FSI computation. In contrast, the IC approach generally uses spatially fixed coordinates with a technique to capture moving structures by the fixed coordinates. That technique is characteristic to each IC method and so there are various IC FSI methods [19, 28, 32–34, 37, 54–73, 75–96]. A well-known one is the immersed boundary (IB) method [54–61] in which structures in a fluid are described as part of the fluid via the incompressible Navier–Stokes equations. Interface-capturing methods have the advantage of being applicable to complex FSI problems because of the spatially fixed coordinates, but they suffer in relation to theoretical compatibility and resultant numerical accuracy, stability, and credibility. These drawbacks arise from the non-interfacially fitted spatial coordinates and function spaces. The point at issue for IC methods is how to solve or reduce the inconsistency arisen from non-interface-fitted coordinates and enhance the utility of IC computation.

Based on efforts made over past decades, an IC approach has been proposed that addresses the aforementioned issue directly [76–78]. It is based on an extended finite element method (XFEM) [74, 110–114, 116] and a Lagrange multiplier (LM) method [115–118]. The XFEM is a way to reproduce arbitrary functions (including discontinuous ones) in a domain with finite support. For example, partition of unity of XFEM can formulate FSI without the fictitious fluid domain that is introduced in computational FSI models based on the IB method [54–61], the immersed finite element (IFE) method [62–65], or the distributed Lagrange multiplier/fictitious domain (DLM/FD) method [66–71]. Furthermore, an enriched form of XFEM can reproduce discontinuities at the interface, something that is intrinsic to IT methods but not to general IC methods. In FSI and fluid dynamics generally, XFEM was applied first to Stokes flow around rigid particles [72, 73], then to two-phase flow [74], and subsequently to flow in a microstructure [75] with a

homogenization theory for fluid–solid mixtures. A version of XFEM for FSI is proposed in [76–78] with extension to the enriched space-time (EST) method [80]. Further advancements are made in [79, 81–90] with consideration of fluid–solid and fluid–thin-solid interactions [81, 83], fluid–shell interaction [79, 86, 88], partitioned iterative schemes [81], numerical integration [83, 86], discretization of LMs [84, 86], and contact [85], among others. A combination of XFEM and the IFE method [63, 64] is given in [65]. These studies develop a way of giving a consistent FSI interface to the IC computation. The XFEM *reproduces* the requisite discontinuities at the FSI interface, and the LM method embeds that interface in a fluid mesh that *captures* the interface. By these means, XFEM-based methods provide a framework for the *interface-reproducing capturing (IRC) method* as an advanced IC method.

This chapter describes the IRC FSI method and its applications, focusing mainly on the method that we proposed in [79, 86, 88]. Section 2 describes the governing equations for FSI and flow problems in strong and weak forms, where the FSI coupling condition at the interface is formulated weakly by the LM method. This weak form is universal for IT, IC, and IRC methods. In Sect. 3, we describe the XFEM for FSI interfaces, which covers fluid–thin-structure interactions (FTSI) [79, 86–90], fluid–solid interactions [81–85], and flow around fixed objects [75, 87]. Some remarks are made on enrichments such as numerical integration of enriched elements [86]. In Sect. 4, we describe the finite element discretization of the LMs and the constraint equation [86]. Because the IRC method discretizes these on a Lagrangian mesh, we refer to the present LM method as the Lagrangian LM (LLM) method. The LLM technique removes the need for interface-fitting background procedures in the present IRC method in combination with quadrature of enriched elements. Section 5 summarizes the spatial discretization techniques used for FTSI computation [88]. Section 6 describes a time-integration algorithm for the IRC method [88], which employs an implicit scheme based on the Newmark-β method [119] and a quasi-direct solution technique.

In Sect. 7, we verify the IRC computation on a domain-partitioning interface (DPI) problem [79, 86, 88]. The DPI problem assesses the accuracy of the numerical integration of enriched elements [86], of the weak imposition of the FSI coupling condition by the LLM technique [86], and of the enriched discontinuities [88]. Furthermore, we show how the enrichment and discontinuity affect the FSI computation. The results indicate which terms are important when we compute FSI on an IC mesh. Section 8 shows two-dimensional (2D) [88] and three-dimensional (3D) [87, 90] applications of the IRC method to FTSI problems. In particular, we introduce the fluttering/flapping of flexible thin objects in a flow. The FSI phenomena are rich [120–128] and their numerical simulation remains a challenge. We compare the IRC results with those obtained by the IT ALE FSI method [12, 13]. The 3D applications to flag flapping [90] and paper fluttering [87] show the ability of the IRC approach to handle large deformations and complex geometry, and they demonstrate its potential for further application. In Sect. 9, we show applications to flows past fixed objects [89] and in a periodic microstructure [75] to introduce another aspect of the IRC method. Section 10 concludes the chapter.

2 Governing Equations in Strong and Weak Forms

In this section, we introduce the governing equations for the fluid and structures to be addressed by the present method, followed by those for interactions and moving boundaries. The basis of our IRC method for FSI and flow with moving boundaries is given by Eqs. (20) and (21), which are written in weak form and for which we used an LM method [76, 86, 115–118] for the fluid–structure coupling.

2.1 Fluid Flows

Let Ω_f be the fluid domain with boundary Γ_f, and let $\mathbf{x}$ and t be the spatial coordinates and time, respectively. We describe the Navier–Stokes equations governing fluid flows in an Eulerian frame with $\mathbf{x}$ fixed in space. Under the assumption of incompressibility, the Navier–Stokes equations are

$$\rho_f \left(\left. \frac{\partial \mathbf{v}}{\partial t} \right|_x + \mathbf{v} \cdot \nabla_x \mathbf{v} \right) = \nabla_x \cdot \mathbf{T}_f + \rho_f \mathbf{g}, \tag{1}$$

$$\nabla_x \cdot \mathbf{v} = 0, \tag{2}$$

where ρ_f, $\mathbf{v}$, $\mathbf{T}_f$, and $\mathbf{g}$ are the density, velocity, Cauchy stress, and the body forces, respectively, and $\partial(\cdot)/\partial t|_x$ and ∇_x denote the time-derivative and gradient operators, respectively, with respect to the Eulerian coordinates $\mathbf{x}$. We assume that the Cauchy stress is given in Newtonian form:

$$\mathbf{T}_f = -p\mathbf{I} + 2\mu_f \mathbf{D}, \tag{3}$$

where p and μ_f are the pressure and the viscosity, respectively, $\mathbf{I}$ is the identity tensor, and $\mathbf{D}$ is the deformation rate tensor given by

$$\mathbf{D} = \frac{1}{2} \left\{ (\nabla_x \mathbf{v}) + (\nabla_x \mathbf{v})^T \right\}. \tag{4}$$

The Dirichlet and Neumann boundary conditions given at the corresponding subsets of Γ_f, namely $\Gamma_{f\underline{v}}$ and $\Gamma_{f\underline{t}}$, are written, respectively, as

$$\mathbf{v} - \underline{\mathbf{v}} = \mathbf{0} \quad \text{at } \Gamma_{f\underline{v}}, \tag{5}$$

$$\mathbf{T}_f \cdot \mathbf{n}_f - \underline{\mathbf{t}}_f = \mathbf{0} \quad \text{at } \Gamma_{f\underline{t}}, \tag{6}$$

where $\underline{\mathbf{v}}$ and $\underline{\mathbf{t}}_f$ are given functions and $\mathbf{n}_f$ is the unit normal vector of the boundary pointing outward. Equation (5) denotes that the flow is subject to the no-slip condition at the boundary.

Let δ be the variational operator to the unknown functions and their functionals. The weak form of the boundary value problem is given by

$$\delta W_f = 0, \tag{7}$$

where δW_f is the virtual power given by

$$\delta W_f = \int_{\Omega_f} \delta \mathbf{v} \cdot \rho_f \left(\left. \frac{\partial \mathbf{v}}{\partial t} \right|_x + \mathbf{v} \cdot \nabla_x \mathbf{v} - \mathbf{g} \right) d\Omega + \int_{\Omega_f} \delta \mathbf{D} : \mathbf{T}_f d\Omega - \int_{\Gamma_{f\underline{t}}} \delta \mathbf{v} \cdot \underline{\mathbf{t}}_f d\Gamma + \int_{\Omega_f} \delta p \, (\nabla_x \cdot \mathbf{v}) \, d\Omega, \tag{8}$$

where $\delta \mathbf{v}$ and δp are the virtual velocity and pressure, respectively, and $\delta \mathbf{D}$ is

$$\delta \mathbf{D} = \frac{1}{2} \left\{ (\nabla_x \delta \mathbf{v}) + (\nabla_x \delta \mathbf{v})^T \right\}. \tag{9}$$

2.2 Structural Deformations

Let Ω_{0s} and Ω_s be the initial and current domains of the structures, respectively, with Γ_{0s} and Γ_s the corresponding boundaries, respectively. Let $\mathbf{X}$ be the material (i.e., Lagrangian) coordinates of the structures. We use the Cauchy equilibrium equation to describe the deformation of a structure relative to its initial configuration:

$$\rho_{0s} \left. \frac{\partial^2 \mathbf{u}}{\partial t^2} \right|_X = \nabla_X \cdot \left(\mathbf{S}_s \cdot \mathbf{F}^T \right) + \rho_{0s} \mathbf{g}, \tag{10}$$

where ρ_{0s} is the nominal density of the structure, $\mathbf{u}$ is the displacement, $\mathbf{S}_s$ and $\mathbf{F}$ are the second Piola–Kirchhoff stress and deformation gradient tensors, respectively, and $\partial(\cdot)/\partial t|_X$ and ∇_X are the time-derivative and gradient operators, respectively, with respect to the Lagrangian coordinates $\mathbf{X}$. We assume that the stress tensor is given in Saint-Venant form:

$$\mathbf{S}_s = \lambda_s \, (\mathrm{tr}\mathbf{E}) \, \mathbf{I} + 2\mu_s \mathbf{E}, \tag{11}$$

where λ_s and μ_s are Lamé constants and $\mathbf{E}$ is the Green–Lagrange strain tensor given by

$$\mathbf{E} = \frac{1}{2} \left\{ (\nabla_X \mathbf{u}) + (\nabla_X \mathbf{u})^T + (\nabla_X \mathbf{u}) \cdot (\nabla_X \mathbf{u})^T \right\}. \tag{12}$$

The Dirichlet and Neumann boundary conditions given at the corresponding boundaries $\Gamma_{s\underline{u}}$ and $\Gamma_{s\underline{t}}$ are written, respectively, as

$$\mathbf{u} - \underline{\mathbf{u}} = \mathbf{0} \quad \text{at } \Gamma_{s\underline{u}}, \tag{13}$$

$$\mathbf{T}_s \cdot \mathbf{n}_s - \underline{\mathbf{t}}_s = \mathbf{0} \quad \text{at } \Gamma_{s\underline{t}}, \tag{14}$$

where $\underline{\mathbf{u}}$ and $\underline{\mathbf{t}}_s$ are given functions, $\mathbf{T}_s$ is the Cauchy stress, and $\mathbf{n}_s$ is the outward normal vector of the boundary.

We describe the weak form of the boundary value problem as

$$\delta W_s = 0, \tag{15}$$

where δW_s is the virtual power given by

$$\delta W_s = \int_{\Omega_{0s}} \delta\dot{\mathbf{u}} \cdot \rho_{0s} \left(\left. \frac{\partial^2 \mathbf{u}}{\partial t^2} \right|_X - \mathbf{g} \right) d\Omega + \int_{\Omega_{0s}} \delta\dot{\mathbf{E}} : \mathbf{S}_s d\Omega - \int_{\Gamma_{s\underline{t}}} \delta\dot{\mathbf{u}} \cdot \underline{\mathbf{t}}_s d\Gamma, \tag{16}$$

where the dot above $\dot{\mathbf{u}}$ and $\dot{\mathbf{E}}$ denotes the Lagrangian time derivative. Thus, $\delta\dot{\mathbf{u}}$ is the virtual displacement rate and $\delta\dot{\mathbf{E}}$ is given by:

$$\delta\dot{\mathbf{E}} = \frac{1}{2} \left\{ (\nabla_X \delta\dot{\mathbf{u}}) + (\nabla_X \delta\dot{\mathbf{u}})^T + (\nabla_X \delta\dot{\mathbf{u}}) \cdot (\nabla_X \mathbf{u})^T + (\nabla_X \mathbf{u}) \cdot (\nabla_X \delta\dot{\mathbf{u}})^T \right\}. \tag{17}$$

2.3 FSI and Flow with Moving Boundaries

(i) FSI with No Slip We begin by considering the frequently used no-slip interface as our fluid–structure interface, denoted by Γ_i. The FSI conditions for the no-slip interface are given by

$$\mathbf{v} - \dot{\mathbf{u}} = \mathbf{0} \quad \text{at } \Gamma_i, \tag{18}$$

$$-\mathbf{T}_f \cdot \mathbf{n}_i + \mathbf{T}_s \cdot \mathbf{n}_i = \mathbf{0} \quad \text{at } \Gamma_i, \tag{19}$$

where $\mathbf{n}_i$ is the normal vector of the interface and is defined by $\mathbf{n}_i = \mathbf{n}_s = -\mathbf{n}_f$.

We use an LM method to impose the FSI conditions weakly on the fluid and the structures [76, 86, 118], giving a monolithic variational (weak) formulation of the FSI problem:

$$\delta W_f + \delta W_s + \delta W_i = 0, \tag{20}$$

where δW_f and δW_s are given by Eqs. (8) and (16), respectively, and δW_i is the weak form of the above FSI conditions and is written as

$$\delta W_i = \int_{\Gamma_i} \delta\boldsymbol{\lambda} \cdot (\mathbf{v} - \dot{\mathbf{u}})\, d\Gamma + \int_{\Gamma_i} (\delta\mathbf{v} - \delta\dot{\mathbf{u}}) \cdot \boldsymbol{\lambda} d\Gamma, \tag{21}$$

where $\boldsymbol{\lambda}$ is the LM function that corresponds to the coupling force at Γ_i and yields

$$\mathbf{T}_f \cdot \mathbf{n}_i = \mathbf{T}_s \cdot \mathbf{n}_i = \boldsymbol{\lambda} \quad \text{at } \Gamma_i \tag{22}$$

in strong form (see [76]). Therefore, Eq. (21) ensures velocity continuity and force balance simultaneously.

(ii) FSI with Slip Because finely adjusting the strong and corresponding weak forms of the FSI conditions provides various interfacial cases, we describe some useful cases below. For an interface with slip but no friction, velocity continuity is written as a scalar constraint, namely

$$(\mathbf{v} - \dot{\mathbf{u}}) \cdot \mathbf{n}_i = 0 \quad \text{at } \Gamma_i, \tag{23}$$

and consequently Eq. (21) becomes

$$\delta W_i = \int_{\Gamma_i} (\delta\lambda_n \mathbf{n}_i) \cdot (\mathbf{v} - \dot{\mathbf{u}})\, d\Gamma + \int_{\Gamma_i} (\delta\mathbf{v} - \delta\dot{\mathbf{u}}) \cdot (\lambda_n \mathbf{n}_i)\, d\Gamma, \tag{24}$$

where λ_n is the scalar LM function that forces continuity of the fluid and structural velocities at Γ_i in the normal direction and forms a pressure jump at Γ_i.

If the interface has friction, we can decompose the vectorial LM $\boldsymbol{\lambda}$ into a normal component $\lambda_n \mathbf{n}_i$ and tangential components $\lambda_t \mathbf{t}_i$. This decomposition provides an augmented LM method with penalty terms when the tangential components are projected formally onto the velocity function spaces:

$$\begin{aligned} \delta W_i = & \int_{\Gamma_i} (\delta\lambda_n \mathbf{n}_i) \cdot (\mathbf{v} - \dot{\mathbf{u}})\, d\Gamma + \int_{\Gamma_i} (\delta\mathbf{v} - \delta\dot{\mathbf{u}}) \cdot (\lambda_n \mathbf{n}_i)\, d\Gamma \\ & + \int_{\Gamma_i} (\delta\mathbf{v} - \delta\dot{\mathbf{u}}) \cdot \alpha\, (\mathbf{v} - \dot{\mathbf{u}})\, d\Gamma, \end{aligned} \tag{25}$$

where α is the penalty parameter that models friction. The relative velocity in the third term of Eq. (25) is the tangential velocity in practice because the first term forces the relative velocity onto Γ_i (see [15, 18, 19]).

(iii) Weak Imposition of Dirichlet Boundary Conditions For flow with a boundary or interface that moves according to a given displacement-rate function $\underline{\dot{\mathbf{u}}}(\mathbf{X}, t)$, we can rewrite Eq. (20) as

$$\delta W_f + \delta W_{i\underline{\dot{u}}} = 0, \tag{26}$$

with the virtual power $\delta W_{i\underline{\dot{u}}}$ being

$$\delta W_{i\underline{\dot{u}}} = \int_{\Gamma_{i\underline{\dot{u}}}} \delta\boldsymbol{\lambda} \cdot \left(\mathbf{v} - \underline{\dot{\mathbf{u}}}(\mathbf{X}, t)\right) d\Gamma + \int_{\Gamma_{i\underline{\dot{u}}}} \delta\mathbf{v} \cdot \boldsymbol{\lambda} d\Gamma, \tag{27}$$

where $\Gamma_{i\underline{\dot{u}}}$ denotes the moving interface or boundary. In that case, $\boldsymbol{\lambda}$ becomes the reaction force from $\Gamma_{i\underline{\dot{u}}}$ to the flow. Equation (27) covers naturally the weak imposition of Dirichlet boundary conditions of Eq. (5), where we mean that the boundary does not move but the velocity of the flow is specified by a given function $\underline{\mathbf{v}}(\mathbf{x}, t)$ at the boundary. In that case, we are able to replace $\underline{\dot{\mathbf{u}}}(\mathbf{X}, t)$ in Eq. (27) by $\underline{\mathbf{v}}(\mathbf{x}, t)$ to give

$$\delta W_{i\underline{v}} = \int_{\Gamma_{i\underline{v}}} \delta\boldsymbol{\lambda} \cdot \left(\mathbf{v} - \underline{\mathbf{v}}(\mathbf{x}, t)\right) d\Gamma + \int_{\Gamma_{i\underline{v}}} \delta\mathbf{v} \cdot \boldsymbol{\lambda} d\Gamma. \tag{28}$$

where $\Gamma_{i\underline{v}}$ can also be written as $\Gamma_{f\underline{v}}$.

3 Extended Finite Element Method for FSI Interfaces

The weak formulations of FSI conditions given by Eq. (21) or its variants enable a fluid mesh to capture interfaces between structures while enabling structural meshes to embed interfaces in a fluid mesh. In a broad sense, the weak formulation is considered a technique for imposing fluid and structural continuity on the respective equations. However, IC meshes do not have the interface-fitted function spaces that IT meshes have with proper smoothness and discontinuity by nature. This means that the capturing meshes cannot approximate boundary layers smoothly and sharply, nor can they reproduce interfacial discontinuities with sufficient smoothness and sharpness. These deficiencies decrease the quality and credibility of computational results and convergence. This section describes the XFEM for FSI interfaces as a solution to these issues. To the best of our knowledge, Legay and Zilian (née Kölke) [76–78, 80], Gerstenberger [81–85], and Sawada [75, 79, 86–90] are the pioneers in this area. As mentioned in Sect. 1, the methods proposed by those authors can be referred to collectively as the IRC method or technique for FSI and flow with moving boundaries. This is because the fluid elements crossed by the interface not only capture the interface but also reproduce the discontinuity.

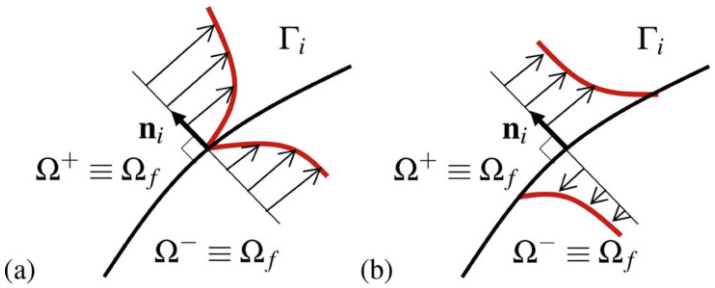


Fig. 1 Illustration of spatial profiles of (**a**) velocity and (**b**) pressure at and near a no-slip interface between a fluid and a thin structure. One way to approximate such a profile is to combine weak and strong discontinuities

3.1 Fluid–Shell Interactions

We begin by considering FTSI, in which a thin structure is modeled as a 2D (resp. 1D) surface in a 3D (resp. 2D) flow by structural elements such as shells, membranes, and beams, and the interface Γ_i is approximated by a neutral plane of structural elements [129]. In such cases, Γ_i partitions the fluid flow physically into flows on either side of Γ_i, and this partitioning requires discontinuity for the finite element function spaces of the flow. For no-slip interfaces, the discontinuity is described by a weak (i.e., gradient) discontinuity of velocity and a strong discontinuity of pressure, as illustrated in Fig. 1a, b, respectively [77]; for slip interfaces, the discontinuity is described by a strong discontinuity for both. If the discontinuities are not given to the function spaces of flow or if continuous fluid elements are employed without care,[1] flows on both sides can become coupled. This coupling becomes worse and more apparent with Reynolds number and is observed as, for example, penetration, inverse flow, artificial vortices, and underestimation of pressure and FSI forces.

(i) Enrichments for No-Slip Interface To give the proper discontinuity to the flow function spaces at Γ_i, we enrich the finite element function spaces of velocity and pressure locally at the elements crossed by Γ_i with a framework of XFEM enrichment [110–112]. Figure 2 shows a representative geometry for the fluid and structural meshes. Let Q_{fi} represent the set of nodes covering the fluid elements crossed by Γ_i, let Q_f represent all nodes including Q_{fi}, and let us suppose the

[1] An elegant technique and concept proposed by Tezduyar and named FSILT-ED (fluid–solid interface locator technique-extended domain) can be found in [28, 33, 34, 37]. FSILT-ED applies a stabilized LM method or a penalty method at the interface, with an elaborate treatment of the interpolation of pressure across the interface, introducing fluid domains extended virtually from one side of the interface to the other. We consider the ED technique as a way to extrapolate from one side to the interface in a finite element. See [28, 33, 34, 37].

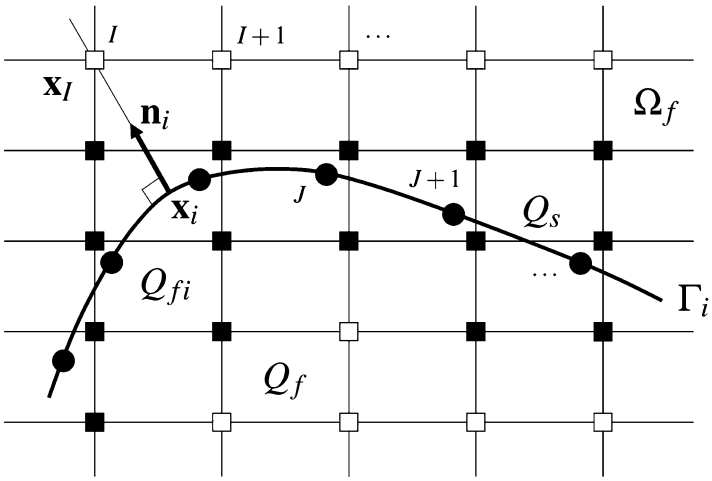


Fig. 2 Interface-capturing fluid mesh and interface-embedding Lagrangian mesh of a thin structure (filled square: fluid nodes enriched; open square: fluid nodes not enriched; filled circle: Lagrangian nodes of structure)

elements to be Q1Q1 elements that have bi-linear shape functions for velocity and pressure. The enrichments for the no-slip interface are written as

$$\mathbf{v}(\mathbf{x},t) = \sum_{I \in Q_f} N_I \mathbf{V}_I + e(\mathbf{x},t) \sum_{I \in Q_{fi}} N_I \tilde{\mathbf{V}}_I, \tag{29}$$

$$p(\mathbf{x},t) = \sum_{I \in Q_f} N_I P_I + h(\mathbf{x},t) \sum_{I \in Q_{fi}} N_I \tilde{P}_I, \tag{30}$$

where N_I is the Q1 shape function of fluid node I, $\mathbf{V}_I$ and P_I are the usual nodal unknowns for the velocity and pressure, respectively, and $\tilde{\mathbf{V}}_I$ and $\tilde{P}_I$ are the new unknowns for the enrichments. In Fig. 2, the fluid nodes that have the new unknowns (i.e., the enriched nodes) are depicted as black squares whereas the non-enriched nodes are depicted as white squares.

The functions $e(\mathbf{x},t)$ and $h(\mathbf{x},t)$ are the enrichment functions for the weak and strong discontinuities, respectively. We use the following functions [77, 86]:

$$e(\mathbf{x},t) = |f(\mathbf{x},t)| = h(\mathbf{x},t)\, f(\mathbf{x},t), \tag{31}$$

$$h(\mathbf{x},t) = \mathrm{sign}\{f(\mathbf{x},t)\}, \tag{32}$$

where $f(\mathbf{x},t)$ is the level-set function defined as the signed distance from Γ_i computed with the finite element approximation [112] as follows:

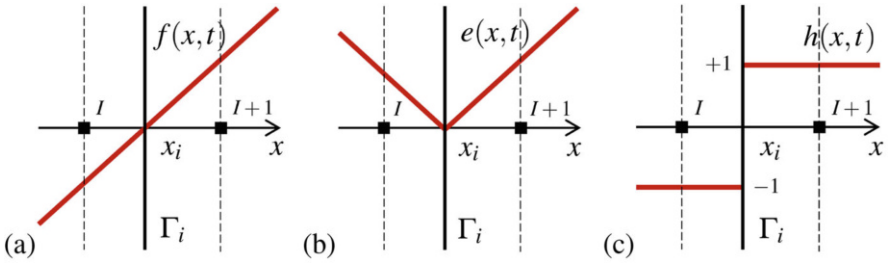


Fig. 3 (**a**) Level-set function defining signed distance from interface; (**b**) edge and (**c**) Heaviside enrichment functions for weak and strong discontinuities, respectively, with normal vector of interface pointing to the right (filled square: enriched fluid nodes; center line: interface; bold line: function)

$$f(\mathbf{x}, t) = \sum_{I \in Q_f} N_I F_I, \tag{33}$$

where F_I is the nodal level set of fluid node I given by

$$F_I = \min_{\mathbf{x} \in \Gamma_i} |\mathbf{x}_I - \mathbf{x}| \cdot \text{sign}\left\{(\mathbf{x}_I - \mathbf{x}) \cdot \mathbf{n}_i(\mathbf{x}, t)\right\}, \tag{34}$$

where $\mathbf{x}$ and $\mathbf{x}_I$ are the spatial coordinates of Γ_i and node I, respectively. The level-set and enrichment functions are shown in Fig. 3 with the normal vector $\mathbf{n}_i$ pointing to the right. We call $e(\mathbf{x}, t)$ and $h(\mathbf{x}, t)$ the edge and Heaviside functions, respectively, to distinguish between their different purposes.

Legay and Kölke [77] first proposed a set of the edge and Heaviside functions, using it with Q1Q1 function spaces to enrich Q2Q1 velocity and pressure function spaces that satisfied the well-known inf-sup condition. By comparison, our approach proposed in [79] and described herein enriches Q1Q1 function spaces with the same Q1Q1 spaces. Therefore, we employ the SUPG (streamline upwind/Petrov–Galerkin), PSPG (pressure stabilizing/Petrov–Galerkin), and (if necessary) DCDD (discontinuity-capturing directional dissipation) stabilization techniques proposed by Brooks, Hughes, Tezduyar, and others [97–102, 105, 107, 109], which were extended to the VMS (variational multiscale) method [103, 104, 106, 108], as an enhanced Petrov–Galerkin formulation of Eq. (8) to stabilize flows with sharp interfaces and the equal-order function spaces. The weak form of the no-slip FTSI problem is given by Eqs. (20) and (21).

(ii) Enrichments for Slip Interface For interfaces with slip, we can apply the Heaviside function $h(\mathbf{x}, t)$ to the velocity enrichment, giving

$$\mathbf{v}(\mathbf{x}, t) = \sum_{I \in Q_f} N_I \mathbf{V}_I + h(\mathbf{x}, t) \sum_{I \in Q_{fi}} N_I \tilde{\mathbf{V}}_I, \tag{35}$$

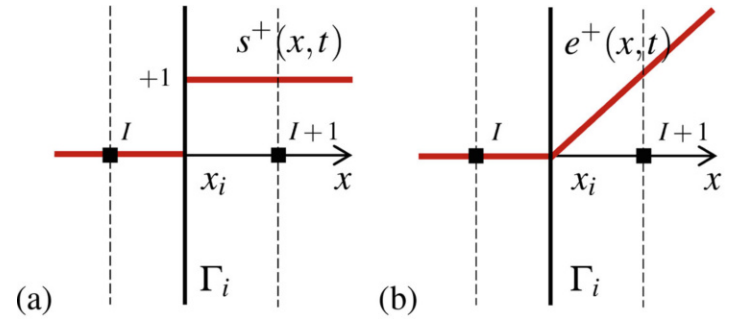


Fig. 4 (**a**) Step and (**b**) edge functions for positive domain of level-set function (filled square: fluid nodes; center line: interface; bold line: function)

on which Eq. (24) imposes the frictionless slip condition while Eq. (25) models slip with friction. Note that Eq. (21) imposes the no-slip condition. If the edge function $e(\mathbf{x}, t)$ is used on velocity instead of the Heaviside function, the slip becomes equal on either side. In this way, enrichments provide a way to model various types of interface in combination with the weak imposition of interfacial continuities.

3.2 Fluid–Solid Interactions

Gerstenberger and Wall [81] proposed XFEM for fluid–solid interactions. We introduce their method herein as a precursor to Sect. 3.3.

(i) Step Function for a Thick Solid For fluid–solid interactions in which the solid occupies a finite volume in the fluid mesh, the step function $s(\mathbf{x}, t)$ shown in Fig. 4a is introduced to define the physical fluid domain in the mesh. The function spaces are given by multiplying the original function spaces and the step function as follows:

$$\mathbf{v}(\mathbf{x}, t) = s(\mathbf{x}, t) \sum_{I \in Q_f} N_I \mathbf{V}_I, \tag{36}$$

$$p(\mathbf{x}, t) = s(\mathbf{x}, t) \sum_{I \in Q_f} N_I P_I, \tag{37}$$

where $s(\mathbf{x}, t)$ cuts off non-physical spaces from the originals and reproduces interface-fitted ones. Figure 5 shows the no-slip case. Equation (36) can also be written as the following equation (pressure is omitted) considering that Q_f includes Q_{fi} in the present notation:

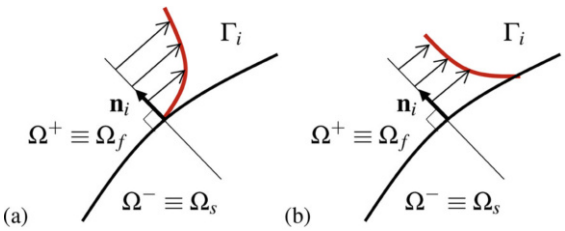


Fig. 5 Illustration of spatial profiles of (**a**) velocity and (**b**) pressure at and near a no-slip interface between a fluid and a solid structure

$$\mathbf{v}(\mathbf{x},t) = s(\mathbf{x},t) \sum_{I \in \mathcal{Q}_f \setminus \mathcal{Q}_{fi}} N_I \mathbf{V}_I + s(\mathbf{x},t) \sum_{I \in \mathcal{Q}_{fi}} N_I \mathbf{V}_I. \tag{38}$$

The step function $s(\mathbf{x},t)$ is given by the level-set function $f(\mathbf{x},t)$. The weak form of FSI is given by Eq. (20).

(ii) Multiple Step Functions for a Thin Solid For fluid–solid interactions in which the solid is thick but the thickness does not exceed the width of a fluid element, Gerstenberger and Wall [81] proposed the following multiple function spaces:

$$\mathbf{v}(\mathbf{x},t) = \sum_{I \in \mathcal{Q}_f \setminus \mathcal{Q}_{fi}} N_I \mathbf{V}_I + s^+(\mathbf{x},t) \sum_{I \in \mathcal{Q}_{fi}} N_I \mathbf{V}_I^+ + s^-(\mathbf{x},t) \sum_{I \in \mathcal{Q}_{fi}} N_I \mathbf{V}_I^-, \tag{39}$$

where $s^+(\mathbf{x},t)$ and $s^-(\mathbf{x},t)$ are step functions that define the fluid domain on either side of the thin solid. The function spaces can be written in a simple alternative form corresponding to Eq. (36):

$$\mathbf{v}(\mathbf{x},t) = s^+(\mathbf{x},t) \sum_{I \in \mathcal{Q}_f} N_I \mathbf{V}_I^+ + s^-(\mathbf{x},t) \sum_{I \in \mathcal{Q}_f} N_I \mathbf{V}_I^-. \tag{40}$$

The pressure can be written in the same form. The weak form of an FSI problem has two constraints for the two interfaces and is written as

$$\delta W_f + \delta W_s + \delta W_i^+ + \delta W_i^- = 0, \tag{41}$$

where δW_i^+ and δW_i^- are given by Eq. (21) when the no-slip condition applies.

3.3 Flow Around Fixed Objects

To compute flow around fixed objects, Eqs. (36) and (37) provide suitable function spaces to find a solution of Eq. (26). In contrast, XFEM also provides an approach [75, 89] that does not require the weak imposition of the no-slip condition given by $\dot{\mathbf{u}}(\mathbf{X},t) = \mathbf{0}$ at Γ_i. The latter technique is given by shift-type enrichment:

$$\mathbf{v}(\mathbf{x},t) = s(\mathbf{x},t) \sum_{I \in Q_f \setminus Q_{fi}} N_I \mathbf{V}_I + e^+(\mathbf{x},t) \sum_{I \in Q_{fi}} N_I \mathbf{V}_I, \tag{42}$$

where the enrichment function for the interfacial fluid nodes Q_{fi} is shifted from the step function $s(\mathbf{x},t)$ to the one-sided edge function $e^+(\mathbf{x},t)$ shown in Fig. 4b, compared with Eq. (38). Because the edge function is naturally zero at Γ_i, Eq. (42) meets and reproduces the no-slip condition as shown in Fig. 5a. The pressure is given by Eq. (37). The weak form to be solved is only $\delta W_f = 0$ of Eq. (7), which requires neither LMs nor penalties. Although the degeneration is valid for flows around fixed bodies or along fixed boundaries, and blending elements/functions occur, there are cases where this approach is useful. In practice, we apply the technique based on Eq. (42) to compute flows in microstructures [75] with extension to fluid and solid phases on a voxel mesh, and to develop a user-friendly tool for computational fluid dynamics [89]. Applications are shown in Sect. 9.

3.4 Remarks on Enrichment

We make some remarks on enrichment and discuss some open issues and practical solutions.

(i) Spatial Gradient Because enrichment functions tend to have spatial profiles, the gradients are taken into account for the spatial gradients of unknown functions through a chain rule as follows. Let $\mathbf{a}$ represent an unknown function such as a velocity or pressure, with $\mathbf{A}_I$ and $\tilde{\mathbf{A}}_J$ the usual and new nodal unknowns for $\mathbf{a}$, respectively. Let $\phi(\mathbf{x},t)$ represent the enrichment function. A general form of the finite element approximation of $\mathbf{a}$ with enrichments can be written as

$$\mathbf{a}(\mathbf{x},t) = \sum_{I \in Q_f} N_I(\mathbf{x}) \mathbf{A}_I(t) + \sum_{J \in Q_{fi}} \phi(\mathbf{x},t) M_J(\mathbf{x}) \tilde{\mathbf{A}}_J(t), \tag{43}$$

where we have supposed cases for which $N_I = M_J$ and $N_I \neq M_J$. Thus, the spatial gradient of $\mathbf{a}$ with respect to $\mathbf{x}$ is given by

$$\nabla_x \mathbf{a} = \sum_{I \in Q_f} (\nabla_x N_I) \mathbf{A}_I + \sum_{J \in Q_{fi}} \{(\nabla_x \phi) M_J + \phi (\nabla_x M_J)\} \tilde{\mathbf{A}}_J. \tag{44}$$

For functions such as the Heaviside function $h(\mathbf{x},t)$, $\nabla_x \phi$ is zero and the corresponding term of Eq. (44) vanishes. For functions such as the edge function $e(\mathbf{x},t)$, $\nabla_x \phi$ has a finite value that we compute using Eq. (33) and the definition of $\phi(\mathbf{x},t)$. For example, the gradient of $e(\mathbf{x},t)$ is given by

$$\nabla_x e(\mathbf{x},t) = h(\mathbf{x},t) \nabla_x f(\mathbf{x},t) = h(\mathbf{x},t) \sum_{I \in Q_f} (\nabla_x N_I) F_I. \tag{45}$$

(ii) Time Derivative For FSI and flow with moving boundaries, the enrichment function varies also in time at the spatial point **x**. In theory, this time dependence affects the time derivatives of **a** as follows:

$$\left.\frac{\partial \mathbf{a}}{\partial t}\right|_x = \sum_{I \in Q_f} N_I \left.\frac{\partial \mathbf{A}_I}{\partial t}\right|_x + \sum_{J \in Q_{fi}} M_J \left(\left.\frac{\partial \phi}{\partial t}\right|_x \tilde{\mathbf{A}}_J + \phi \left.\frac{\partial \tilde{\mathbf{A}}_J}{\partial t}\right|_x \right), \tag{46}$$

where the second and third terms represent the time dependence. How to compute the second and third terms when we apply XFEM to dynamical interfacial problems remains an open issue. For example, Legay and Zilian proposed an enriched space-time (EST) method for computing the time dependence in 2D applications [77, 78, 80]. Our approach employs a finite difference method in time, in which we compute $\partial\phi/\partial t|_x$ explicitly at every iteration step in an implicit time integration; this is the same as computing a convective velocity using the Navier–Stokes equations. We use the finite difference scheme to compute $\left.\partial\tilde{\mathbf{A}}_J/\partial t\right|_x$ implicitly, as well as other time-dependent quantities.

(iii) Integration of Fluid Elements Crossed by Interface Discontinuities at the interface require practicable measures [130, 131] for numerical integration of the fluid elements crossed by the interface. This point is a long-standing issue in XFEM. One standard solution is to integrate each fluid element using locally interfacially fitted triangular/tetrahedral cells as illustrated in Fig. 6a [76, 80, 81, 83, 110–112]. This approach is reasonable in relation to consistency and practical accuracy. However, it is often regarded as a drawback to XFEM because one of the expectations of XFEM is that it is mesh-free. For FSI and flow with moving boundaries, this approach requires triangulation/tetrahedralization at every iteration step in the nonlinear time-dependent computations because the interfaces are moving. Considering these drawbacks, we apply many-point Gaussian quadrature to the integration of enriched elements as shown in Fig. 6b; the numerical accuracy are reported in [86], which we summarize in Sect. 7. By this way, we remove such interface-fitting background procedures from the computation of FSI based on XFEM, prioritizing instead its enhanced applicability to geometrically complex problems in combination with the LLM technique described later. If one is faced with unacceptable numerical accuracy, triangulation or adaptive h-refinement quadrature (see [19, 132]) is recommended as another solution.

4 Lagrange Multiplier Technique for FSI

This section describes the LLM technique for finite element computation of Eq. (21) and its variants. The LLM technique was proposed in [79] and removes the need for any interface-fitted background procedures in the present XFEM in combination with quadrature of the LMs and the fluid elements crossed by Γ_i. The numerical aspect and performance of the LLM technique are reported in [86, 88].

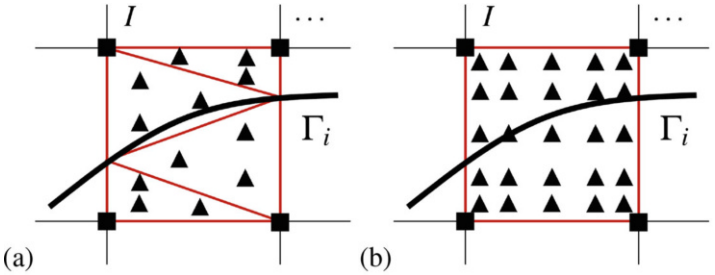


Fig. 6 Domain integration of an fluid element crossed by an interface: (**a**) triangulation to interfacially fitted sub-cells; (**b**) many-point Gaussian quadrature (filled square: fluid nodes; triangle: integration points; thick line: interface)

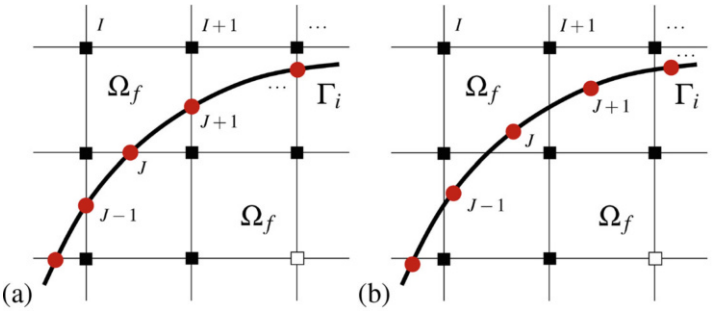


Fig. 7 Finite element discretization of Lagrange multiplier (LM) space: (**a**) intersection approach; (**b**) Lagrangian mesh approach (filled square, open square fluid nodes; filled circle: LM nodes)

(i) Intersections and Lagrangian Nodes Let us suppose an FTSI in which the discontinuities at Γ_i are enriched by a set of edge and Heaviside functions. Figure 7 illustrates the LLM approach alongside a major approach in this area, namely the intersection approach [76, 80, 81, 83]. The intersection approach discretizes the LM function space at the intersections of Γ_i and the edges of the fluid elements and triangulates them to complete the LM function spaces. An essential advantage is that the FSI conditions can be imposed reliably, whereas a drawback is the geometrical gap between the intersection line and the Lagrangian mesh line, which limits the convergence. In contrast to the intersection approach, the LLM technique discretizes the LMs at the Lagrangian nodes of the structural mesh on the interface Γ_i, and the shape functions for displacement are also used for the LMs. In other words, the LLM technique uses the Lagrangian mesh not only for structures but also for the LMs to discretize Eq. (21) without triangulation. Compared with the intersection-based method, the LLM approach involves no geometrical gaps between the Lagrangian, Eulerian, and multiplier meshes.

(ii) Function Spaces for Constraints To describe the LLM technique, let Q_s represent the Lagrangian nodes on Γ_i with L_J the shape functions on Γ_i. For FTSI, Γ_i is the neutral plane of a structural element and L_J are the shape functions defining the plane. The finite element function spaces for the LM and its variational are written, respectively, as

$$\boldsymbol{\lambda}\left(\mathbf{X}_i, t\right) = \sum_{J \in Q_s} L_J\left(\mathbf{x}_i\right) \boldsymbol{\Lambda}_J \quad \text{at } \Gamma_i, \tag{47}$$

$$\delta\boldsymbol{\lambda}\left(\mathbf{X}_i, t\right) = \sum_{J \in Q_s} L_J\left(\mathbf{x}_i\right) \delta\boldsymbol{\Lambda}_J \quad \text{at } \Gamma_i, \tag{48}$$

where $\mathbf{X}_i$ represents the Lagrangian coordinates of Γ_i tracking the present position, $\mathbf{x}_i = \mathbf{x}(\mathbf{X}_i, t)$ and denotes that $\boldsymbol{\lambda}$ is the Lagrangian variable in the technique, and $\boldsymbol{\Lambda}_J$ and $\delta\boldsymbol{\Lambda}_J$ are the nodal unknowns and variational, respectively. $\boldsymbol{\Lambda}_J$ is the new unknown for node J. The function spaces for the structural displacement rate and its variational are the same as those for the LM and are written, respectively, as

$$\dot{\mathbf{u}}\left(\mathbf{X}_i, t\right) = \sum_{J \in Q_s} L_J\left(\mathbf{x}_i\right) \dot{\mathbf{U}}_J \quad \text{at } \Gamma_i, \tag{49}$$

$$\delta\dot{\mathbf{u}}\left(\mathbf{X}_i, t\right) = \sum_{J \in Q_s} L_J\left(\mathbf{x}_i\right) \delta\dot{\mathbf{U}}_J \quad \text{at } \Gamma_i, \tag{50}$$

where $\dot{\mathbf{U}}_J$ are the nodal displacement rates and thus $\mathbf{U}_J$ are the nodal displacements. The function space for the velocity of the fluid at Γ_i is given through interpolation from the fluid elements crossed by Γ_i and is written as

$$\mathbf{v}\left(\mathbf{x}_i, t\right) = \sum_{I \in Q_{fi}} N_I\left(\mathbf{x}_i\right) \mathbf{V}_I \quad \text{at } \Gamma_i, \tag{51}$$

$$\delta\mathbf{v}\left(\mathbf{x}_i, t\right) = \sum_{I \in Q_{fi}} N_I\left(\mathbf{x}_i\right) \delta\mathbf{V}_I \quad \text{at } \Gamma_i, \tag{52}$$

from which the enrichment term of Eq. (29) ensuring weak discontinuity at Γ_i is absent because in theory the edge function $e(\mathbf{x}, t)$ becomes zero at Γ_i. This property, namely $e(\mathbf{x}_i, t) = 0$, enables us to discretize Eq. (21) using only the usual parts of the FEM.[2]

[2] XFEM can accept various types of enrichment function $\phi(\mathbf{x}, t)$. For example, as well as the edge function $e(\mathbf{x}, t)$, the ramp function $r(\mathbf{x}, t)$ given by

$$r\left(\mathbf{x}, t\right) = \sum_{I \in Q_f} N_I \left|F_I\right| - e\left(\mathbf{x}, t\right) \tag{53}$$

can reproduce a weak discontinuity and is generally considered superior to the edge function in doing so because it can reproduce the discontinuity within crossed elements without the need for partially enriched surrounding elements called blending elements in XFEM. However, if we adopt the ramp function, the interfacial velocity has time-dependent enrichment terms as follows:

$$\mathbf{v}\left(\mathbf{x}_i, t\right) = \sum_{I \in Q_{fi}} N_I \mathbf{V}_I + r\left(\mathbf{x}_i, t\right) \sum_{I \in Q_{fi}} N_I \tilde{\mathbf{V}}_I, \quad \text{with} \quad r\left(\mathbf{x}_i, t\right) = \sum_{I \in Q_{fi}} N_I \left|F_I\right|. \tag{54}$$

(iii) Integration for Constraint Equation The LLM technique applies Gaussian quadrature to Eq. (21) on the Lagrangian mesh, taking care of any over- or under-constraining fluid function spaces [118, 133, 134]. The practical strategy that we use currently is as follows:

1. Over-constraints occur when a Lagrangian structural mesh is finer than the fluid mesh. This is because the smoothness of the displacement rate of the structure cannot be reproduced by the function space of the flow. Therefore, we apply the LLM technique in well-conditioned situations in which (i) the resolutions of the structural and fluid meshes are approximately equal around Γ_i and (ii) the fluid mesh is finer than the structural mesh around Γ_i.
2. However, the preferred setting can give rise to under- or no-constraints if there are too few quadrature points. The LLM technique increases the number of quadrature points adaptively while considering locally the difference in resolution between the fluid and structural meshes. In most cases, we use six to ten points for each natural coordinate of a structural element. This adaptive increase (or decrease) enhances the solution.
3. Although potential relaxation techniques already exist, basically we do not introduce them because the restriction is reasonable considering the boundary-layer resolution.

The LLM technique discretizes the first term of Eq. (21) as

$$\int_{\Gamma_i} \delta\boldsymbol{\lambda} \cdot \mathbf{v} d\Gamma = \sum_{J \in Q_s} \sum_{I \in Q_{fi}} \delta\boldsymbol{\Lambda}_J \cdot \mathbf{V}_I \int_{\Gamma_i} L_J\left(\mathbf{x}_i^*\right) N_I\left(\mathbf{x}_i^*\right) d\Gamma, \tag{55}$$

where $\mathbf{x}_i^*$ denotes that the number of quadrature points of the structural element has increased. At each $\mathbf{x}_i^*$, a fluid element is sought that covers $\mathbf{x}_i^*$, and the set of interpolation functions $N_I(\mathbf{x}_i^*)$ of the fluid velocity are computed by an iteration scheme. After computation of a Lagrangian element, the LLM technique checks whether there are unconstrained fluid elements crossed by the Lagrangian element. If there are, the Lagrangian element is recomputed with adaptive increase of the number of integration points. The third term of Eq. (21) is given by the transpose of Eq. (55). The discretized form is given by

$$\int_{\Gamma_i} \delta\mathbf{v} \cdot \boldsymbol{\lambda} d\Gamma = \sum_{I \in Q_{fi}} \sum_{J \in Q_s} \delta\mathbf{V}_I \cdot \boldsymbol{\Lambda}_J \int_{\Gamma_i} N_I\left(\mathbf{x}_i^*\right) L_J\left(\mathbf{x}_i^*\right) d\Gamma. \tag{56}$$

For FSI and flow with moving boundaries, enrichment functions that meet neither $\partial\phi(\mathbf{x}_i, t)/\partial t = 0$ nor $\phi(\mathbf{x}_i, t) = 0$ seem to cause temporal instability even if the time dependence is accounted for in the computation. We therefore select the edge function for the enrichment.

For Eq. (55), the set of shape functions $N_I(\mathbf{x}_i^*)$ is the interpolation of $\mathbf{v}(\mathbf{x}_i, t)$ from the Eulerian nodes to the Lagrangian point. For Eq. (56), those are the distribution of $\boldsymbol{\lambda}(\mathbf{X}_i, t)$ from the Lagrangian point to the Eulerian nodes.

Meanwhile, the LLM technique completes the second term of Eq. (21) with the Lagrangian mesh only. The discretized form is given by

$$\int_{\Gamma_i} \delta\boldsymbol{\lambda} \cdot \dot{\mathbf{u}} d\Gamma = \sum_{J \in Q_s} \sum_{K \in Q_s} \delta\boldsymbol{\Lambda}_J \cdot \dot{\mathbf{U}}_K \int_{\Gamma_i} L_J(\mathbf{x}_i) L_K(\mathbf{x}_i) d\Gamma \tag{57}$$

and is computed on the Lagrangian mesh with normal quadrature points. Its transpose gives the final term of Eq. (21), which is written as

$$\int_{\Gamma_i} \delta\dot{\mathbf{u}} \cdot \boldsymbol{\lambda} d\Gamma = \sum_{K \in Q_s} \sum_{J \in Q_s} \delta\dot{\mathbf{U}}_K \cdot \boldsymbol{\Lambda}_J \int_{\Gamma_i} L_K(\mathbf{x}_i) L_J(\mathbf{x}_i) d\Gamma. \tag{58}$$

5 Spatial Discretizations of FTSI Equation

In previous sections, we have described FSI techniques for FTSI and fluid–solid interactions, focusing on the interface. In practice, most of our applications are of FTSI, in which the structure is modeled by structural elements such as shells, beams, and membranes. The spatial discretization of Eq. (20) for FTSI problems is summarized as follows [88]:

1. For deformation of thin structures, the Cauchy equilibrium equation described in the total Lagrangian weak form δW_s of Eq. (16) is employed. The δW_s is discretized by the MITC (mixed interpolation of tensorial components) Q1 shell elements proposed by Dvorkin, Bathe, and others [135–138]. The shell is described using Reissner–Mindlin plate theory and has locking-free shear-strain function spaces and finite rotation increments. The "Q1" denotes that the neutral plane is defined by bi-linear shape functions.
2. For fluid flow, the Navier–Stokes equations given by δW_f of Eq. (8) are employed. The δW_f is Eulerian and is discretized by mixed Q1Q1 elements with the SUPG, PSPG, and DCDD stabilization techniques and formulations proposed by Brooks, Hughes, Tezduyar, and others [97, 100, 101, 105, 107, 109].
3. At the interface, we assume the no-slip condition. The Q1Q1 elements crossed by the interface are enriched by the set of edge and Heaviside functions proposed by Legay and Kölke [77] to reproduce weak and strong discontinuities. These enrichments are written as Eqs. (29) and (30).

4. The no-slip condition is imposed by the weak form of δW_i in Eq. (21). The δW_i is discretized by the LLM technique, which embeds Lagrangian interfaces in an Eulerian mesh that thereby captures the interfaces.

For fluid–solid interactions, solid elements and a different set of enrichments such as Eqs. (36) and (37) can be employed.

6 Time Integration

After completing the spatial discretization of Eq. (20), simultaneous nonlinear finite element matrix equations are obtained as an assemblage and are solved as a monolithic equation system in this method [88]. The Newmark-β method [119] is employed as an implicit time integration scheme of the equation, in which the Newton–Raphson method is employed as a tangential iteration scheme for solving the nonlinear equation system.

(i) Matrix Equation The fully assembled matrix equation that we solve at every nonlinear iteration to find the increments is written as

$$\begin{bmatrix} \mathbf{A}^*_{ff} & \mathbf{O} & \mathbf{B}^T_{if} \\ \mathbf{O} & \mathbf{A}^*_{ss} & -\mathbf{B}^T_{is} \\ \mathbf{B}_{if} & -\mathbf{B}_{is} & \mathbf{O} \end{bmatrix} \cdot \Delta \begin{Bmatrix} \boldsymbol{\Phi}_f \\ \dot{\boldsymbol{\Phi}}_s \\ \boldsymbol{\Phi}_i \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_f \\ \mathbf{R}_s \\ \mathbf{R}_i \end{Bmatrix}, \tag{59}$$

where $\mathbf{A}^*_{ff}$ is the fluid matrix composed of all terms with the secant form of the convection term, $\mathbf{A}^*_{ss}$ is the structure matrix composed of mass and tangent stiffness, $\mathbf{B}_{if}$ and $\mathbf{B}_{is}$ are the LM matrices given by Eqs. (55) and (56), respectively, and $\mathbf{R}_f$, $\mathbf{R}_s$, and $\mathbf{R}_i$ are the residual vectors given by Eqs. (8), (16), and (21). The unknown vectors $\boldsymbol{\Phi}_f$, $\dot{\boldsymbol{\Phi}}_s$, and $\boldsymbol{\Phi}_i$ with increment notation Δ are defined, respectively, as

$$\boldsymbol{\Phi}_f = \left\{ \cdots \mathbf{V}_I \ \tilde{\mathbf{V}}_I \ P_I \ \tilde{P}_I \ \cdots \right\}^T, \tag{60}$$

$$\dot{\boldsymbol{\Phi}}_s = \left\{ \cdots \dot{\mathbf{U}}_J \ \dot{\boldsymbol{\Theta}}_J \ \cdots \right\}^T, \tag{61}$$

$$\boldsymbol{\Phi}_i = \left\{ \cdots \boldsymbol{\Lambda}_J \ \cdots \right\}^T, \tag{62}$$

with $\tilde{\mathbf{V}}_I$ and $\tilde{P}_I$ set to zero for the non-enriched fluid nodes and where $\dot{\boldsymbol{\Theta}}_J$ is the nodal finite rotation rate of the shell element [135–138]. Equation (59) is solved by the generalized minimal residual (GMRES) iterative solver [139] with the incomplete LU(1) preconditioner, where the numeral in round brackets denotes the fill-in level of the LU-decomposition-based preconditioner.

(ii) Time Integration Algorithm A typical computational algorithm uses the following scheme:

1. Generate Eulerian mesh for fluid domain
2. Generate Lagrangian meshes for thin structures
3. Start and advance the implicit time integration: $t = t + \Delta t$
   (a) Search fluid nodes around structures to reduce XFEM computations below
   (b) Compute level-set function of fluid nodes: $f(\mathbf{x}, t)$
   (c) Set up IRC fluid nodes: Q_{fi}
   (d) Compute finite element matrices and vectors of Eq. (59)
   (e) Solve Eq. (59) as a monolithic equation by ILU(1)-GMRES
   (f) Update unknown variables $\boldsymbol{\Phi}$ and position of Lagrangian meshes
   (g) Judge convergence. If not converged, go to (a).
4. Judge integrated time. If not finished, go to 3.

This algorithm does not differ much from the usual implicit scheme except for 3(a), 3(b), and 3(c), which are pre-conditions of XFEM, because no triangulation/tetrahedralization procedures are used. Another merit is that the size and structure of the LM array $\boldsymbol{\Phi}_i$ do not change throughout the simulation despite the fact that this array changes with every update of the interface position for the intersection-based LM method. These features simplify the computation program and scheme and also seem to have a stabilizing effect on the time integration of the LM-based coupling.

7 Numerical Tests for Flows with a Domain-Partitioning Interface

Before applying XFEM to FSI problems, the validity of the XFEM for FSI interfaces is often checked using domain-partitioning interface (DPI) problems. The DPI problems check whether a discontinuity enriched by XFEM can partition a flow domain into two domains at the interface. Herein, we discuss a DPI problem that we introduced in [86, 88] and show excerpts of the numerical results. The results indicate which terms are important when we compute FSI problems using IC methods.

Flow with a Cylindrical DPI The DPI problem in [86, 88] is an extension of the flow around a cylinder. Figure 8a shows the setup, in which we consider a cylinder in a square channel as a thin wall Γ_i partitioning the flow channel into inner and outer domains of the cylinder. The diameter of the cylinder is half the length of the channel. The left and right sides of the channel are the inflow and traction-free outflow boundaries, respectively. The upper and lower sides are slip boundaries. At the cylindrical Γ_i, the no-slip condition given by $\dot{\mathbf{u}} = \mathbf{0}$ is imposed weakly by the LLM technique, and the flow functions across Γ_i are enriched by the set of edge and Heaviside functions. A weak form of the DPI problem is given by Eqs. (26) and (27) with $\dot{\mathbf{u}}(\mathbf{X}_i, t) = \mathbf{0}$. The Reynolds number is given by $\mathrm{Re} = \rho_f V D/\mu_f$, where D is the diameter of the cylinder and V is the inflow speed at the left side of the channel.

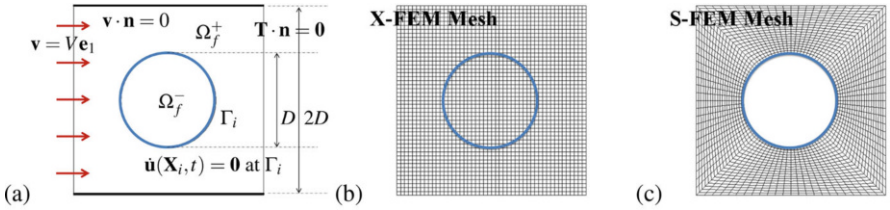


Fig. 8 Flow with a domain-partitioning cylinder and two types of fluid mesh: (**a**) problem setup; (**b**) non-interface-fitted mesh; (**c**) interface-fitted mesh

The fluid mesh is regular and an example is shown in Fig. 8b. The boundary Γ_i is also divided uniformly into Q1 LM elements. The divisions are determined by the element-length equations:

$$l_s = \epsilon l_f \quad \text{with} \quad l_s = \pi D/n_s, \quad l_f = 2\sqrt{2}D/n_f, \tag{63}$$

where l_s and l_f are the element lengths of the multiplier and the fluid mesh, respectively, and n_s and n_f are the respective divisions. The parameter ϵ is the ratio of the resolution of the fluid mesh to that of the multiplier mesh; $\epsilon > 1$ indicates that the fluid mesh is finer than the multiplier mesh. Normally, we have $\epsilon = 1$, which corresponds to having one LM per fluid element in general. The interface-fitted mesh of Fig. 8c, which has nearly the same element length as that of the non-interface-fitted mesh of Fig. 8b on average, gives a reference result.

7.1 Tests on Quadrature of Fully Enriched Fluid Elements

In [86], we conducted detailed numerical tests on Gaussian quadratures of the fully enriched fluid elements crossed by the interface, in which we varied the number of quadrature points, the mesh resolution, and the Reynolds number. Figures 9 and 10 extract the numerical results for

$$\mathrm{Re} = 10, \quad n_f = 100, \quad \epsilon = 1, \quad q_f = 2,\ 3,\ 4, \quad q_i = 10,$$

and

$$\mathrm{Re} = 1000, \quad n_f = 100, \quad \epsilon = 1, \quad q_f = 2,\ 4,\ 6, \quad q_i = 10,$$

respectively, where q_f denotes the number of quadrature points of the enriched elements per local coordinate and q_i denotes the corresponding number for the LM elements. The corresponding number for non-enriched fluid elements are two points. Each figure shows computed nodal velocities and LMs after normalization by the inflow velocity V and the reference pressure $1/2\rho_f V^2$, respectively. Figures 9d and 10d show the reference results obtained with the interface-fitted mesh.

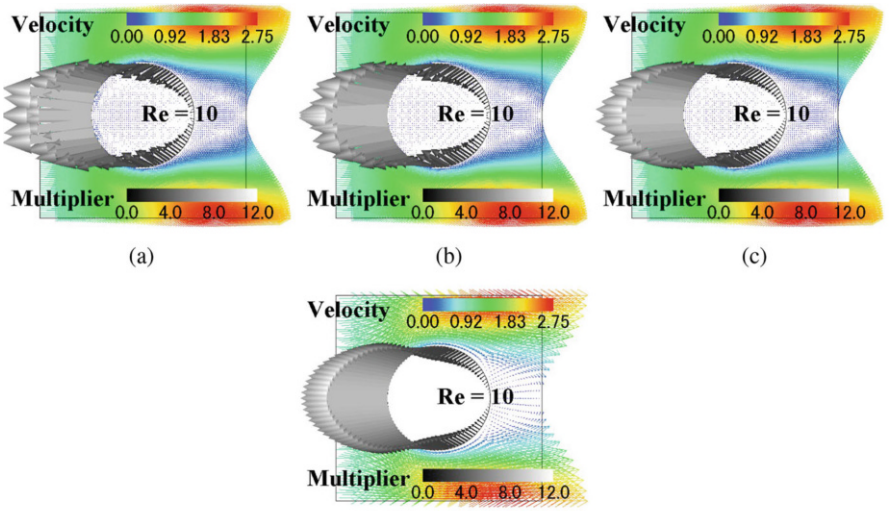


Fig. 9 Quadrature of enriched fluid elements: Re $= 10$ with $n_f = 100$, $\epsilon = 1$, and $q_i = 10$. (**a**) $q_f = 2$. (**b**) $q_f = 3$. (**c**) $q_f = 4$. (**d**) Reference

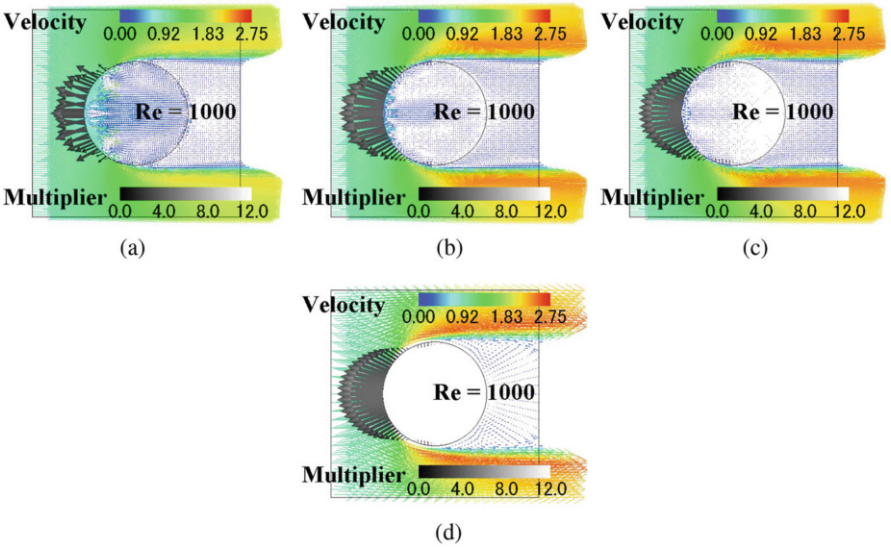


Fig. 10 Quadrature of enriched fluid elements: Re $= 1000$ with $n_f = 100$, $\epsilon = 1$, and $q_i = 10$. (**a**) $q_f = 2$. (**b**) $q_f = 4$. (**c**) $q_f = 6$. (**d**) Reference

These results tell us the following:

1. The numerical solutions are enhanced by increasing the number q_f of quadrature points. They show convergence to the reference solution given by the interface-fitted mesh.
2. For low Reynolds number (i.e., approaching Stokes flow), Gaussian quadrature gives an acceptable result even if the number of quadrature points is small. As the Reynolds number is increased, the integration accuracy gradually decreases. However, Fig. 10 shows that the accuracy can be enhanced by applying more quadrature points.
3. When the number of quadrature points is insufficient, the enrichments lose smoothness along the interface and finally lose the discontinuity. The loss is often observed as penetration/leakage and results in oscillation and underestimation of LMs as can be seen in Fig. 10a.
4. If the numerical accuracy is considered acceptable, Gaussian quadrature can be used as an approximate numerical integration scheme; otherwise, other approaches such as interface-fitting triangulation should be considered.

For applications we are concerned with, we deem the quadrature (the number of points of which can control the computational accuracy) to be acceptable. Normally, we apply ten points per local coordinate considering accuracy and various geometries.

7.2 Tests on Quadrature of Lagrange Multipliers

In [86], we also conducted numerical tests on Gaussian quadrature of the LM elements, varying the number of quadrature points, the ratio ϵ of the resolution of the fluid mesh to that of the LM mesh, and the Reynolds number to validate the LLM technique. Figures 11 and 12 extract the numerical results for

$$\mathrm{Re} = 10, \quad n_f = 100, \quad \epsilon = 2, \quad q_f = 10, \quad q_i = 2,\ 3,\ 4,\ 5,$$

and

$$\mathrm{Re} = 1000, \quad n_f = 100, \quad \epsilon = 2, \quad q_f = 10, \quad q_i = 2,\ 3,\ 4,\ 5,$$

respectively, where we set $q_f = 10$ to discuss the integration of LM elements only.[3] All figures are drawn in the same way as were Figs. 9 and 10. The references are given by Figs. 9d and 10d, respectively.

[3]The multiplier mesh for $\epsilon = 2$ is twice as coarse as that for $\epsilon = 1$ if n_f is fixed. Therefore, the multipliers shown in Figs. 11 and 12 are twice as coarse as those of Figs. 9 and 10, respectively.

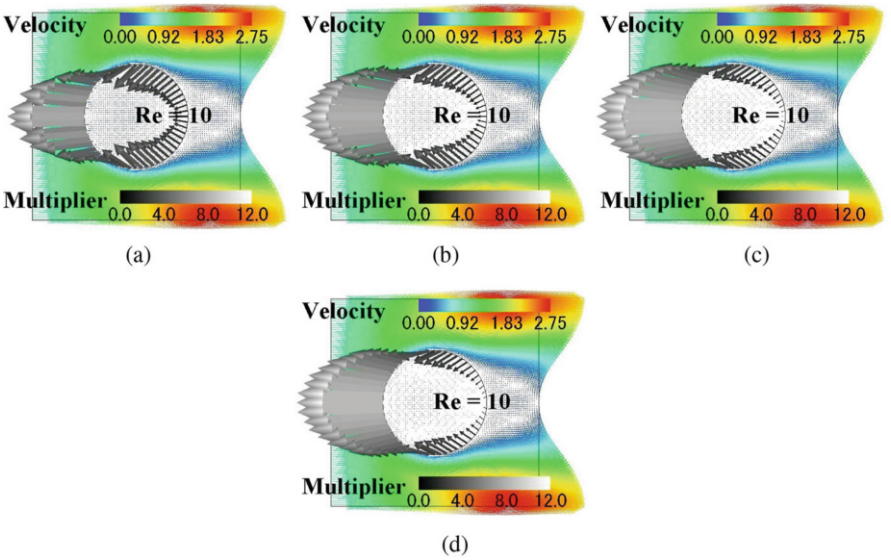


Fig. 11 Quadrature of Lagrangian LMs: Re $= 10$ with $n_f = 100$, $\epsilon = 2$, and $q_f = 10$. (**a**) $q_i = 2$. (**b**) $q_i = 3$. (**c**) $q_i = 4$. (**d**) $q_i = 5$

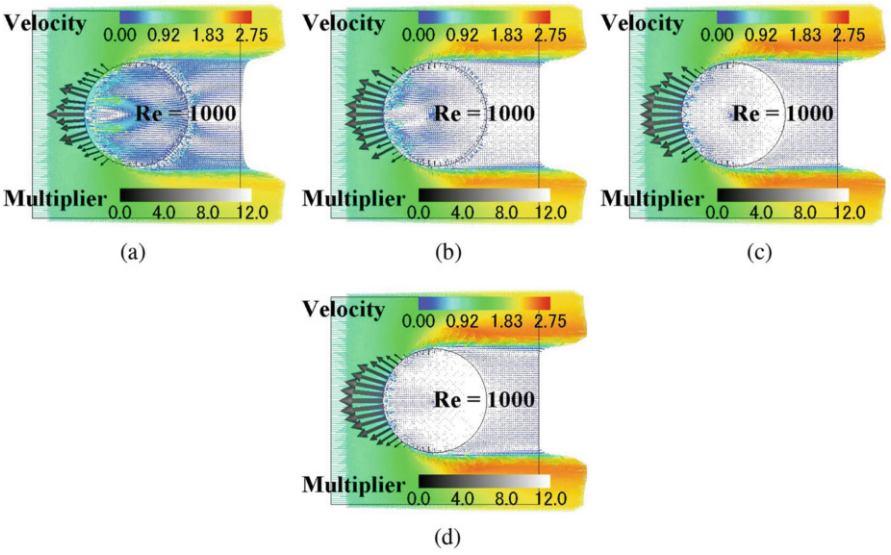


Fig. 12 Quadrature of Lagrangian LMs: Re $= 1000$ with $n_f = 100$, $\epsilon = 2$, and $q_f = 10$. (**a**) $q_i = 2$. (**b**) $q_i = 3$. (**c**) $q_i = 4$. (**d**) $q_i = 5$

The numerical features of the results seem to accord with the quadrature of enriched fluid elements. The strategies used by the LLM technique were summarized in Sect. 4. For the cases of Figs. 11 and 12, the relative resolution of two (i.e., $\epsilon = 2$) corresponds to there being roughly one LM per two or three fluid elements. Therefore, for the LM elements, this case requires at least three quadrature points per local coordinate. As the increase of q_i, imposition of the constraint may become smooth along the interface, which may enhance the results. When there are too few integration points for the relative resolution ϵ, some elements may not be subject to the constraint. Figures 11 and 12 meet the estimation and the inferences. The numerical results recommend that we use more than 2ϵ points for better results. Normally, we use 4ϵ points or more per local coordinate considering the smoothness of the imposition and the safety factor against under-constraint. The computational cost of doing so is nearly negligible compared with that of integrating the enriched fluid elements. We note again that the LLM technique as it stands is unable to handle cases for which $\epsilon < 1$, as described in Sect. 4. Some results for the over-constrained situation are also shown in [86].

7.3 Continuous Versus Discontinuous Interfaces

Using XFEM for FSI interfaces gives a way to identify numerically how the presence of a discontinuity at the interface enhances the computational results whereas its absence degrades them. In [88], we discussed the issue by conducting numerical tests on four types of interface that could be modeled and reproduced by the present XFEM. Herein, we discuss the issue, referring to the results in [88].

The four types of interface are given by turning the edge and Heaviside function enrichments on or off separately. Doing so reproduces interfaces (a) without discontinuity, (b) with weak discontinuity of velocity only, (c) with strong discontinuity of pressure only, and (d) with both discontinuities. The four types and the corresponding results will be denoted as Γ_i^{**}, Γ_i^{E*}, Γ_i^{*H}, and Γ_i^{EH}, respectively, with the two superscripts associated with the enrichments being on or off (i.e., discontinuities). Figure 13 displays each numerical result for

$$\mathrm{Re} = 10, \quad n_f = 48 \quad \epsilon = 1, \quad q_f = q_i = 10,$$

and Fig. 14 does the same for

$$\mathrm{Re} = 1000, \quad n_f = 48 \quad \epsilon = 1, \quad q_f = q_i = 10.$$

The references are given by Figs. 9d and 10d ($n_f = 100$), respectively.

We begin by confirming that Γ_i^{EH} provides the most comparable results to the corresponding references among the four interfaces. The results for Γ_i^{*H} seem also to be comparable, but the multipliers for $\mathrm{Re} = 10$ have slight hour-glass mode oscillations and a twin vortex is stimulated for $\mathrm{Re} = 1000$ in the inner domain of

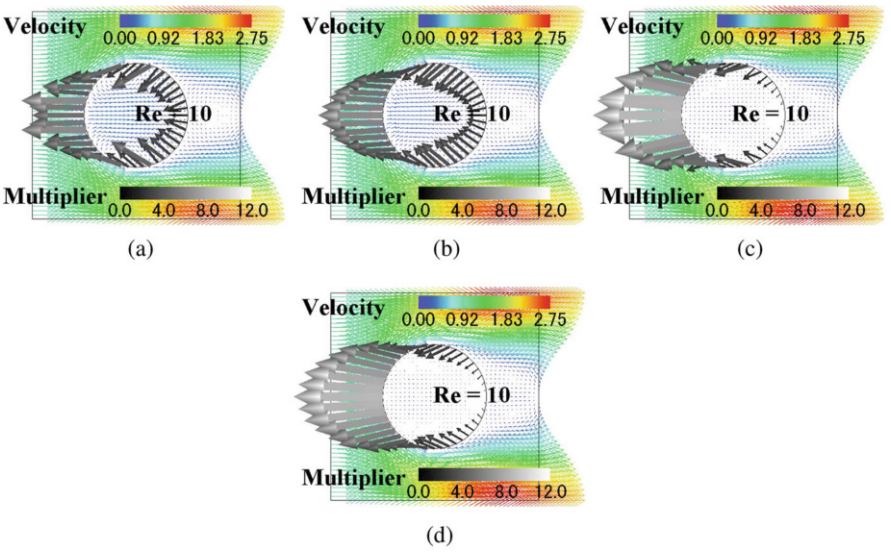


Fig. 13 Continuous versus discontinuous interfaces: Re $= 10$ with $n_f = 48$ and $\epsilon = 1$. (**a**) For Γ_i^{**}. (**b**) For Γ_i^{E*}. (**c**) For Γ_i^{*H}. (**d**) For Γ_i^{EH}

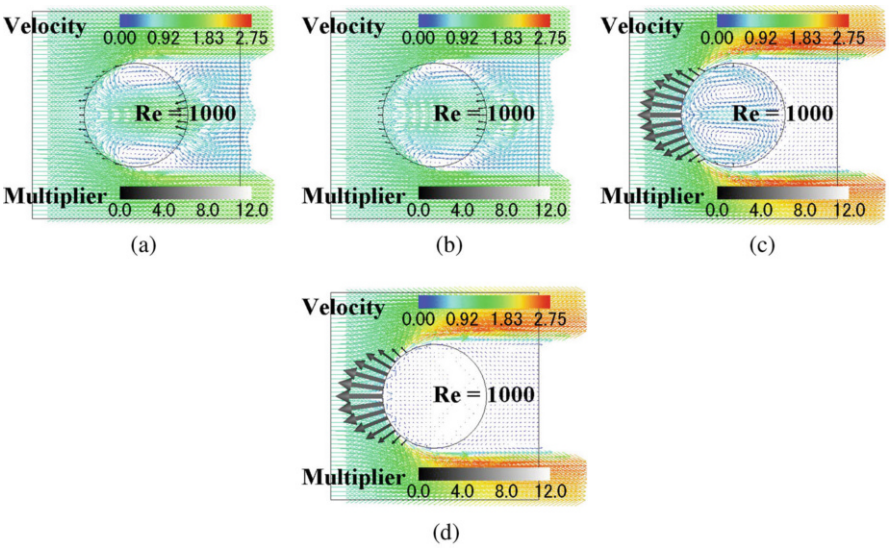


Fig. 14 Continuous versus discontinuous interfaces: Re $= 1000$ with $n_f = 48$ and $\epsilon = 1$. (**a**) For Γ_i^{**}. (**b**) For Γ_i^{E*}. (**c**) For Γ_i^{*H}. (**d**) For Γ_i^{EH}

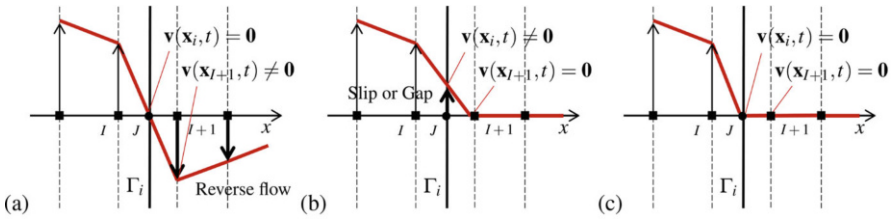


Fig. 15 Continuous versus discontinuous interfaces: (**a**) no-slip constraint without gradient discontinuity; (**b**) numerical slip for continuous function; (**c**) no-slip constraint with gradient discontinuity

the cylinder. Although the results for Γ_i^{**} and Γ_i^{E*} seem out of the question, they clearly show that pressure becomes dominant as the Reynolds number is increased. Hence, Γ_i^{**} and Γ_i^{E*} underestimate the multipliers at an unacceptable level for $\mathrm{Re} = 1000$. The inner-domain flow is also in an unacceptable level and exhibits an even stronger twin vortex. The flow is on the brink of penetration. For Γ_i^{EH}, these errors are not observed because of the existence of interfacial discontinuities. The convergence rate of Γ_i^{EH} shown in [88] is comparable to that of the interface-fitted mesh. These results also remind us of the following:

1. When the FSI condition is imposed at the interface, we have to pay attention to the numerical coupling of the flows on both sides. This coupling cannot be neglected as the Reynolds number is increased.
2. Generally, an interface with no pressure discontinuity is unacceptable because it causes the multiplier to be underestimated and thus also the coupling forces. However, there is room for discussion in the case of Stokes flow.
3. One way to model the discontinuity is to use elements such as Q1P0 and P1P0; however, these are considered to show only first-order convergence of the discontinuity. The numerical performance is classified as Γ_i^{*H}. By contrast, the enriched Q1Q1 and P1P1 elements show second-order convergence [88] and are more compatible with the SUPG/PSPG stabilization methods [97, 100, 101, 105, 107, 109].
4. Weak velocity discontinuity decouples the flows into separate ones on either side and also makes the multiplier smoother. Figure 15 illustrates the underlying mechanism [16]. If the velocity function space has no gradient discontinuity, the no-slip condition at the interface cannot be satisfied without flow in the reverse direction, as shown in Fig. 15a; otherwise, numerical slip or a gap is required on the interface, as shown in Fig. 15b. The former case explains the twin vortex stimulated in the inner domain and the hour-glass mode behavior of the multiplier observed for Γ_i^{*H}. Figure 15c corresponds to Γ_i^{EH}.

In the aforementioned ways, XFEM for FSI interfaces and the present results provide numerical data with which to discuss FSI interfaces.

8 Applications to FTSI Problems

This section shows 2D and 3D applications of the present method to FTSI problems. The method was summarized in Sect. 5. In particular, the fluttering of flexible thin objects in a flow and its numerical simulation are introduced for benchmarking the method and techniques. The FTSI problem contains essential difficulties of FSI dynamics [120–128] and its numerical simulation is still classified as a theme worth challenging because of the richness of the dynamics and the difficulties in handling large deformations and complex geometries.

8.1 Flutter of a Flexible Filament in a Flow

We introduce 2D numerical simulations of a flexible filament in a flowing soap flow. The given parameters are based mainly on the experiments of Zhang and others [121], who studied the dynamics of flexible filaments as a model for 1D flags in a 2D flow and revealed interesting quasi-statics and dynamics. Numerical simulations of such a flow-induced flutter of a flexible object are reported in [13, 14, 56, 59, 88, 91]. In [13] and [14], an IT ALE FEM [7], which has an interface-fitted mesh, and a superposed-mesh version of the ALE method [14, 16] are used for the simulation, respectively. In [88], the present method is applied, and in [56, 59], an IB method [54, 55] is employed.

Setup Our computation approximates the experiment with a mesh of one-layer 3D elements under 2D constraints. Therefore, the parameters are written in 3D space. We set a filament with its leading edge fixed in a flow channel. The length, span, and thickness of the channel are $3.5L$, $2.0L$, and $3.0\,\mu$m, respectively, where $L = 3.6$ cm is the length of the filament. The flow slips on the boundaries of the channel. Figure 16a shows an enlarged view of the regular Eulerian fluid mesh around the flapping filament; its resolution ratio to the Lagrangian mesh of the filament is one ($\epsilon = 1$). The density, Young's modulus, and Poisson's ratio of the shell elements that discretize the filament are 1.132×10^3 kg/m^3, 4.0 MPa, and zero, respectively. The thickness and depth-direction width are 0.130 mm and 0.136 mm, respectively. The density and viscosity of the fluid are 1.0×10^3 kg/m^3 and 0.12 Pa·s, respectively. The viscosity is increased from that of a soap solution to reduce the computational cost and to refer to other simulations introduced above. The inflow is uniform at a speed V of 2.2 m/s, and Reynolds number defined by the filament length is 660. Gravity acts on the filament and the flow; the acceleration due to gravity is 9.8 m/s^2.

Results Figure 17a shows the simulation results. The left panels show the time history of the horizontal displacement of the trailing edge of the filament. The others show three snapshots in half a period of the local velocity norm of the flow, which is normalized by the inflow speed V. There is no external disturbance except for the inflow; nevertheless, the filament begins to flap and settles into a sustained

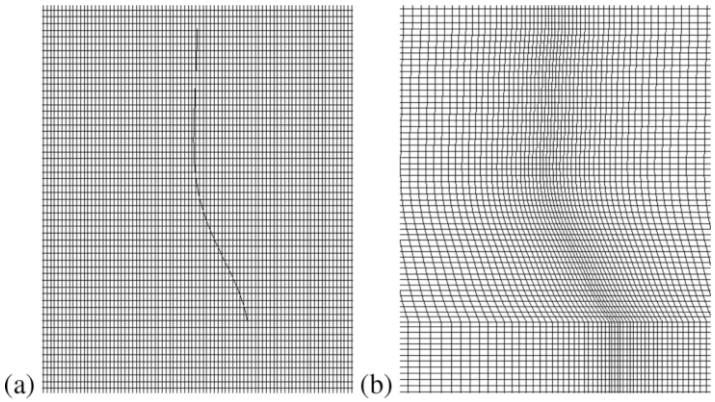


Fig. 16 Finite element meshes for a flexible filament in a flow: (**a**) IC mesh for XFEM-LLM method; (**b**) IT mesh for ALE method

flapping state under this condition. In [88], we also showed that if the interface does not have a pressure discontinuity, namely for Γ_i^{**} or Γ_i^{E*}, the filament cannot flap. The computed frequency of flapping is approximately 39.5 Hz. In [121], the frequency recorded experimentally at 2.2 m/s is not stated unfortunately; however, the frequency at 2.8 m/s is given as about 51 Hz. It is known experimentally [124] and numerically [7] that the flapping frequency is approximately proportional to the inflow speed because of the existence of a dominant traveling wave. The equivalent value at 2.2 m/s is about $51 \times 2.2/2.8 \approx 40.1$ Hz. Therefore, the computed frequency is deemed to be reasonable. The flapping shapes shown in Fig. 17a are in good agreement with an experimental snapshot shown in [121]. It is also known that the filament does not flap when its mass (strictly speaking, the mass ratio to that of the surrounding fluid) is smaller than some value [7, 56, 121, 128]. Figure 17b shows the results for such a condition, in which the mass is half the experimental value. The filament indeed does not flap but instead remains in what is known as the dynamically stretched-straight state [121].

IRC Versus ALE We compute the flapping state with an IT ALE FEM [7] to verify the IRC computation numerically. The IT mesh is shown in Fig. 16b, where a flapping filament is almost in the same place as in Fig. 16a. The ALE has nearly the same resolution as that of the Eulerian mesh. Figure 17c shows the results. The time step used for the ALE computation is the same as that for the IRC computation. Although there is a discrepancy between the IRC and ALE computations if compared in detail, this comparison shows that the present IRC method gives almost the same result with almost the same resolution as the IT ALE method.

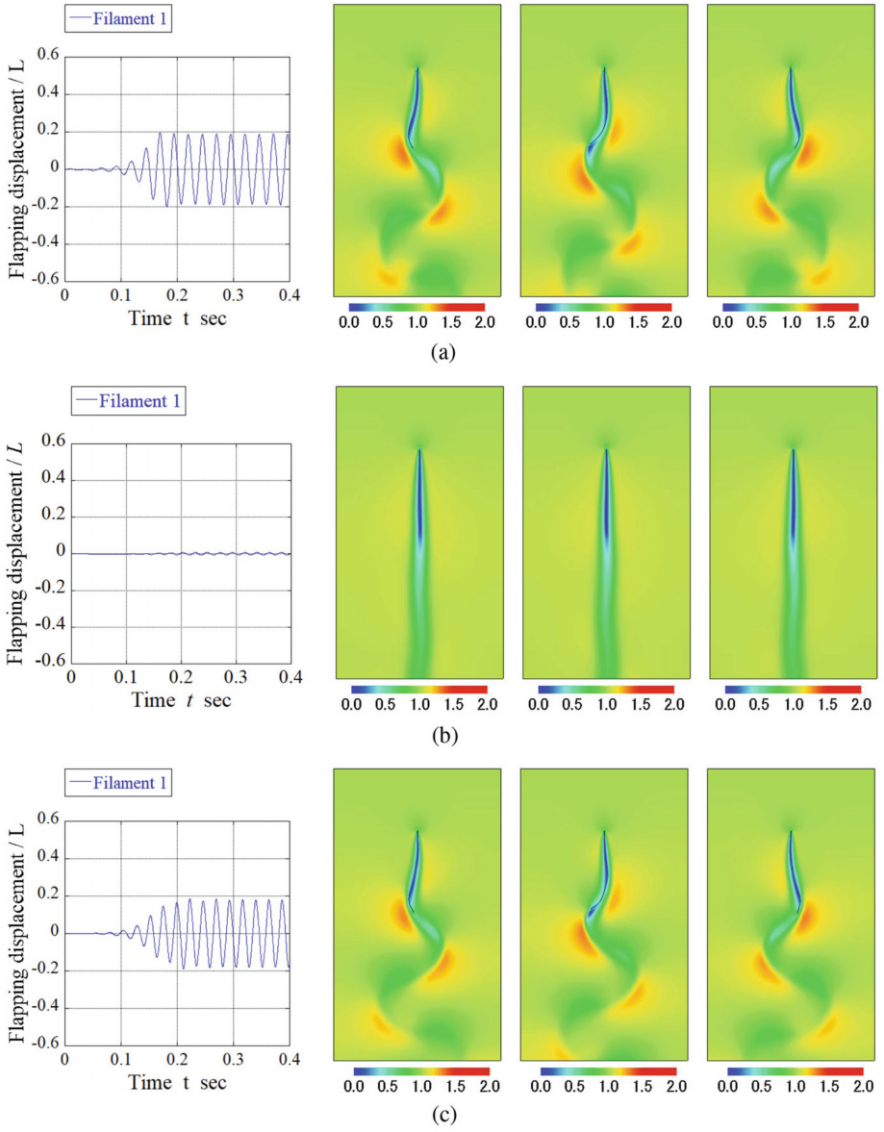


Fig. 17 Flexible filament in a flow: (graphs) horizontal displacement of a trailing edge of a filament; (images) numerical snapshots of norm of local flow velocity. (**a**) Results for IRC FTSI method. (**b**) Half mass: results for IRC FTSI method. (**c**) Results for IT ALE FTSI method

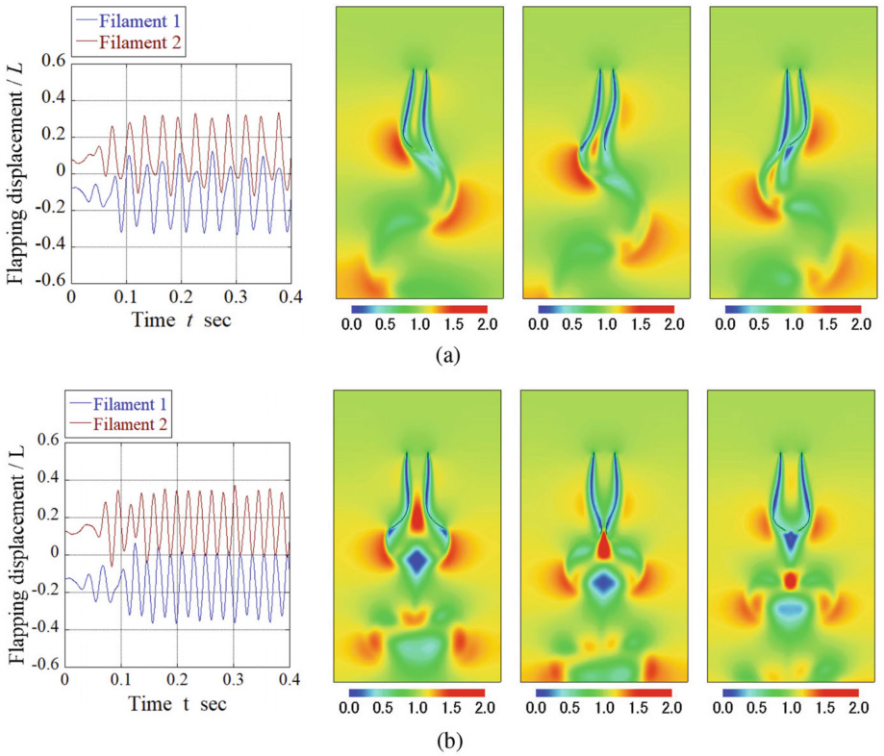


Fig. 18 Coupled flutter of two parallel filaments: (graphs) horizontal displacements of trailing edges; (images) snapshots of local flow velocity; (parallel distance) d/L. (**a**) In-phase flapping state: $d/L = 0.15$. (**b**) Anti-phase flapping state: $d/L = 0.25$

8.2 Coupled Flutter of Flexible Filaments in a Flow

In [121, 126], it is also shown experimentally that two parallel filaments in a flow flap either in phase or in anti-phase with each other depending on the distance between them. They flap in phase when they are closer together but in anti-phase when they are farther apart. Of course, if the separation is large enough to sever the coupling of the two filaments, they flap independently. For the experiment of [121], the critical d distance separating in-phase and anti-phase flapping is given in the range $d/L = 0.21 \pm 0.04$ for $L = 3.6\,\text{cm}$ and $V = 2.2\,\text{m/s}$. Although the parameter values are either partially or totally different from those used in the experiments, numerical simulations that capture the two states are reported in [57, 92].

In Parallel We computed the cases for $d/L = 0.15$ and 0.25. The results are shown in Fig. 18a, b, respectively. Each state is reproduced at the reasonable distance. It is stated in [121] that the frequency of anti-phase flapping is approximately 35%

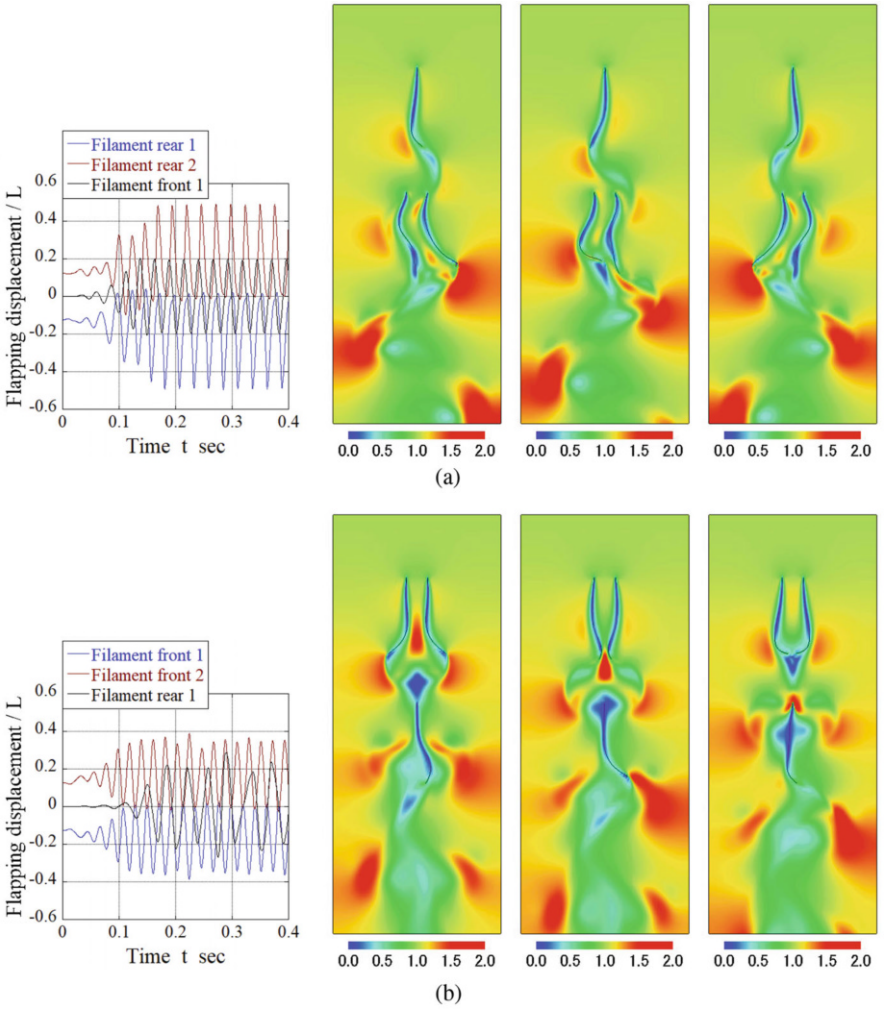


Fig. 19 Coupled flutter of three filaments: (graphs) horizontal displacements of trailing edges; (images) snapshots of local flow velocity; (parallel distance) $d/L = 0.30$. (**a**) One in front and two behind. (**b**) Two in front and one behind

higher than that of in-phase flapping. The computed frequencies are 48.0 Hz and 33.0 Hz, respectively, which represents an increase of roughly 45%.

In Tandem Figure 18a, b shows the cases of one filament in front and two parallel filaments behind it and vice versa, respectively. A similar simulation for two tandem filaments (one in front and one behind) can be found in [60]. Normally, two filaments in parallel at this distance ($d/L = 0.30$) flap in anti-phase, as is the case for the front filaments of Fig. 19b. However, for the case of Fig. 19a,

parallel filaments behind a flapping filament flap in phase with each other. Therefore, the front filament is forcing the parallel filaments behind it to flap in phase. The amplitudes are amplified by the flapping of the front filament. The frequencies of the three filaments are the same, approximately 39.5 Hz. The Reynolds number defined by twice the length of the filament is 1320. These applications demonstrate that the present method can capture the physics of and can handle complex FTSI problems.

8.3 Flapping Flag in a Wind

We carried out a 3D application involving a flag flapping in a wind [90] to check the applicability of the present IRC method to FTSI problems with large deformation. Therefore, the parameter values are partially artificial to amplify the deformation while reducing the computational cost.

Setup The flag is 42 cm long and 30 cm wide. Its density, Young's modulus, Poisson's ratio, and thickness are $1.0 \times 10^3\,\mathrm{kg/m^3}$, 0.1 GPa, 0.3, and 1.0 mm, respectively. Assuming that the flag is made of cloth, the in-plane anisotropy of the cloth is modeled with a shear stiffness that is 3.0×10^{-4} times smaller than the isotropic value. The density of the fluid is that of ordinary air, namely $1.2\,\mathrm{kg/m^3}$, the wind speed is 10 m/s, and normal gravity acts on the flag. The fluid mesh used is not fine enough to resolve actual boundary layers. Therefore, the viscosity is increased so that the Reynolds number defined by the length of the flag becomes 1000. The Lagrangian mesh of the flag has a resolution ratio of $\epsilon \approx 1.2$, which prevents the computation from locking. When two interfaces enter one fluid element because of the large deformation, our FTSI program approximates the two as one discontinuous surface.

Results Figure 20 shows the results for the flag flapping in the wind under gravity. The previous applications to flexible filaments in a flow are 2D ones, whereas here it is 3D. Each snapshot visualizes the flow around the flapping flag with streamlines and velocity vectors at a cross section with colors representing local norms. Time progresses from the upper-left snapshot to the lower-right snapshot. Although the Reynolds number has been lowered, this simulation captures well the dynamical competition between the flapping motion due to the wind and the drooping motion due to gravity, which we see often in everyday life. We confirm the applicability of the IRC method to an FTSI problem with large 3D deformation.

8.4 Dynamic Response of Papers to Air Blasts

As a more challenging application example, we introduce 3D simulations of the dynamical behavior of a stack of papers to blasts of air. These simulations were conducted in joint research with a company, so we are not allowed to describe the

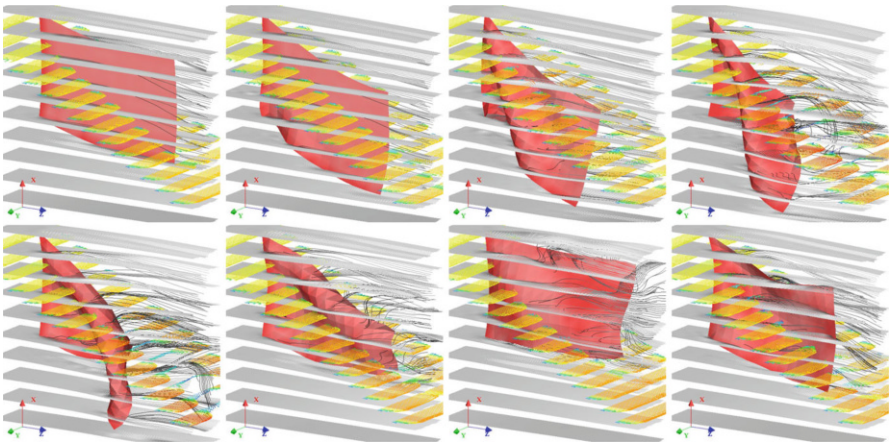


Fig. 20 3D simulation of a flag in a wind under gravity by IRC FTSI method

simulation conditions and material parameters exactly. Instead, we show some early stage results that we reported in [87] in Japanese language.

Setup This application starts with placing a stack of papers horizontally with two front points fixed in a rectangular computational domain, as shown in the upper figures of Figs. 21 and 22. There are small inlets on the left and right sides of the domain, and air blasts from these inlets flutter the papers. The bottom and sides of the domain except for the inlets are no-slip and slip boundaries, respectively, while the top is a traction-free open boundary. The FTSI method as it stands is unable to handle contact between the papers. Therefore, there are small gaps between the papers at the start of the FTSI computation. To resolve the dynamically changing gaps, a thinly layered fluid mesh is used. To reduce the computational cost of the test simulation, the air flow is computed only in the rectangular domain. For those parts of the papers that leave the domain, only the structural computation is continued; namely, no LMs are computed outside the domain. The Reynolds number of the gap flow is estimated to be roughly 3000. The parameter values are all real ones.

Results Figures 21 and 22 show the dynamical responses of the papers and the air flows for the cases of five thick and thin pieces of paper, respectively. The air flows gradually into the gaps and opens the papers, and the five pieces of paper ascend nearly sequentially. The dynamical response is more intricate for thin papers than it is for thick papers. These results agree with experimental observations and tell us that the flow of air passes into the gaps. From the spatial profile of the gap flow, we can confirm no-slip conditions and discontinuities of the flow at and across the papers. Although more accurate and real simulation is needed to advance the FTSI method to compute the physics further (e.g., contact, friction, electrostatic effects), the results suggest that the FTSI method is a base for doing so.

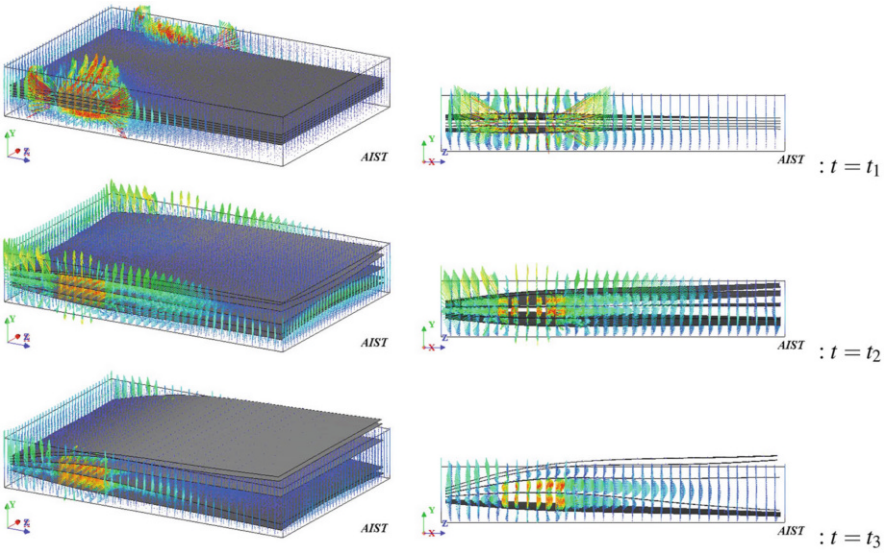


Fig. 21 3D simulation of five thick pieces of paper with air blasts from the right and left inlets by the IRC FTSI method. Velocity vectors of the flow are visualized at three time points by coloring the norm: (left) diagonal views; (right) side views

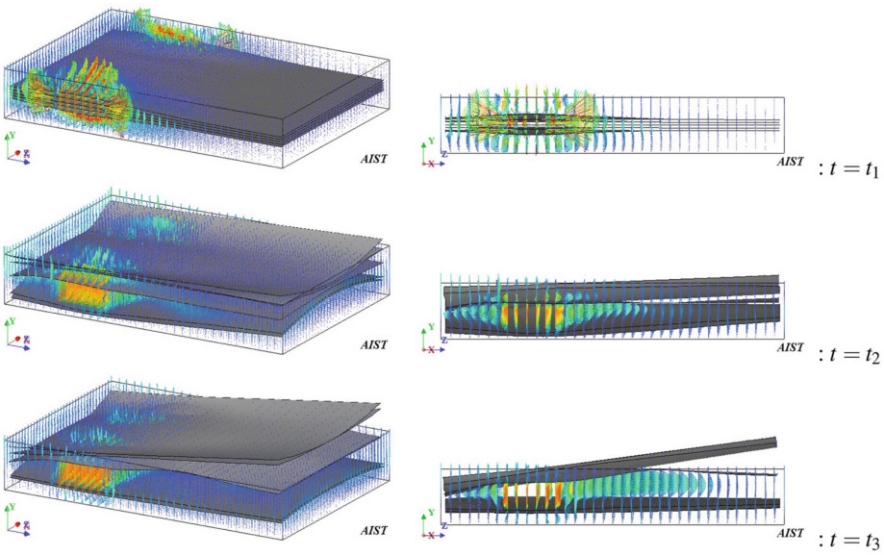


Fig. 22 3D simulation of five thin pieces of paper with air blasts from the right and left inlets by the IRC FTSI method. Velocity vectors of the flow are visualized at three time points by coloring the norm: (left) diagonal views; (right) side views

9 Applications to Flow Past/in a Fixed Object

The XFEM and LM method for FTSI problems is the full-set version of the computation program. However, as described in Sect. 3.3, the method can be simplified to compute flow past/in a fixed object. The simplified version is given by Eqs. (42) and (37) and meets the no-slip condition on the interface and does not require LM computation to impose the no-slip condition, which enables us to automate computation of flow past/in a fixed object.

9.1 Flow Past Fixed Objects

Figure 23 shows an automated process for computing the flow past a cylinder fixed in a slip channel [89]. The left and right sides are the inlet and outlet of the channel, respectively. We begin by putting the surface data (mesh) of the cylinder into the computational domain of the flow. The program generates the fluid mesh and computes the level-set function $f(\mathbf{x}, t)$ of Fig. 23a from the surface of the cylinder automatically. The level-set function gives the one-sided edge function $e^{+}(\mathbf{x}, t)$ of Fig. 23b. The program then begins to compute the flow. The results for velocity and pressure at a particular sample time are shown in Fig. 23c, d, respectively. The Reynolds number is 1000. Equations (42) and (37) can compute the flow without the usual need for manual pre-processing such as interface-fitted mesh generation or setting up the boundary conditions. Figure 24 shows a 3D application involving real objects whose surface data were obtained by 3D scanning. The Stereolithography (STL) data file outputted by the scanner defines the surface of each object. We have to fix the STL file, but the rest of the process is automated and is the same as that

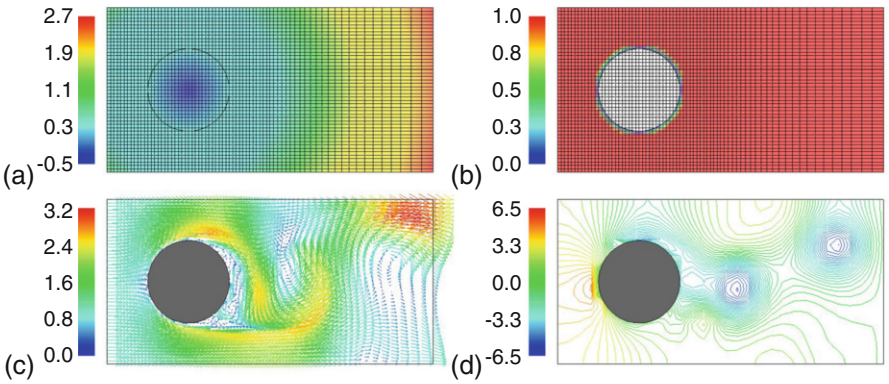


Fig. 23 2D computation of flow past a fixed cylinder with function space given by Eqs. (42) and (37): (**a**) fluid mesh and level-set function $f(\mathbf{x}, t)$; (**b**) one-sided edge function $e^{+}(\mathbf{x}, t)$; results for (**c**) velocity and (**d**) pressure of the flow

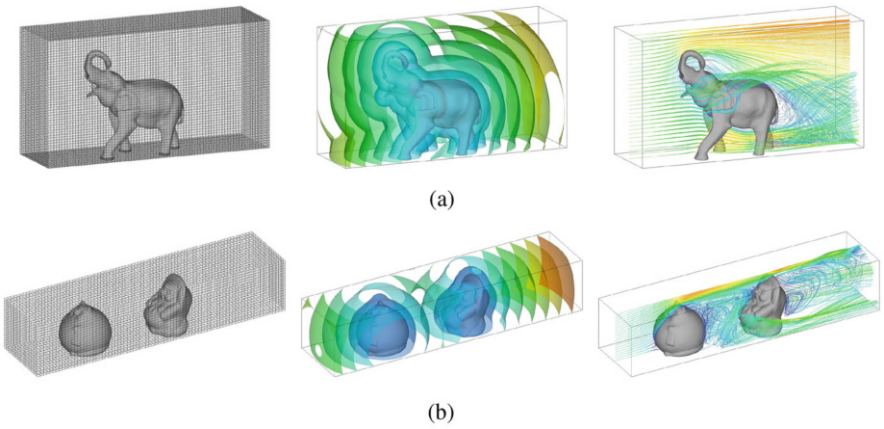


Fig. 24 3D computation of flows past fixed objects with function space given by Eqs. (42) and (37): (left) structured fluid mesh and surface data of real object(s) obtained by 3D scanning; (center) automatically computed level-set function; (right) streamlines visualizing flow past object(s). (**a**) Flow past an elephant ornament. (**b**) Flow past raccoon ornaments arranged in front and behind

used in the 2D application. As shown in the previous section, the XFEM-based method enables us to take on the challenge of computing complex FSI problems. It also enables us to compute simple flow problems such as these in an elegant manner.

9.2 Flow in a Microstructure

Application to flow in a periodic microstructure is introduced in [75]. The XFEM-based method is also useful for modeling microstructures that have fluid and solid phases. The application shown in Fig. 25 models the two phases of a periodic body-centered cubic (BCC) structure with a regular voxel-type mesh. To compute the flow in the gaps of the BCC structure, we apply Eqs. (42) and (37) to the Stokes equation, and to compute the stress distribution of the BCC structure, we apply Eq. (36) to the displacement function spaces to solve the linear elasticity equation. The pressure of the gap flow acts on the BCC structure. In practice, the formulation is extended to a homogenization method for solid–fluid mixtures that computes the macroscopic flow and stress with the homogenized material parameters of elasticity, FSI compressibility, and permeability given by the microscopic computation shown in Fig. 25. The validity of the IRC computation is shown in [75], compared with results given by a surface-fitted mesh.

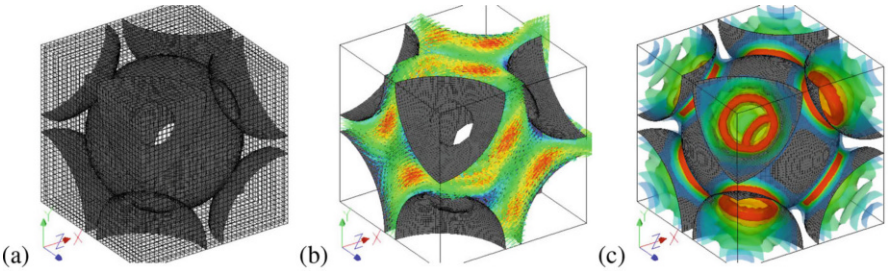


Fig. 25 Flow in a BCC structure: (**a**) surface data of structure and voxel mesh; (**b**) Stokes flow in gap at given pressure gradient; (**b**) von Mises stress of BCC structure at given fluid pressure

10 Concluding Remarks

This chapter gives a brief overview of the interface-reproducing capturing (IRC) method based on extended finite element method (XFEM) and a Lagrange multiplier method, and it introduces our contributions to the development of the IRC method and techniques for fluid–thin-structure interaction (FTSI). The presented IRC method and techniques open a new framework and concept whose function is locally equivalent to that of interface-tracking (IT) methods. We show verifications of the IRC FTSI method in computing flow with a domain-partitioning interface (DPI). The DPI problem causes no problems for IT methods, which have interface-fitted coordinates. However, for IC methods, the DPI problem causes a lot of problems and reveals the essence of modeling an FSI interface numerically. The two- and three-dimensional applications to flow-induced flutter of flexible thin objects demonstrate the ability and potential of the IRC method. These applications include data not published in international journals. To the best of our knowledge, there are only a few published comparisons of the computational results of IT and IC methods obtained under the same condition, namely the resolution. The application to flows past fixed objects shows another aspect of the IRC method, which enables us to take on the challenge of computing complex FSI problems. At the same time, it enables us to compute simple problems with ease. We also describe open issues associated with the IRC approach. The existence of problems that still require solutions indicates that there is still much room for advancement.

Acknowledgements I would like to thank Professor Tayfun E. Tezduyar at Rice University, USA, for useful comments on advanced FSI and stabilization techniques. I would also like to thank Dr. Jun-ichi Matsumoto at AIST, Japan, for daily discussions about computational flow techniques.

References

1. Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian–Eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl Mech Engrg 29:329–349.
2. Belytschko T, Flanagan DP, Kennedy J (1982) Finite element method with user-controlled meshes for fluid–structure interactions. Comput Methods Appl Mech Engrg 33:689–723.
3. Huerta A, Liu WK (1988) Viscous flow with large free surface motion. Comput Methods Appl Mech Engrg 69:277–324.
4. Huerta A, Liu W (1988), Viscous flow structure interaction. J Pressure Vessel Tech 110:15–21.
5. Nitikitpaiboon C, Bathe KJ (1993) An arbitrary Lagrangian–Eulerian velocity potential formulation for fluid–structure interaction. Comput Struct 47(4):871–891.
6. Bathe K, Zhang H, Wang M (1995) Finite element analysis of incompressible and compressible fluid flows with free interfaces and structural interactions. Comput Struct 56:193–213.
7. Zhang Q, Hisada T (2001) Analysis of fluid–structure interaction problems with structural buckling and large domain changes by ALE finite element method. Comput Methods Appl Mech Engrg 190:6341–6357.
8. Watanabe H, Hisada T, Sugiura S, Okada J, Fukunari H (2002) Computer simulation of blood flow, left ventricular wall motion and their interrelationship by fluid–structure interaction finite element method. JSME Int J Ser C 45(4):1003–1012.
9. Kuhl E, Hulshoff S, Borst DR (2003) An arbitrary Lagrangian Eulerian finite-element approach for fluid–structure interaction phenomena. Int J Numer Meth Engng 57:117–142.
10. Watanabe H, Sugiura S, Hisada T (2004) Multiphysics simulation of left ventricular filling dynamics using fluid–structure interaction finite element method. Biophys J 87(3):2074–2085.
11. Ishihara D, Yoshimura S (2005) A monolithic approach for interaction of incompressible viscous fluid and an elastic body based on fluid pressure Poisson equation. Int J Numer Meth Engng 64:167–203.
12. Sawada T, Hisada T (2006) Fluid–structure interaction analysis of a two-dimensional flag-in-wind problem by the ALE finite element method. JSME Int J Ser A 49(2):170–179.
13. Sawada T, Hisada T (2007) Fluid–structure interaction analysis of the two-dimensional flag-in-wind problem by an interface-tracking ALE finite element method. Comput Fluids 36:136–146.
14. Sawada T, Tezuka A, Hisada T (2007) Overlaying mesh method for large deformation fluid–shell interaction analysis using interface-tracking ALE local mesh and immersed boundary global mesh. Trans JSCES 20070029:1–10 (in Japanese Language).
15. Bazilevs Y, Hughes TJR (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput Fluids 36:12–26.
16. Sawada T, Tezuka A, Hisada T (2008) Performance comparison between the fluid–shell coupled overlaying mesh method and the immersed boundary method. Trans JSCES 20080005:1–14 (in Japanese Language).
17. Ishihara D, Horie Y, Denda M (2009) Two dimensional computational study on fluid–structure interaction cause of wing pitch changes in dipteran flapping flight. J Exp Bio 212:1–10.
18. Bazilevs Y, Hsu M-C, Scott MA (2012) Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. Comput Methods Appl Mech Engrg 249–252: 28–41.
19. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015) An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. Comput Methods Appl Mech Engrg 284: 1005–1053.

20. Tezduyar TE, Behr M, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. Comput Methods Appl Mech Engrg 94:339–351.
21. Tezduyar TE, Behr M, Mittal S, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space–time procedure: II. Computations of free-surface flows, two-liquid flows, and flows with drifting cylinders. Comput Methods Appl Mech Engrg 94:353–371.
22. Tezduyar TE, Behr M, Mittal S, Johnson AA (1992) Computation of unsteady incompressible flows with the finite element methods: Space–time formulations, iterative strategies and massively parallel implementations. New Methods in Transient Analysis, PVP-Vol.246/AMD-Vol.143, ASME, New York, 7–24.
23. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S (1993) Parallel finite-element computation of 3D flows. Computer 26(10):27–36.
24. Johnson AA, Tezduyar TE (1994) Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. Comput Methods Appl Mech Engrg 119:73–94.
25. Mittal S, Tezduyar TE (1995) Parallel finite element simulation of 3D incompressible flows – fluid–structure interactions. Int J Numer Meth Fluids 21:933–953.
26. Kalro V, Tezduyar TE (2000) A parallel 3D computational method for fluid–structure interactions in parachute systems. Comput Methods Appl Mech Engrg 190:321–332.
27. Tezduyar T, Osawa Y (2001) The multi-domain method for computation of the aerodynamics of a parachute crossing the far wake of an aircraft. Comput Methods Appl Mech Engrg 191:705–716.
28. Tezduyar TE (2001) Finite element methods for flow problems with moving boundaries and interfaces. Arch Comput Meth Engng 8:83–130.
29. Stein K, Tezduyar T, Benney R (2003) Mesh moving techniques for fluid–structure interactions with large displacements. J Appl Mech 70:58–63.
30. Stein K, Tezduyar TE, Benney R (2004) Automatic mesh update with the solid-extension mesh moving technique. Comput Methods Appl Mech Engrg 193:2019–2032.
31. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE (2006) Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation. Comput Methods Appl Mech Engrg 195:1885–1895.
32. Tezduyar TE, Sathe S, Keedy R, Stein K (2006) Space–time finite element techniques for computation of fluid–structure interactions. Comput Methods Appl Mech Engrg 195:2002–2027.
33. Tezduyar TE (2006) Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. Comput Methods Appl Mech Engrg 195:2983–3000.
34. Akin JE, Tezduyar TE, Ungor M (2007) Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT). Comput Fluids 36:2–11.
35. Tezduyar TE, Sathe S (2007) Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques. Int J Numer Meth Fluids 54:855–900.
36. Tezduyar TE, Sathe S, Cragin T, Nanna B, Conklin BS, Pausewang J, Schwaab M (2007) Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics. Int J Numer Meth Fluids 54:901–922.
37. Cruchaga MA, Celentano DJ, Tezduyar TE (2007) A numerical model based on the mixed interface-tracking/interface-capturing technique (MITICT) for flows with fluid–solid and fluid–fluid interfaces. Int J Numer Meth Fluids 54:1021–1030.
38. Sathe S, Tezduyar TE (2008) Modeling of fluid–structure interactions with the space–time finite elements: Contact problems. Comput Mech 43:51–60.
39. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE (2008) Fluid–structure interaction modeling of a patient-specific cerebral aneurysm: Influence of structural modeling. Comput Mech 43:151–159.

40. Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J (2010) Multiscale sequentially-coupled arterial FSI technique. Comput Mech 46:17–29.
41. Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J (2010) Space–time finite element computation of complex fluid–structure interactions. Int J Numer Meth Fluids 64:1201–1218.
42. Takizawa K, Christopher J, Tezduyar TE, Sathe S (2010) Space–time finite element computation of arterial fluid–structure interactions with patient-specific data. Int J Numer Meth Biomed Engng 26:101–116.
43. Takizawa K, Tezduyar TE (2011) Multiscale space–time fluid–structure interaction techniques. Comput Mech 48(3):247–267.
44. Takizawa K, Moorman C, Wright S, Spielman T, Tezduyar TE (2011) Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes. Int J Numer Meth Fluids 65:271–285.
45. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE (2011) Influencing factors in image-based fluid–structure interaction computation of cerebral aneurysms. Int J Numer Meth Fluids 65:324–340.
46. Takizawa K, Tezduyar TE (2012) Computational methods for parachute fluid–structure interactions. Arch Comput Meth Engng 19:125–169.
47. Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. Arch Comput Meth Engng 19:171–225.
48. Takizawa K, Tezduyar TE (2012) Space–time fluid–structure interaction methods. Mathematical Models and Methods in Applied Sciences 22(supp02):1230001.
49. Takizawa K, Tezduyar TE (2014) Main aspects of the space–time computational FSI techniques and examples of challenging problems solved. Mechanical Engineering Reviews 1:CM0005, inaugural issue.
50. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space–time interface-tracking with topology change (ST-TC). Comput Mech 54:955–971.
51. Takizawa K, Tezduyar TE, Kolesar R, Boswell C, Kanai T, Montel K (2014) Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes. Comput Mech 54:1461–1476.
52. Takizawa K, Bazilevs Y, Tezduyar TE, Long CC, Marsden AL, Schjodt K (2014) ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling. Mathematical Models and Methods in Applied Sciences 24:2437–2486.
53. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Øiseth O, Mathisen KM, Kostov N, McIntyre S (2014) Engineering analysis and design with ALE-VMS and space–time methods. Arch Comput Meth Engng 21:481–508.
54. Peskin CS (1972) Flow patterns around heart valves: A numerical method. J Comput Phys 10:252–271.
55. Peskin CS (2002) The immersed boundary method. Acta Numerica 11:479–517.
56. Zhu L, Peskin CS (2002) Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. J Comput Phys 179:452–468.
57. Zhu L, Peskin CS (2003) Interaction of two flapping filaments in a flowing soap film. Phys Fluids 15(7):1954–1960.
58. Peskin C (1977) Numerical analysis of blood flow in the heart. J Comput Phys 25:220–252.
59. Huang WX, Shin SJ, Sung HJ (2007) Simulation of flexible filaments in a uniform flow by the immersed boundary method. J Comput Phys 226:2206–2228.
60. Zhu L (2009) Interaction of two tandem deformable bodies in a viscous incompressible flow. J Fluid Mech 635:455–475.
61. Li Z (1997) Immersed interface methods for moving interface problems. Numerical Algorithms 14(4):269–293.
62. Boffi D, Gastaldi L (2003) A finite element approach for the immersed boundary method. Comput Struct 81:491–501.

63. Wang X, Liu WK (2004) Extended immersed boundary method using FEM and RKPM. Comput Methods Appl Mech Engrg 193:1305–1321.
64. Zhang L, Gerstenberger A, Wang X, Liu WK (2004) Immersed finite element method. Comput Methods Appl Mech Engrg 193:2051–2067.
65. Wang H, Chessa J, Liu WK, Belytschko T (2008) The immersed/fictitious element method for fluid–structure interaction: Volumetric consistency, compressibility and thin members. Int J Numer Meth Engng 74:32–55.
66. Glowinski R, Pan TW, Périaux J (1998) Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. Comput Methods Appl Mech Engrg 151:181–194.
67. Glowinski R, Pan TW, Hesla TI, Joseph DD, and Périaux J (1999) A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application of particulate flow. Int J Numer Meth Fluids 30:1043–1066.
68. Hart DJ, Peters GWM, Schreurs PJG, Baaijens FPT (2000) A two-dimensional fluid–structure interaction model of the aortic valve. J Biomech 33:1079–1088.
69. Hart DJ, Baaijens FPT, Peters GWM, Schreurs PJG (2003) A computational fluid–structure interaction analysis of a fiber-reinforced stentless aortic valve, J Biomech 36:699–712.
70. Loon RV, Anderson PD, Hart DJ, Baaijens FPT (2004) A combined fictitious domain/adaptive meshing method for fluid–structure interaction in heart valves. Int J Numer Meth Fluids 46:533–544.
71. Yu Z (2005) A DLM/FD method for fluid/flexible-body interactions. J Comput Phys 207(1):1–27.
72. Wagner GJ, Moës N, Liu WK, Belytschko T (2001) The extended finite element method for rigid particles in Stokes flow. Int J Numer Meth Engng 51:293–313.
73. Wagner GJ, Ghosal S, Liu WK (2003) Particulate flow simulations using lubrication theory solution enrichment. Int J Numer Meth Engng 56:1261–1289.
74. Chessa J, Belytschko T (2003) An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension. Int J Numer Meth Engng 58:2041–2064.
75. Sawada T, Nakasumi S, Tezuka A, Fukushima M, Yoshizawa Y (2009) Extended-FEM for the solid–fluid mixture two-scale problems with BCC and FCC microstructures. Interaction & Multiscale Mech Int J 2(1):45–68.
76. Legay A, Chessa J, Belytschko T (2006) An Eulerian–Lagrangian method for fluid–structure interaction based on level sets. Comput Methods Appl Mech Engrg 195:2070–2087.
77. Legay A, Kölke A (2006) An enriched space–time finite element method for fluid–structure interaction – Part I: Prescribed structural displacement. Proc III ECCM, Solid Struct Coupling Prob Eng, Lisbon Portugal, 5–8 Jun 2006.
78. Kölke A, Legay A (2006) An enriched space–time finite element method for fluid–structure interaction – Part II: Thin flexible structures. Proc III ECCM, Solid Struct Coupling Prob Eng, Lisbon Portugal, 5–8 Jun 2006.
79. Sawada T, Tezuka A, Hisada T (2007) Extended finite element method for the fluid–structure interaction problems based on discontinuous interpolations on level set interfaces. Proc APCOM'07–EPMESC XI, MS20-3(2):1–10, Kyoto Japan, 3–6 Dec 2007.
80. Zilian A, Legay A (2008) The enriched space–time finite element method (EST) for simultaneous solution of fluid–structure interaction. Int J Numer Meth Engng 75:305–334.
81. Gerstenberger A, Wall WA (2008) An extended finite element method/Lagrange multiplier based approach for fluid–structure interaction. Comput Methods Appl Mech Engrg 197:1699–1714.
82. Gerstenberger A, Wall WA. (2008) Enhancement of fixed-grid methods towards complex fluid–structure interaction applications. Int J Numer Meth Fluids 57:1227–1248.
83. Mayer UM, Gerstenberger A, Wall WA (2009) Interface handling for three-dimensional higher-order XFEM-computations in fluid–structure interaction. Int J Numer Meth Engng 79:846–869.
84. Gerstenberger A, Wall WA (2010) An embedded Dirichlet formulation for 3D continua. Int J Numer Meth Engng 82:537–563.

85. Mayer UM, Popp A, Gerstenberger A, Wall WA (2010) 3D fluid–structure–contact interaction based on a combined XFEM FSI and dual mortar contact approach. Comput Struct 46:53–67.
86. Sawada T, Tezuka A (2010) High-order Gaussian quadrature in X-FEM with the Lagrange-multiplier for fluid–structure coupling. Int J Numer Meth Fluids 64:1219–1239.
87. Nakamoto H, Sawada T, Hattori S, Tezuka A (2010) Advanced simulation technology for innovating air-assisted paper-feed mechanism. Toshiba Review 65(8):35–39 (in Japanese Language).
88. Sawada T, Tezuka A (2011) LLM and X-FEM based interface modeling of fluid–thin structure interactions on a non-interface-fitted mesh. Comput Mech 48:319–332.
89. Sawada T, Nagahama S, Sasaki S, Tezuka A (2011) Development of simulation-based design (SBD) framework for flow with structure interfaces using X-FEM. Trans JSCES 20110003:1–13 (in Japanese language).
90. Sawada T (2013) Foundation and application of extended finite element method. Nagare 32:221–225 (in Japanese language).
91. Farnell DJJ, David T, Barton DC (2004) Numerical simulations of a filament in a flowing soap film. Int J Numer Meth Fluids 44:313–330.
92. Farnell DJJ, David T, Barton DC (2004) Coupled states of flapping flags. J Fluids Struct 19:29–36.
93. Cirak F, Radovitzky R (2005) A Lagrangian–Eulerian shell–fluid coupling algorithm base on level sets. Comput Struct 85:491–498.
94. Takizawa K, Yabe T, Tsugawa Y, Tezduyar TE, Mizoe H (2007) Computation of free-surface flows and fluid–object interactions with the CIP method based on adaptive meshless Soroban grids. Comput Mech 40:167–183.
95. Wang H, Belytschko T (2009) Fluid–structure interaction by the discontinuous-Galerkin method for large deformations. Int J Numer Meth Engng 77:30–49.
96. Hashimoto G, Ono K (2010) Interface treatment under no-slip conditions using level-set virtual particles for fluid–structure interaction. Theor Appl Mech Japan 58:325–342.
97. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Engrg 32:199–259.
98. Tezduyar TE, Liou J, Ganjoo DK (1990) Incompressible flow computations based on the vorticity-stream function and velocity-pressure formulations. Comput Struct 35:445–472.
99. Tezduyar TE, Mittal S, Shih R (1991) Time-accurate incompressible flow computations with quadrilateral velocity-pressure elements. Comput Methods Appl Mech Engrg 87:363–384.
100. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comput Methods Appl Mech Engrg 95:221–242.
101. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adva Appl Mech 28:1–44.
102. Hannani SK, Stanislas M, Dupont P (1995) Incompressible Navier–Stokes computations with SUPG and GLS formulations – A comparison study. Comput Methods Appl Mech Engrg 124:153–170.
103. Hughes TJR (1995) Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput Methods Appl Mech Engrg 127:387–401.
104. Hughes TJR, Feij'oo GR, Mazzei L, Quincy JB (1998) The variational multiscale method – a paradigm for computational mechanics. Comput Methods Appl Mech Engrg 166:3–24.
105. Tezduyar TE, Osawa Y (2000) Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Engrg 190:411–430.
106. Hughes TJR, Mazzei L, Jansen KE (2000) Large Eddy Simulation and the variational multiscale method. Comput Visual Sci 3:47–59.
107. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Meth Fluids 43:555–575.

108. Hughes TJR, Sangalli G (2007) Variational multiscale analysis: the fine-scale GreenAfs function, projection, optimization, localization, and stabilized methods. SIAM J Numer Anal 45:539–557.
109. Hsu M-C, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2010) Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. Comput Methods Appl Mech Engrg 199:828–840.
110. Moës N, Dolbow J, Belytschko T (1999) A finite element method for crack growth without remeshing. Int J Numer Meth Engng 46(1):131–150.
111. Belytschko T, Moës N, Usui S, Parimi C (2001) Arbitrary discontinuities in finite elements. Int J Numer Meth Engng 50(4):993–1013.
112. Sukumar N, Chopp DL, Moës N, Belytschko T (2001) Modeling holes and inclusions by level sets in the extended finite-element method. Comput Methods Appl Mech Engrg 190:6183–6200.
113. Moës N, Cloirec M, Cartraud P, Remacle JF (2003) A computational approach to handle complex microstructure geometries. Comput Methods Appl Mech Engrg 192:3163–3177.
114. Chessa J, Belytschko T (2004) Arbitrary discontinuities in space–time finite elements by level sets and X-FEM. Int J Numer Meth Engng 61:2595–2614.
115. Barbosa HJC, Hughes TJR (1991) The finite element method with Lagrange multipliers on the boundary: Circumventing the Babuska–Brezzi condition. Comput Methods Appl Mech Engrg 85(1):109–128.
116. Ji H, Dolbow JE (2004) On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. Int J Numer Meth Engng 61:2508–2535.
117. Fernández-Méndez S, Huerta A (2004) Imposing essential boundary conditions in mesh-free methods. Comput Methods Appl Mech Engrg 193(12–14):1257–1275.
118. Moës N, Béchet E, Tourbier M (2006) Imposing Dirichlet boundary conditions in the extended finite element method. Int J Numer Meth Engng 67(12):1641–1669.
119. Newmark NM (1959) A method of computation for structural dynamics. J Engng Mech Div, Proc ASCE 85(EM3):67–94.
120. Huber G (2000) Swimming in flatsea. Nature 408:777–778.
121. Zhang J, Childress S, Libchaber A, Shelley M (2000) Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. Nature 408:835–839.
122. Watanabe Y, Suzuki S, Sugihara M, Sueoka Y (2002) An experimental study of paper flutter. J Fluids Struct 16:529–542.
123. Watanabe Y, Isogai K, Suzuki S, Sugihara M (2002) An theoretical study of paper flutter. J Fluids Struct 16:543–560.
124. Shelley M, Vandenberghe N, Zhang J (2005) Heavy flags undergo spontaneous oscillations in flowing water. Phys Rev Lett 94:094302(4).
125. Eloy C, Souilliez C, Schouveiler L (2007) Flutter of a rectangular plate. J Fluids Struct 23:904–919.
126. Schouveiler L, Eloy C (2009) Coupled flutter of parallel plates. Phys Fluids 21:081703(4).
127. Alben S, Shelley MJ (2008) Flapping states of a flag in an inviscid fluid: bistability and the transition to chaos. Phys Rev Lett 100:074301(4).
128. Shelley MJ, Zhang J (2011) Flapping and bending bodies interacting with fluid flows. Annu Rev Fluid Mech 43:449–465.
129. Singh RK, Kant T, Kakodkar A (1991) Coupled shell–fluid interaction problems with degenerate shell and three-dimensional fluid elements. Comput Struct 38(5):515–528.
130. Ventura G, Gracie R, Belytschko T (2009) Fast integration and weight function blending in the extended finite element method. Int J Numer Meth Engng 77(1):1–29.
131. Mousavi SE, Sukumar N (2010) Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. Comput Mech 47:535–554.
132. Düster A, Parvizian J, Yang Z, Rank E (2008) The finite cell method for three-dimensional problems of solid mechanics. Comput Methods Appl Mech Engrg 197: 3768–3782.
133. Flemisch B, Wohlmuth BI (2007) Stable Lagrange multipliers for quadrilateral meshes of curved interfaces in 3D. Comput Methods Appl Mech Engrg 196(8):1589–1602.

134. Cho JY, Song YM, Choi YH (2008) Boundary locking induced by penalty enforcement of essential boundary conditions in mesh-free methods. Comput Methods Appl Mech Engrg 197(13–16): 167–1183.
135. Dvorkin EN, Bathe KJ (1984) A continuum mechanics based four-node shell element for general nonlinear analysis. Eng Comput 1:77–88.
136. Dvorkin EN (1988) On a non-linear formulation for curved Timoshenko beam elements considering large displacement/rotation increments. Int J Numer Meth Engng 26:1597–1613.
137. Parisch H (1991) An investigation of a finite rotation four node assumed strain shell element. Int J Numer Meth Engng 31:127–150.
138. Noguchi H, Hisada T (1993) Sensivity analysis in post-buckling problems of shell structures. Comput Struct 47(4):699–710.
139. Saad Y, Schultz MH (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7:856–869.