

## Immersogeometric analysis of compressible flows with application to aerodynamic simulation of rotorcraft

Fei Xu\*

*Department of Mechanical Engineering, Iowa State University,  
2043 Black Engineering, Ames, Iowa 50011, USA  
feixu@iastate.edu*

Yuri Bazilevs

*School of Engineering, Brown University,  
184 Hope Street, Providence, Rhode Island 02912, USA  
yuri\_bazilevs@brown.edu*

Ming-Chen Hsu

*Department of Mechanical Engineering, Iowa State University,  
2043 Black Engineering, Ames, Iowa 50011, USA  
jmchsu@iastate.edu*

Received (Day Month Year)

Revised (Day Month Year)

Communicated by (xxxxxxxxxx)

We present an Immersogeometric Analysis (IMGA) approach for the simulation of compressible flows around complex geometries. In this method, compressible flow simulations are performed directly on various boundary representations (B-reps) of mechanical designs, circumventing the labor-intensive and time-consuming cleanup of complex geometric models. A new formulation for the weak imposition of essential boundary conditions in the context of non-body-fitted meshes is proposed. The formulation employs the non-symmetric Nitsche method, which yields good performance especially when the penalty parameters are difficult to estimate. We test the proposed immersogeometric formulation on benchmark problems for a wide range of Reynolds and Mach numbers, showing its robustness and accuracy. Finally, the methodology is applied to the simulation of a UH-60 Black Hawk helicopter in forward flight, illustrating the ability of the proposed approach to support the design of real-world engineering systems through high-fidelity aerodynamic analysis.

*Keywords:* Compressible flows; immersogeometric analysis; weakly enforced boundary conditions; non-symmetric Nitsche method; B-rep; rotorcraft.

AMS Subject Classification: 35Q30, 65M60, 76M10, 81T80

\*Corresponding author

## 1. Introduction

The aerodynamic simulation of rotorcraft involves moving boundaries and meshing around complex geometries. Such a numerical simulation has been traditionally carried out using moving-mesh methods, among which the Arbitrary Lagrangian–Eulerian (ALE) method<sup>1,2</sup> and the Space–Time (ST) method<sup>3,4</sup> are the most frequently used. These two classes of methods both use meshes that conform to and deform with the moving boundaries, and have been successfully applied to a wide range of applications such as wind turbines,<sup>5–16</sup> cardiovascular flows,<sup>17–24</sup> spacecraft parachutes,<sup>25–28</sup> and insect flights.<sup>29–32</sup> Comprehensive reviews on the fundamentals and applications of ALE and ST methods can be found in Refs. 33–35.

Despite the success in terms of accuracy and robustness, moving-mesh methods sometimes face challenges of mesh distortion when large translational, rotational, and other motions are present. However, in many cases, these may be addressed by remeshing<sup>36–38</sup>, special-purpose mesh moving techniques<sup>33,39–42</sup>, and sliding-interface formulations.<sup>11,12,43–45</sup> The bigger challenge, in our view, is that the generation of body-fitted mesh around geometrically complex objects often requires labor-intensive and time-consuming processes, such as defeating, geometry cleanup, and mesh manipulation.<sup>46–49</sup> An alternative that can alleviate these two challenges is the immersed method. Immersed methods approximate the solutions of Navier–Stokes equations on a non-body-fitted fluid domain discretization. The first immersed boundary method was developed by Peskin<sup>50</sup> to deal with computational fluid dynamics (CFD) problems with moving boundaries. Unlike its body-fitted counterpart where the fluid mesh needs to conform to a given surface description, an immersed method uses a non-body-fitted fluid mesh that is independent of the surface representation. This releases the strict mesh conforming constraint and greatly simplifies the mesh generation process, especially when geometrically complex objects are involved. Several instantiations of immersed methods for simulating flows around complex geometries can be found in Refs. 51–53.

Although immersed methods alleviate the meshing issues associated with the fluid domain, the conversion of a complex computer-aided design (CAD) model to an analysis mesh remains non-trivial, as the CAD surfaces often contain features that are difficult to discretize.<sup>46–49</sup> To overcome this challenge, immersogeometric analysis (IMGA)<sup>54–57</sup> was proposed to directly immerse the boundary representation (B-rep) of CAD models into the non-body-fitted background fluid discretization and perform simulations of incompressible flows.

In IMGA, Dirichlet boundary conditions are enforced weakly in the sense of Nitsche method.<sup>58</sup> Weak imposition of essential boundary conditions for incompressible flows in the context of body-fitted simulations was introduced by Bazilevs et al.<sup>59–61</sup> The weak boundary condition methodology forgoes setting the values of the velocity solution at boundary nodes in favor of imposing the same conditions by introducing an operator into the underlying weak form that involves integrals over the boundary surface. While in Refs. 59–61 it was assumed that the boundary over

which the integrals are taken is conforming to the fluid mesh, the IMGA no longer makes this assumption; the integrals are performed over a surface that is immersed into the volumetric fluid mesh in an arbitrary fashion, and the trial and test functions involved in the weak-boundary-condition operator are taken directly from the volumetric fluid mesh. Therefore, in the IMGA framework, CAD geometries can be used directly in CFD analysis as long as quadrature rules can be introduced on the CAD surfaces (e.g., Spline<sup>56</sup> or analytic<sup>57</sup> surfaces). These CAD models, in principle, do not require geometry cleanup, de-featuring, and mesh manipulation. The IMGA approach provides substantial flexibility for CFD, as one can directly work with the CAD models provided by the designers and therefore greatly shorten the design-to-analysis time.

In this paper, we aim to introduce these attractive features of IMGA in the important applications of compressible flows, such as the aerodynamic simulation of rotorcraft. To this end, we utilize a recently developed stabilized finite element formulation augmented with weakly enforced essential boundary conditions and a sliding-interface formulation for the simulation of compressible flows on moving domains.<sup>62</sup> In this work, the weak enforcement of no-slip boundary conditions, which is an essential component of IMGA, is extended to enforce Dirichlet boundary conditions for compressible flows on non-body-fitted meshes. In addition to facilitating the implementation of IMGA for compressible flows, this development also enables us to achieve good accuracy on relatively coarse boundary-layer meshes, by allowing the flow to slip on the solid surface in the case when the wall-normal mesh size is relatively large.<sup>10,61</sup> The weak-boundary-condition operator developed in Ref. 62 is based on a symmetric Nitsche approach, which provides very good accuracy and robustness. However, its performance is heavily dependent on the appropriate estimation of the penalty parameter, which is directly associated with the fluid mesh size at the boundary and the boundary-surface mesh size. In body-fitted methods, this estimation leads to the penalty parameter being a function of the wall-normal element size and is generally well defined. However, in immersed methods, the fluid elements at the boundary of the object are intersected by the object's surface; the estimation of the penalty parameter is particularly complicated in this situation and often does not yield the desired accuracy.<sup>63</sup>

Recently, a non-symmetric Nitsche approach was investigated in the context of non-body-fitted methods. The non-symmetric Nitsche method was first introduced by Baumann, Oden and coworkers<sup>64–66</sup> as part of a Discontinuous Galerkin (DG) methodology. Compared with the symmetric Nitsche method, the non-symmetric Nitsche method does not require additional stabilization and therefore does not depend on the penalty stabilization parameter. Schillinger et al.<sup>67</sup> found that the non-symmetric Nitsche method yields reduced  $L^2$  accuracy but significantly improves accuracy of the derivative quantities. It was shown in Refs. 68–70 that some stabilization in the non-symmetric Nitsche method is capable of improving the  $L^2$  accuracy. The penalty stabilization parameter in this case does not need to be a large value, as it is not the only source of stabilization as in the case of symmetric

Nitsche methods. Therefore, the penalty stabilization is less likely to overshadow the variational consistency that is largely responsible for the good performance of weak-boundary-condition enforcement. Dettmer et al.<sup>71</sup> studied a parameter-free non-symmetric Nitsche method in the context of CFD simulations of low-Reynolds-number incompressible flows using immersed methods. In the present paper, we advocate the use of the non-symmetric Nitsche method in the compressible-flow IMGA framework, and explore its performance in high-Reynolds-number turbulent flows around complex geometries.

We also devote a significant portion of this paper to the IMGA simulation of a full-scale UH-60 Black Hawk helicopter in forward flight. We illustrate how IMGA can be applied to efficiently simulate the aerodynamics of this military rotorcraft. The key idea is based on the observation that the most aerodynamically important surfaces are the rotor blades, which are not so geometrically complex as to present significant challenges to mesh generation. On the other hand, the fuselage and landing gear, which are much more challenging to mesh, are not as critical from the aerodynamics standpoint. Using this observation, we illustrate how a hybrid approach, which utilizes a body-fitted methodology for rotor blades and IMGA elsewhere, delivers high accuracy with greater efficiency relative to traditional methods.

The paper is organized as follows. In Section 2, we introduce the IMGA formulation of the Navier–Stokes equations of compressible flows with the non-symmetric Nitsche method. Section 3 gives implementation details that facilitate the use of IMGA with various types of immersed surface representations. In Section 4, we compute several 3D examples to verify and validate our proposed IMGA methodology for compressible flows. The examples span a wide range of Mach and Reynolds numbers to illustrate the robustness and accuracy of the formulation in a variety of flow regimes. In Section 5, we apply the IMGA methodology to the simulation of a UH-60 Black Hawk helicopter in forward flight, illustrating the ability of IMGA to support design of real-world engineering systems through high-fidelity aerodynamic analysis. In Section 6, we draw conclusions and motivate future work.

## 2. Numerical Methodology

### 2.1. *Compressible flows on moving domains*

In this section, we summarize the variational formulation for the Navier–Stokes equations of compressible flows on moving domains and its discretization in space and time. We also introduce the stabilization methodology and weak enforcement of boundary conditions for compressible flows. Note that the framework presented in this section equally holds for IMGA and body-fitted methods.

2.1.1. *Strong form*

The Navier–Stokes equations of compressible flows with a reduced form of the energy equation<sup>62</sup> are considered as the governing equations in this work. In what follows, Roman indices take on values  $\{1, \dots, d\}$ , where  $d = 2$  or  $3$  is the spatial dimension, and summation convention on repeated indices is applied. We introduce a conservation-variable vector,  $\mathbf{U}$ , and a pressure-primitive-variable vector,  $\mathbf{Y}$ , as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e \end{bmatrix}, \quad (2.1)$$

$$\mathbf{Y} = \begin{bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{bmatrix}, \quad (2.2)$$

where  $\rho$  is the density,  $u_i$  is the  $i$ th velocity component,  $e$  is the fluid internal energy density,  $p$  is the pressure, and  $T$  is the temperature. Pressure, density, and temperature are related through the ideal gas equation of state,  $p = \rho RT$ , where  $R$  is the ideal gas constant. Furthermore, we assume a calorically perfect gas and define the fluid internal energy density as  $e = c_v T$ , where  $c_v = R/(\gamma - 1)$  is the specific heat at constant volume and  $\gamma$  is the heat capacity ratio. Throughout the paper, we use  $(\cdot)_{,t}$  to denote a partial time derivative taken with respect to a fixed spatial coordinate in the referential domain, and we use  $(\cdot)_{,i}$  to denote the spatial gradient.

The convective arbitrary Lagrangian–Eulerian (ALE)<sup>1</sup> formulation of the balance of mass, linear momentum, and energy may be stated as

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i}^{\text{adv}} + \mathbf{F}^{\text{sp}} - \hat{u}_i \mathbf{U}_{,i} - \mathbf{F}_{i,i}^{\text{diff}} - \mathbf{S} = \mathbf{0}, \quad (2.3)$$

where  $\mathbf{F}_i^{\text{adv}}$  and  $\mathbf{F}_i^{\text{diff}}$  are the vectors of convective and diffusive fluxes, respectively, defined as

$$\mathbf{F}_i^{\text{adv}} = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \\ \rho u_i e \end{bmatrix} + \begin{bmatrix} 0 \\ p \delta_{1i} \\ p \delta_{2i} \\ p \delta_{3i} \\ 0 \end{bmatrix}, \quad (2.4)$$

$$\mathbf{F}_i^{\text{diff}} = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ -q_i \end{bmatrix}, \quad (2.5)$$

$\mathbf{F}^{\text{SP}}$  is the contribution of stress–power in the energy equation, defined as

$$\mathbf{F}^{\text{SP}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ pu_{i,i} - \tau_{ij}u_{j,i} \end{bmatrix}, \quad (2.6)$$

$\mathbf{S}$  is the source term, and  $\hat{u}_i$  is the  $i$ th component of the domain velocity  $\hat{\mathbf{u}}$ . In Eqs. (2.4)–(2.6),  $\delta_{ij}$  is the Kronecker delta, and  $\tau_{ij}$  and  $q_i$  are the viscous stress and heat flux, respectively, given by

$$\tau_{ij} = \lambda u_{k,k} \delta_{ij} + \mu (u_{i,j} + u_{j,i}), \quad (2.7)$$

$$q_i = -\kappa T_{,i}, \quad (2.8)$$

where  $\mu$  is the dynamic viscosity,  $\lambda = -2\mu/3$  is the second coefficient of viscosity, and  $\kappa$  is the thermal conductivity. We further split the convective flux into  $\mathbf{F}_i^{\text{adv}} = \mathbf{F}_i^{\text{adv}\setminus p} + \mathbf{F}_i^p$ , where  $\mathbf{F}_i^{\text{adv}\setminus p}$  and  $\mathbf{F}_i^p$  are the first and second terms, respectively, on the right-hand side of Eq. (2.4).

The balance laws in Eq. (2.3) may be further expressed in the following quasi-linear form involving the conservation variables  $\mathbf{U}$ :

$$\mathbf{U}_{,t} + \hat{\mathbf{A}}_i^{\text{ALE}} \mathbf{U}_{,i} - \left( \hat{\mathbf{K}}_{ij} \mathbf{U}_{,j} \right)_{,i} - \mathbf{S} = \mathbf{0}, \quad (2.9)$$

where  $\hat{\mathbf{A}}_i^{\text{ALE}} = \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^{\text{SP}} - \hat{u}_i \mathbf{I}$ ,  $\hat{\mathbf{A}}_i = \frac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{U}}$ ,  $\hat{\mathbf{A}}_i^{\text{SP}}$  is such that  $\hat{\mathbf{A}}_i^{\text{SP}} \mathbf{U}_{,i} = \mathbf{F}^{\text{SP}}$ ,  $\mathbf{I}$  is a  $5 \times 5$  identity matrix, and  $\hat{\mathbf{K}}_{ij}$  is such that  $\hat{\mathbf{K}}_{ij} \mathbf{U}_{,j} = \mathbf{F}_i^{\text{diff}}$ .

In the case of pressure-primitive variables  $\mathbf{Y}$ , Eq. (2.3) becomes

$$\mathbf{A}_0 \mathbf{Y}_{,t} + \mathbf{A}_i^{\text{ALE}} \mathbf{Y}_{,i} - (\mathbf{K}_{ij} \mathbf{Y}_{,j})_{,i} - \mathbf{S} = \mathbf{0}, \quad (2.10)$$

where  $\mathbf{A}_i^{\text{ALE}} = \mathbf{A}_i + \mathbf{A}_i^{\text{SP}} - \hat{u}_i \mathbf{A}_0$ ,  $\mathbf{A}_0 = \frac{\partial \mathbf{U}}{\partial \mathbf{Y}}$ ,  $\mathbf{A}_i = \frac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{Y}} = \frac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{Y}} = \hat{\mathbf{A}}_i \mathbf{A}_0$ ,  $\mathbf{A}_i^{\text{SP}}$  is such that  $\mathbf{A}_i^{\text{SP}} \mathbf{Y}_{,i} = \mathbf{F}^{\text{SP}}$ , and  $\mathbf{K}_{ij}$  is such that  $\mathbf{K}_{ij} \mathbf{Y}_{,j} = \mathbf{F}_i^{\text{diff}}$ . Based on the splitting of  $\mathbf{F}_i^{\text{adv}}$  into  $\mathbf{F}_i^{\text{adv}\setminus p}$  and  $\mathbf{F}_i^p$ , we can further split  $\mathbf{A}_i$  as  $\mathbf{A}_i = \mathbf{A}_i^{\text{adv}\setminus p} + \mathbf{A}_i^p$  to separate the pressure term from the convective flux. Detailed expressions for the matrices appearing in the quasi-linear forms can be found in Appendix A of Ref. 62.

Finally, the residual for the quasi-linear ALE form of the compressible-flow equations may be defined as

$$\mathbf{Res} = \mathbf{A}_0 \mathbf{Y}_{,t} + \mathbf{A}_i^{\text{ALE}} \mathbf{Y}_{,i} - (\mathbf{K}_{ij} \mathbf{Y}_{,j})_{,i} - \mathbf{S}. \quad (2.11)$$

**Remark 2.1.** Note that the choice of conservation or pressure-primitive variables does not change the balance laws. In this work, pressure-primitive variables are used because, unlike for conservation variables, the incompressible limit of the compressible-flow equations is well defined for the pressure-primitive variables.<sup>72</sup> Pressure-primitive variables are also convenient for setting boundary conditions and implementing fluid–structure interaction (FSI) coupling.<sup>33</sup>

## 2.1.2. Weak form

Let  $\Omega \in \mathbb{R}^d$  denote the spatial domain and  $\Gamma$  be its boundary. Consider a collection of disjoint elements  $\{\Omega^e\}$  such that  $\Omega \subset \cup_e \overline{\Omega^e}$ . Let  $\mathcal{S}^h$  be the discrete trial-function space for the pressure-primitive variables and  $\mathcal{V}^h$  be the discrete test-function space for the compressible-flow equation system, both defined over  $\{\Omega^e\}$ . The strong form (2.10) may be recast in a weak form and posed over these discrete spaces to produce the following semi-discrete stabilized variational problem: Find  $\mathbf{Y}^h \in \mathcal{S}^h$  such that for all  $\mathbf{W}^h \in \mathcal{V}^h$ ,

$$B(\mathbf{W}^h, \mathbf{Y}^h) - F(\mathbf{W}^h) = 0, \quad (2.12)$$

where

$$\begin{aligned} B(\mathbf{W}, \mathbf{Y}) &= \int_{\Omega} \mathbf{W} \cdot \left( \mathbf{A}_0 \mathbf{Y}_{,t} + \left( \mathbf{A}_i^{\text{adv}\setminus\text{p}} + \mathbf{A}_i^{\text{sp}} - \hat{u}_i \mathbf{A}_0 \right) \mathbf{Y}_{,i} \right) d\Omega \\ &\quad - \int_{\Omega} \mathbf{W}_{,i} \cdot \left( \mathbf{A}_i^{\text{p}} \mathbf{Y} - \mathbf{K}_{ij} \mathbf{Y}_{,j} \right) d\Omega \\ &\quad + \sum_e \int_{\Omega^e \cap \Omega} \left( \left( \mathbf{A}_i^{\text{ALE}} \right)^{\text{T}} \mathbf{W}_{,i} \right) \cdot \left( \mathbf{A}_0^{-1} \hat{\boldsymbol{\tau}}_{\text{SUPG}} \right) \mathbf{Res} d\Omega \\ &\quad + \sum_e \int_{\Omega^e \cap \Omega} \mathbf{W}_{,i} \cdot \left( \hat{\mathbf{K}}_{\text{DC}} \mathbf{A}_0 \right) \mathbf{Y}_{,i} d\Omega, \end{aligned} \quad (2.13)$$

and

$$F(\mathbf{W}) = \int_{\Omega} \mathbf{W} \cdot \mathbf{S} d\Omega + \int_{\Gamma_{\mathbf{H}}} \mathbf{W} \cdot \mathbf{H} d\Gamma. \quad (2.14)$$

In the above,  $\mathbf{A}_0^{-1} = \frac{\partial \mathbf{Y}}{\partial \mathbf{U}}$ ,  $\mathbf{H}$  contains the fluid traction (in the momentum-equation slots) and heat flux (in the energy-equation slot) boundary conditions, and  $\Gamma_{\mathbf{H}}$  is the subset of  $\Gamma$  where  $\mathbf{H}$  is specified.

On the right-hand side of Eq. (2.13), the first two terms correspond to the Galerkin form of the Navier–Stokes equations of compressible flows, the fourth term is the discontinuity-capturing (DC) operator,<sup>73–85</sup> and the third term is the streamline upwind/Petrov–Galerkin (SUPG) stabilization, which was first introduced to compressible-flow applications in Ref. 86 and later refined in Refs. 72, 87–96. The stabilization matrix  $\hat{\boldsymbol{\tau}}_{\text{SUPG}}$  is defined as

$$\hat{\boldsymbol{\tau}}_{\text{SUPG}} = \left( \frac{4}{\Delta t^2} \mathbf{I} + G_{ij} \hat{\mathbf{A}}_i^{\text{ALE}} \hat{\mathbf{A}}_j^{\text{ALE}} + C_I G_{ij} G_{kl} \hat{\mathbf{K}}_{ik} \hat{\mathbf{K}}_{lj} \right)^{-\frac{1}{2}}, \quad (2.15)$$

where  $\Delta t$  is the time step size,  $C_I$  is a positive constant derived from an appropriate element-wise inverse estimate,<sup>97</sup> and  $G_{ij}$  contains the information about the element size derived from the element geometric mapping from the parent to physical coordinates  $\mathbf{x}(\xi)$ :

$$G_{ij} = \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}. \quad (2.16)$$

Equation (2.15) requires computation of the square-root-inverse of a  $5 \times 5$  matrix for 3D problems. For this purpose, the modified Denman–Beavers algorithm proposed by Xu et al.<sup>62</sup> is employed.

Lastly,  $\hat{\mathbf{K}}_{\text{DC}}$  in Eq. (2.13) is the DC-operator diffusivity matrix defined using a diagonal form:

$$\hat{\mathbf{K}}_{\text{DC}} = \text{diag}(\hat{\kappa}_{\text{C}}, \hat{\kappa}_{\text{M}}, \hat{\kappa}_{\text{M}}, \hat{\kappa}_{\text{M}}, \hat{\kappa}_{\text{E}}), \quad (2.17)$$

where the diagonal entries are given by

$$\hat{\kappa}_{\text{C}} = C_{\text{C}} |\text{Res}_1| (G_{ij} U_{1,i} U_{1,j})^{-\frac{1}{2}}, \quad (2.18)$$

$$\hat{\kappa}_{\text{M}} = C_{\text{M}} \left( \sum_{\alpha=2}^{d+1} \text{Res}_{\alpha}^2 \right)^{\frac{1}{2}} \left( \sum_{\alpha=2}^{d+1} G_{ij} U_{\alpha,i} U_{\alpha,j} \right)^{-\frac{1}{2}}, \quad (2.19)$$

$$\hat{\kappa}_{\text{E}} = C_{\text{E}} |\text{Res}_{d+2}| (G_{ij} U_{d+2,i} U_{d+2,j})^{-\frac{1}{2}}. \quad (2.20)$$

In the above,  $C_{\text{C}}$ ,  $C_{\text{M}}$ , and  $C_{\text{E}}$  are the  $\mathcal{O}(1)$  positive constants corresponding to the continuity, momentum, and energy equations, respectively, and  $U_{(\cdot)}$  are the entries of the conservation-variable vector  $\mathbf{U}$ .

**Remark 2.2.** The SUPG stabilization and DC-operator diffusivity matrices remain a subject of active research to date.<sup>78,79,94</sup> In the present work, we adopt the philosophy of designing  $\hat{\tau}_{\text{SUPG}}$  and  $\hat{\mathbf{K}}_{\text{DC}}$  for the conservation variables and transforming to the pressure-primitive-variable formulation.<sup>62,93</sup> The SUPG and DC formulations presented here are verified and validated extensively in Ref. 62 for body-fitted simulations for a wide range of Reynolds and Mach numbers.

## 2.2. Weak-boundary-condition operator

We assume that essential boundary conditions of the velocity and temperature fields are enforced on  $\Gamma_{\mathbf{u}}^{\text{D}} \in \Gamma$  and  $\Gamma_T^{\text{D}} \in \Gamma$ , respectively, where  $\Gamma$  is the fluid domain boundary. Note that  $\Gamma$  does not necessarily coincide with the boundary of element  $\{\Omega^e\}$ . Let  $\mathbf{W} = [q \ \mathbf{w} \ w_{\theta}]^{\text{T}}$  be the vector of test functions with  $\mathbf{w} = [w_1 \ w_2 \ w_3]^{\text{T}}$  being the test functions for the linear-momentum balance equations. The weak-boundary-condition operator for compressible flows is given by

$$\begin{aligned} B_{\text{WBC}}(\mathbf{W}, \mathbf{Y}) = & - \int_{\Gamma_{\mathbf{u}}^{\text{D}}} \mathbf{w} \cdot (-p\mathbf{n}) \, d\Gamma - \int_{\Gamma_{\mathbf{u}}^{\text{D}}} \rho q \mathbf{n} \cdot (\mathbf{u} - \mathbf{g}) \, d\Gamma \\ & - \int_{\Gamma_{\mathbf{u}}^{\text{D}}} \mathbf{w} \cdot ((\lambda \nabla \cdot \mathbf{u}) \mathbf{I} + \mu (\nabla \mathbf{u} + \nabla^{\text{T}} \mathbf{u})) \mathbf{n} \, d\Gamma \\ & - \beta \int_{\Gamma_{\mathbf{u}}^{\text{D}}} ((\lambda \nabla \cdot \mathbf{w}) \mathbf{I} + \mu (\nabla \mathbf{w} + \nabla^{\text{T}} \mathbf{w})) \mathbf{n} \cdot (\mathbf{u} - \mathbf{g}) \, d\Gamma \\ & - \int_{\Gamma_{\mathbf{u}}^{\text{D},-}} \mathbf{w} \cdot \rho ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n}) (\mathbf{u} - \mathbf{g}) \, d\Gamma \\ & + \int_{\Gamma_{\mathbf{u}}^{\text{D}}} \mathbf{w} \cdot \tau_{\mu}^{\text{B}} (\mathbf{u} - \mathbf{g}) \, d\Gamma + \int_{\Gamma_{\mathbf{u}}^{\text{D}}} (\mathbf{w} \cdot \mathbf{n}) \tau_{\lambda}^{\text{B}} (\mathbf{u} - \mathbf{g}) \cdot \mathbf{n} \, d\Gamma \end{aligned}$$

$$\begin{aligned}
 & - \int_{\Gamma_T^D} w_\theta \kappa \nabla T \cdot \mathbf{n} \, d\Gamma - \beta \int_{\Gamma_T^D} \kappa \nabla w_\theta \cdot \mathbf{n} (T - T^B) \, d\Gamma \\
 & - \int_{\Gamma_T^{D,-}} w_\theta \rho c_v ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n}) (T - T^B) \, d\Gamma \\
 & + \int_{\Gamma_T^D} w_\theta \tau_\kappa^B (T - T^B) \, d\Gamma, \tag{2.21}
 \end{aligned}$$

where  $\mathbf{g}$  is the prescribed velocity on  $\Gamma_u^D$ ,  $T^B$  is the prescribed temperature on  $\Gamma_T^D$ , and the superscript “-” denotes the “inflow” part of  $\Gamma$  where  $(\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n} < 0$ . The first seven terms on the right-hand side of Eq. (2.21) correspond to the weak enforcement of the velocity boundary conditions, while the last four terms ensure weak enforcement of temperature boundary conditions. The choice of parameter  $\beta$  in Eq. (2.21) between 1 and  $-1$  decides whether the formulation is symmetric or non-symmetric Nitsche method, respectively. Note that the formulation given by Eq. (2.21) may be employed, as written, for body-fitted and non-body-fitted methods.

The penalty terms associated with  $\tau_\mu^B$ ,  $\tau_\lambda^B$ , and  $\tau_\kappa^B$  in the weak-boundary-condition operator help to better satisfy the Dirichlet boundary conditions and offer additional stabilization. The estimation of the penalty parameters can be based on a generalized eigenvalue problem.<sup>63</sup> In the case of symmetric Nitsche method using body-fitted discretization,<sup>59,60,62</sup> the solution of the eigenvalue problem leads to the form of  $\tau_\mu = \frac{C_I^B \mu}{h_n}$ ,  $\tau_\lambda = \frac{C_I^B |\lambda|}{h_n}$ , and  $\tau_\kappa = \frac{C_I^B \kappa}{h_n}$ , where  $h_n$  is the element size in the wall-normal direction and  $C_I^B$  is a positive constant.  $C_I^B$  needs to be sufficiently large for the overall stability of the formulation; however,  $C_I^B$  should not be chosen too large, since too large of a penalty can overshadow the variational consistency responsible for the good performance of the method.

The symmetric Nitsche method provides very good accuracy and robustness, but its performance in the context of immersed methods is heavily dependent on the proper estimation of penalty parameters. In immersed methods, the reliable solution of the eigenvalue problem becomes very challenging, as the arbitrarily intersected elements deteriorate the conditioning of the problem. Special techniques are needed,<sup>98,99</sup> which can be complex and delicate from the algorithmic viewpoint. In addition, careful estimation of the penalty parameters in the immersed methods does not necessarily lead to good accuracy and robustness of the computational results.

With these observations, the non-symmetric Nitsche method becomes attractive for immersed approaches. The penalty stabilization parameter in this case does not need to be above a specific lower bound, and thus does not require a very careful estimation. For IMGA using the non-symmetric Nitsche method, we scale the penalty parameters as  $\tau_\mu^B = \tau_\lambda^B = \frac{C_I^B \rho h_n^e}{\Delta t}$ , and  $\tau_\kappa^B = c_v \tau_\mu^B$ , inspired by Wu et al.<sup>100</sup> Note that in this case, the  $h_n^e$  is calculated from the *full element*, which

greatly simplifies the evaluation in the presence of intersected elements. Taking full advantage of either method, in this work, the symmetric Nitsche method is used in the cases of body-fitted simulations, while the non-symmetric Nitsche method is employed in the cases of non-body-fitted simulations.  $C_I^B$  is set to 4 for all the cases.

**Remark 2.3.** One can find the correspondence between the weak-boundary-condition operator and a variety of DG methods. In Eq. (2.21),  $\beta = 1$  corresponds to the symmetric interior penalty Galerkin (SIPG) method.<sup>101</sup> The choice of  $\beta = -1$  corresponds to the non-symmetric interior penalty Galerkin (NIPG) method.<sup>102</sup> The penalty-free non-symmetric Nitsche method ( $\beta = -1$ ,  $\tau_\mu^B = \tau_\lambda^B = \tau_\kappa^B = 0$ ) corresponds to the method of Baumann and Oden.<sup>65</sup>

### 2.3. Time integration and solution strategies

Including all the numerical constituents into a single framework, the final semi-discrete form of the Navier–Stokes equations of compressible flows may be stated as: Find  $\mathbf{Y}^h \in \mathcal{S}^h$ , such that  $\forall \mathbf{W}^h \in \mathcal{V}^h$ ,

$$B(\mathbf{W}^h, \mathbf{Y}^h) - F(\mathbf{W}^h) + B_{\text{WBC}}(\mathbf{W}^h, \mathbf{Y}^h) = 0. \quad (2.22)$$

To integrate the semi-discrete compressible-flow equations in time we employ the generalized- $\alpha$  method.<sup>103–105</sup> Generalized- $\alpha$  is an implicit, unconditionally stable, second-order method with control over high-frequency dissipation. At each time step, the solution of the nonlinear algebraic-equation system is performed using the Newton–Raphson technique. At each Newton–Raphson iteration the linear system is solved iteratively using a block-diagonally preconditioned GMRES technique.<sup>106</sup>

## 3. IMGA Implementation details

### 3.1. Adaptive quadrature

The IMGA method introduces elements that are intersected by the geometric boundary, which creates complex, discontinuous integration domains in intersected elements. To ensure geometrically accurate evaluation of volume integrals in intersected elements, we use a sub-cell-based adaptive quadrature scheme.<sup>55,107</sup> The basic concept is to increase the number of quadrature points around immersed geometric boundaries so that arbitrary integration domains resulting from the intersecting boundary can be taken into account accurately. This is achieved by recursively splitting intersected cells into sub-elements. At each level, only those sub-elements intersected by the boundary are further split. For clarity, we illustrate the quadrature scheme based on adaptive sub-cells for triangles in 2D in Fig. 1. We emphasize that splitting is performed for quadrature only, and does not affect the basis functions, which are still defined on the original (unsplit) elements.

The implementation of sub-division-based adaptive quadrature rules requires two types of point membership classifications: the inside/outside membership of

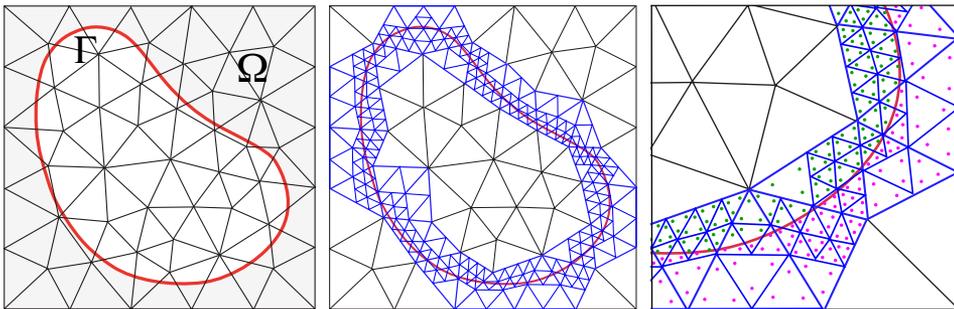


Fig. 1: Illustration the quadrature scheme based on adaptive sub-cells (marked in blue) for triangles in 2D. The fluid domain  $\Omega$  is marked in grey and immersed boundary  $\Gamma$  is marked in red. Quadrature points in the fluid domain (marked in pink) are used in the numerical integration, while quadrature points outside (marked in green) are discarded.

mesh nodes to determine whether an element or sub-element is intersected, and of the quadrature points to determine whether they should be kept or discarded. A frequently used approach is the ray-tracing technology, which has been investigated very thoroughly by the computer graphics community.<sup>108</sup> We employ ray-tracing-based query accelerated by an octree data structure implemented in Ref. 55 for the tessellated immersed surfaces. We also employ the GPU-accelerated point membership classification<sup>56,57,109</sup> developed for the cases where the immersed geometries are B-reps constructed by NURBS or analytic surfaces.

### 3.2. Surface integration

The weak boundary-condition operator only requires us to compute the surface integrals that involve traces of volumetric FE functions defined on the background elements. Therefore the IMGGA framework is able to immerse arbitrary types of boundary-representations (B-reps) into the background fluid mesh, as long as quadrature rules can be specified on the immersed surface representation. Those quadrature points are then needed to be located in the parametric space of the background elements, which requires us to invert the mapping from the finite element parameter space to the physical space. To accelerate this process, we first sort the background elements into an octree hierarchy of bounding boxes. For each surface quadrature point, we then recursively search for its containing finite element in a sub-tree containing the point. With this approach, we need only invert parameter-to-physical-space mappings of the subset of finite elements whose bounding boxes intersect the unique leaf of the octree containing the surface quadrature point.

In B-rep, the boundary of a CAD model is represented using a set of faces. While NURBS is the de-facto surface representation used in B-rep, analytic surfaces are also frequently used for synthetic objects where many flat features with rounded

corners exist. To process those surfaces for surface integration in IMGA, various approaches can be adopted. In traditional finite element analysis, those surfaces are first tessellated into, for example, triangles, and then standard Gauss quadrature rule can be established naturally, see Ref. 110 for example. Note that some surface representation format, such as STL, in which the geometry is represented by tessellated primitives, can be directly used in IMGA for the surface integration as a finite element mesh. On the other hand, the development of IMGA offers us an alternative way to directly evaluate the surface integrals on B-rep surfaces. Hsu, Wang and co-workers<sup>56,57</sup> developed integration rules directly associated with NURBS and analytic surface-based B-reps. This allows for a direct use of B-rep information in the IMGA framework, thus avoiding the labor-intensive geometry cleanup process that is often needed for a water-tight surface tessellation.

### 3.3. *Load balancing*

The parallelization strategy proposed by Hsu et al.<sup>111</sup> is employed for the IMGA simulations presented in this paper. In this strategy the problem mesh is partitioned into subdomains with a nearly equal number of elements, and the partitions are assigned to different processing cores. However, this strategy can create a very unbalanced distribution of quadrature points in IMGA, since quadrature points aggregate in the cut elements due to the use of adaptive quadrature (see Section 4.1 for an example). In the case of compressible flows, the overall cost of the solver is dominated by element matrix formation, and is directly proportional to the total number of element quadrature points. As a result, balancing of the total number of quadrature points on each partition will likely provide a better mesh partitioning strategy for IMGA.

The mesh/graph partitioning package METIS<sup>112</sup> allows users to weight the elements differently, and then decompose the mesh to partitions that have nearly equal summation of user-defined weights. We utilize this feature and propose to weight an element by the actual number of quadrature points it contains. In the IMGA discretization using tetrahedral elements with adaptive quadrature, the elements can be divided into three types: elements inside the fluid domain with a weight of 4, elements inside the immersed B-rep with a weight of 0, and cut elements with weights ranging from 0 to  $4 \times 12^l$ , where  $l$  is the adaptive quadrature refinement level. Benefits of this mesh partitioning approach are illustrated on an example in Section 4.1.

## 4. Benchmark Numerical Examples

### 4.1. *Laminar flow*

In this section, we perform the simulation of laminar flow around a torpedo-shaped body modeled by trimmed analytic surfaces,<sup>57</sup> in both the subsonic and supersonic regimes. The geometry of the torpedo shape and the surface quadrature rule stem

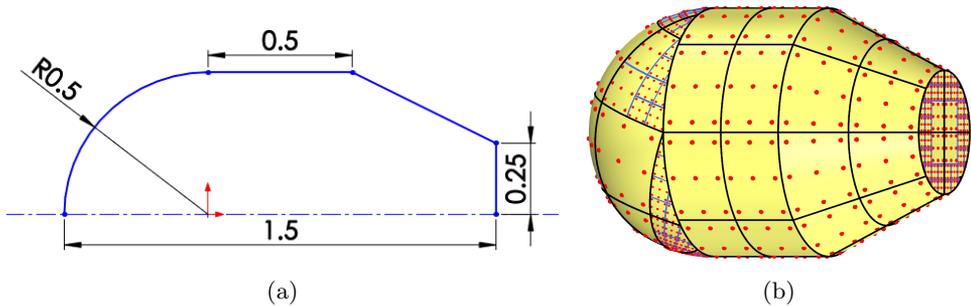


Fig. 2: (a) Dimensions of the torpedo-shaped body; (b) B-rep surface and adaptive surface quadrature points of the torpedo-shaped body.

from a trimmed analytic surface definition (see Figure 2). Uniform flow conditions are imposed on the inlet boundary (see Figure 2a). For the subsonic case ( $M = 0.8$ ), the inflow quantities are set to  $p = 1.1161$ ,  $\|\mathbf{u}\| = 1.0$ , and  $T = 3.8713 \times 10^{-3}$ . In this case, the dynamic viscosity is set to a constant value of 0.01. For the supersonic case ( $M = 2.0$ ), the inflow quantities are set to  $p = 0.1786$ ,  $\|\mathbf{u}\| = 1.0$ , and  $T = 6.1941 \times 10^{-4}$ , and the dynamic viscosity is determined from Sutherland's law:

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + S}, \quad (4.1)$$

where  $S = 1.406 \times 10^{-4}$  and  $C_1 = 0.906$ . Both the subsonic and supersonic cases result in the Reynolds number 100 based on the diameter of the body's spherical cap. On the B-rep surface, the velocity is set to zero and the temperature is set as the stagnation temperature given by

$$T^B = \left(1 + \frac{\gamma - 1}{2} M^2\right) T. \quad (4.2)$$

Note that all the above quantities are dimensionless.

We perform a mesh refinement study to assess the performance of the compressible-flow IMGA formulation. Simulations of the supersonic flow are carried out on two meshes, IM1 and its  $h$ -refined version IM2 (see Ref. 57 for the notation and statistics of these meshes). To illustrate the mesh design, we plot a planar cut through the center of the coarsest mesh IM0 in Figure 3b. The flows are simulated using a time step size of 0.005 until a steady state is reached. The pressure coefficient distribution along the upper crown line of the torpedo-shaped body as a function of the streamwise coordinate is plotted in Figure 4. The results demonstrate that IM1 and IM2 produce almost identical solutions. Note that a two-level recursive adaptive quadrature rule is employed in order to faithfully capture the immersed geometry and produce an essentially converged solution.

As a reference, body-fitted computations of both the subsonic and supersonic cases are carried out on BM2 (see Table 1 in Ref. 57), using the symmetric ver-

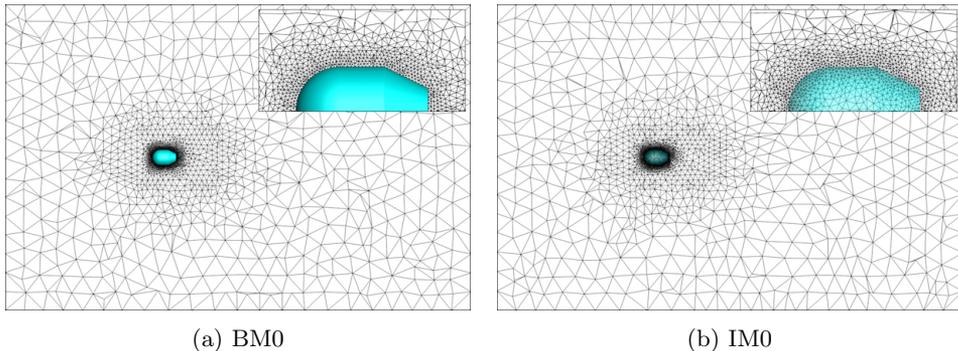


Fig. 3: BM0 and IM0 meshes with a zoom on the region near the torpedo-shaped body. Translucency is applied to the IM0 mesh to show how the immersed surface cuts through the background elements.

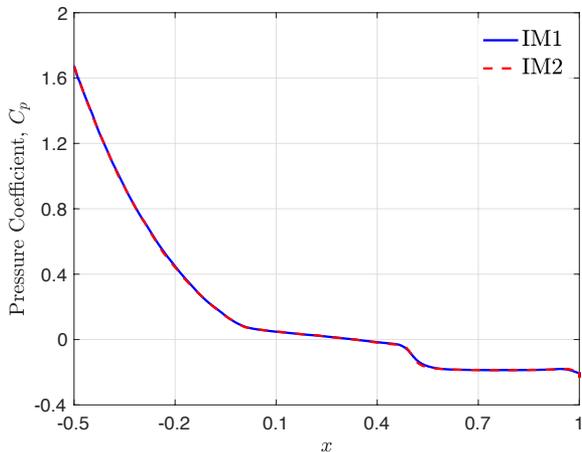


Fig. 4: Pressure coefficient along the upper crown line of the torpedo-shaped body as a function of the streamwise coordinate for the supersonic flow case using IM1 and IM2.

sion of the weakly enforced boundary-condition formulation. Figure 3 compares the mesh design of BM0 and IM0, with a zoom-in view to show the treatment of the boundaries. We choose to visualize these coarse meshes for clarity. We note that IM2 and BM2 have a similar mesh design in terms of the element size around the torpedo-shaped body and refinement patterns in the rest of the fluid domain. Figure 5 shows Mach number contours for both the subsonic and supersonic cases computed on IM2. A distinguishing feature of the supersonic case is the detached bow shock ahead of the leading edge of the torpedo shape. We show the IMGA

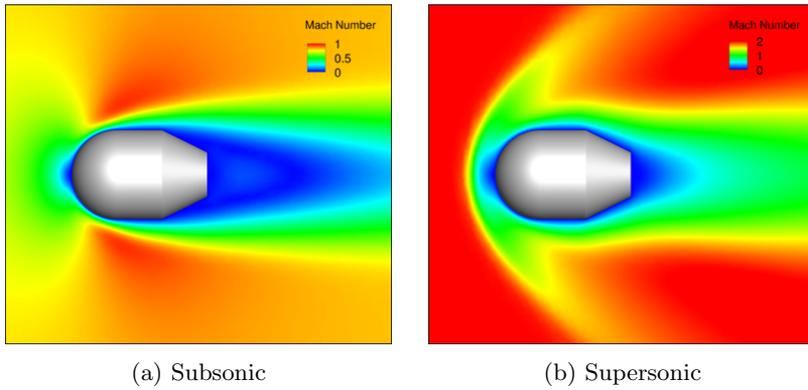


Fig. 5: Mach number contours for the subsonic and supersonic flows around a torpedo-shaped body simulated on IM2.

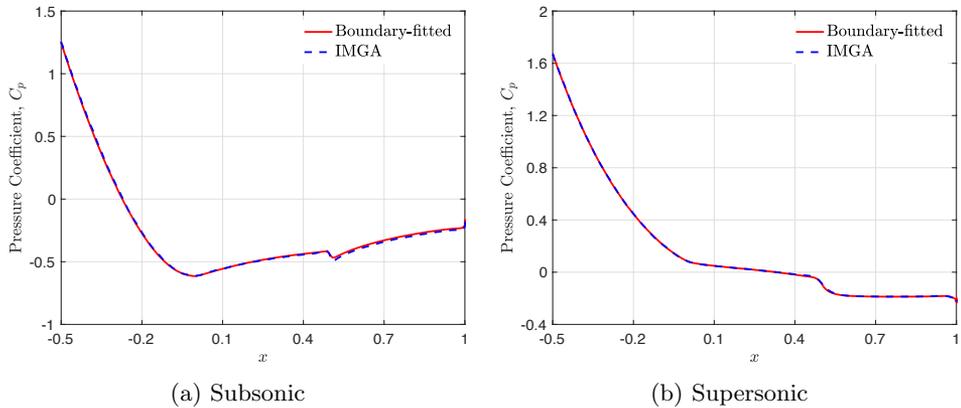


Fig. 6: Pressure coefficient along the upper crown line of the torpedo-shaped body as a function of the streamwise coordinate for the subsonic and supersonic flow cases computed on IM2. Body-fitted mesh results computed using BM2 are plotted for comparison.

results for pressure distributions from IM2, with the body-fitted results obtained from BM2 as a reference, in Figure 6. The results demonstrate excellent agreement between the IMGA and body-fitted simulations.

To show the effectiveness of load balancing strategy proposed in Section 3.3, we take the mesh IM0 (see Table 2 in Ref. 57) and perform the mesh decomposition. The mesh is first decomposed into 12 partitions using the element-based strategy in Ref. 111. A very unbalanced quadrature point distribution is observed (see blue bars in Figure 7). With this partitions, the processing core responsible for partition 11

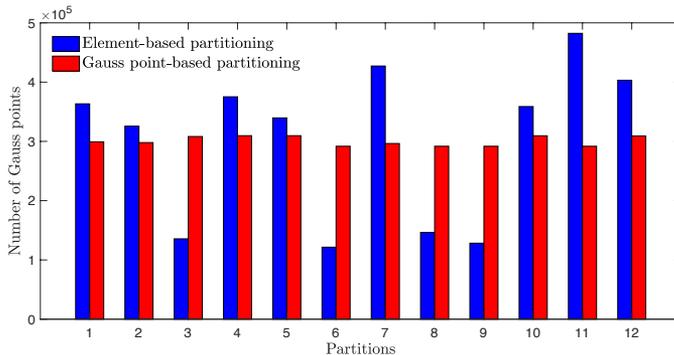


Fig. 7: Number of quadrature points on each mesh partition.

will determine the overall simulation time, thus creating a performance bottleneck. Using the proposed quadrature-point-based load balancing strategy, we are able to produce very similar numbers of quadrature points on each partition, as shown using the red bars in Figure 7.

#### 4.2. Turbulent flow around a 3D circular cylinder at $Re = 200,000$

A benchmark problem of turbulent compressible flow around a 3D cylinder is simulated in this section. Experimental measurements<sup>113</sup> and large eddy simulation<sup>114</sup> of the flow field at  $Re = 200,000$  and  $M = 0.75$  are available for validation. Such an in-flow condition features a transonic flow with a turbulent boundary layer separation. We simulate the flow using the IMGGA formulation. A body-fitted mesh simulation using the symmetric version of the weak boundary-condition enforcement is also carried out for comparison purposes.

The cylinder has a non-dimensional diameter of  $d = 1$ . The length of the cylinder is set to  $4d$ , as suggested by Xu et al.<sup>114</sup>, such that the solution at the middle section of the cylinder is not influenced by the effect of the lateral boundaries. See Figure 8 for the computational setup. In the plane of the cylinder cross-section, mesh size of 0.005 is used in the vicinity of the cylinder surface. Two levels of refinement boxes with mesh size of 0.05 and 0.2, respectively, are specified to capture the wake behind the cylinder and to allow a smooth mesh-size transition. The rest of the domain is filled with tetrahedral elements with a size of 0.5. Figure 9a shows a mesh cut along the cylinder axis. In the direction along the cylinder axis the element size is set to 0.01 (see Figure 9b). The body-fitted mesh is designed in the same way as described above. The IMGGA and the body-fitted mesh contain 11,327,392 and 9,923,711 elements, respectively. The cylinder surface is modeled as a quadratic NURBS surface, with the physical length of knot spans in circumferential and axial directions to be 0.005 and 0.01, respectively, to match the background element size. A  $3 \times 3$ -point Gauss quadrature rule is used in the NURBS surface elements.

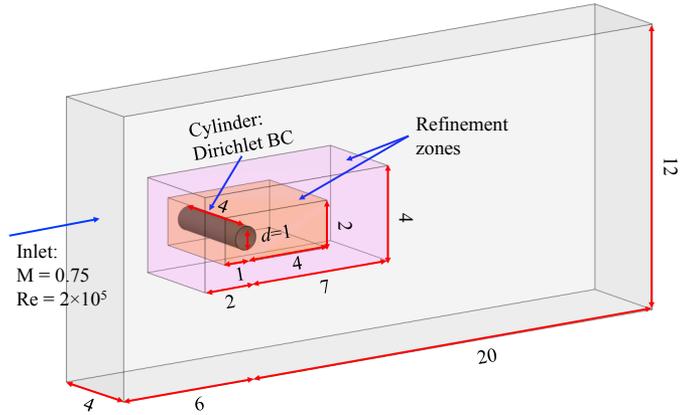


Fig. 8: Problem setup.

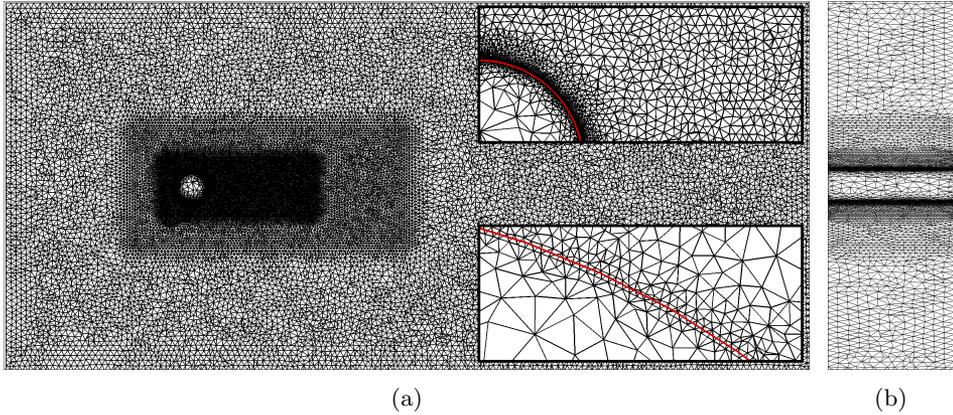


Fig. 9: Problem mesh: (a) Cross-section along the cylinder axis with zoom near the cylinder surface; (b) Vertical plane containing the cylinder axis.

Uniform flow conditions are set on the inlet boundary of the problem domain shown in Figure 9. The non-dimensional inflow quantities are set to  $\|\mathbf{u}\| = 1.0$ ,  $p = 1.2698$ , and  $T = 4.4047 \times 10^{-3}$ . The outlet boundary is set to have the same total traction as the inlet. The dynamic viscosity of the fluid is chosen as  $5 \times 10^{-6}$ , which yields the inflow Reynolds number of  $2 \times 10^5$  based on the cylinder diameter. No penetration, zero traction, and zero heat flux boundary conditions are set on the lateral boundaries of the problem domain. The no-slip velocity and stagnation temperature (calculated using Eq. (4.2)) boundary conditions on the cylinder surface are imposed weakly. A time step size of  $5 \times 10^{-3}$  is used in the simulations.

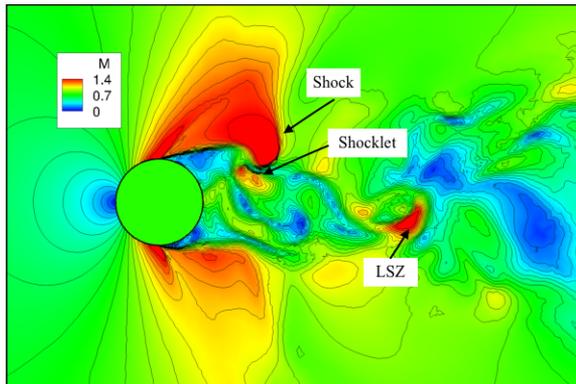


Fig. 10: Instantaneous Mach number contours on a cross-section along the cylinder axis.

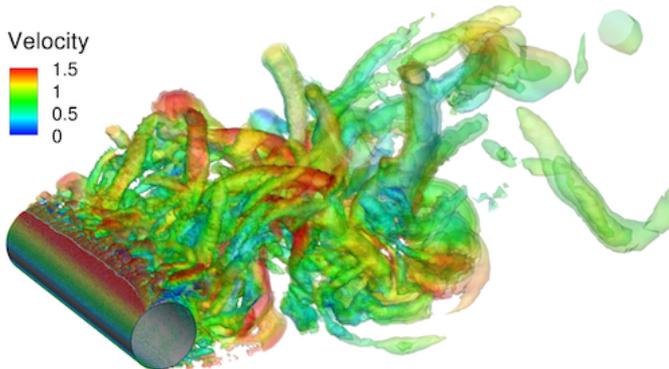


Fig. 11: Vorticity isosurfaces colored by velocity magnitude.

To investigate the flow structures, we first show the Mach number contours in a plane perpendicular the cylinder axis. Typical flow phenomena in the transonic flow regime, such as shocks, localized supersonic zones (LSZ), and shocklets (small shocks), are observed in the solution shown in Figure 10. Note that an LSZ is not necessarily associated with a shocklet. For example, an LSZ denoted in Figure 10 does not result in a shocklet. Instantaneous vortical structures characterized by the region with a negative  $\lambda_2$  parameter<sup>115,116</sup> are shown in Figure 11 to illustrate the flow complexity in 3D.

The mean pressure coefficient, defined as  $C_p = \frac{2(p-p_\infty)}{\rho_\infty \|\mathbf{u}\|_\infty^2}$ , is plotted as a function of the azimuthal angle in Figure 12. The subscript  $\infty$  indicates values taken from the inflow. The mean pressure distribution is obtained by time averaging of the pressure field after a statistically-stationary state of the flow is reached. The experimental data,<sup>113</sup> LES results,<sup>114</sup> and our own body-fitted simulation results are

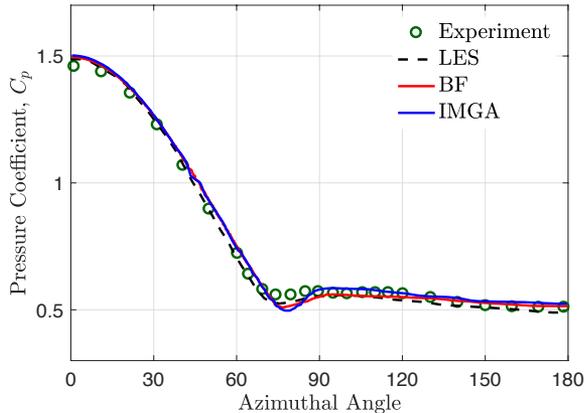


Fig. 12: Pressure coefficient on the cylinder surface as a function of the azimuthal angle. Time-averaged pressure data taken at the central cylinder cross-section. “BF” denotes the results from our own body-fitted simulation.

also plotted for comparison in Figure 12. Excellent agreement between the IMGA and reference results is observed. This example demonstrates that IMGA based on the non-symmetric Nitsche’s method is capable of delivering good pressure-load accuracy for higher Reynolds number flows, and presents a good computational technology for applications such as presented in the next Section.

## 5. UH-60 Helicopter in Forward Flight

In this section, we make use of the compressible-flow IMGA formulation to simulate a full-scale UH-60 Black Hawk helicopter in forward flight. A body-fitted, moving-domain approach is employed in the subdomain containing the aerodynamically important lifting surfaces, i.e., rotor blades. IMGA is used for the remainder of the problem domain that contains the geometrically complex fuselage with appendages, such as landing gear, which are typically hard to mesh. A sliding interface methodology is used to couple the two subdomains.<sup>11,12,43–45</sup> The proposed approach combines superior accuracy of body-fitted methods for aerodynamically important surfaces with relative simplicity of IMGA for analysis-suitable model generation, and is thus attractive for the present and similar applications. We carry out the simulations focusing on rotor and fuselage loads for different flight conditions in an effort to support the design improvements for rotorcraft.

### 5.1. Main rotor geometry

Two types of airfoil profiles are used in the main rotor of the UH-60 Black Hawk helicopter: SC1095 and SC1094R8. Data for the airfoil profiles at different cross-sectional locations in the span-wise direction and the corresponding aerodynamic



Fig. 13: The blade surface and cross-sections used for lofting.

Table 1: Blade cross-section data.

Number	Section characteristic	Radial location (in)	Chord (in)	Quarter chord location (in)	Twist ( $^{\circ}$ )
1	Root cutout	42.000	20.760	0.000	9.675
2	SC1095	62.000	20.760	0.000	9.675
3	SC1095	150.000	20.760	0.000	5.000
4	SC1094R8	160.000	20.965	0.154	3.400
5	SC1094R8	234.500	20.965	-0.184	-0.900
6	SC1094R8	236.910	22.317	-0.184	-0.914
7	SC1094R8	265.000	22.317	-0.184	-1.890
8	SC1095	275.000	22.112	-0.338	-1.130
9	SC1095	277.860	22.112	-0.338	-1.200
10	SC1095	280.000	20.076	-0.338	-1.260
11	SC1095	299.000	20.076	0.000	-3.560
12	SC1095	322.000	22.092	-12.562	-1.330

twist angles are detailed in Ref. 117. A total of 12 cross-sections are constructed and a surface is lofted through these sections to obtain the NURBS geometry of the rotor blade (see Figure 13). The blade cross-section data is summarized in Table 1. The final blade geometry is pitched at  $12^{\circ}$  for the following simulations.

## 5.2. Rotor-fuselage interaction

During flight, there is relative rotational motion between the helicopter rotor and fuselage. To model the interaction, the sliding interface formulation for compressible flows, introduced and validated for gas-turbine simulation in Ref. 62, is employed. Figure 14a illustrates the configuration of the sliding interface. The gray sliding disk divides the fluid domain into two parts: the domain inside the sliding disk rotates at the same speed with the rotor, while the domain outside remains stationary. The ALE formulation makes it straightforward to handle a simulation where both moving and stationary domains are present. The continuity of flow variables is enforced by the sliding interface formulation in a variationally consistent manner. The sliding interface formulation can be interpreted as a Discontinuous Galerkin method,<sup>118</sup> where the discretization is discontinuous only at the sliding interface.

**Remark 5.1.** A pioneering work on the aerodynamic simulation of a “full-machine”

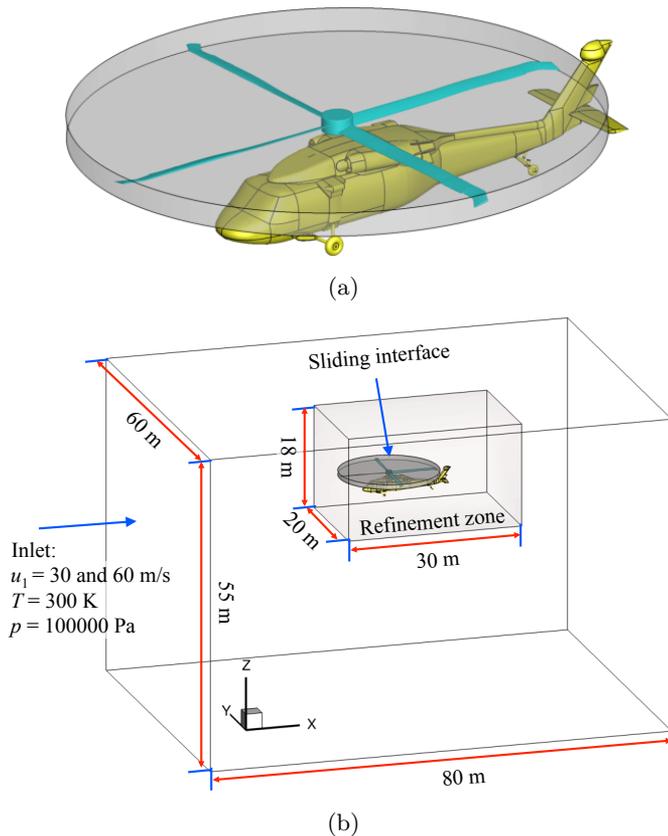


Fig. 14: (a) UH-60 Black Hawk helicopter surface model with a sliding interface around the rotor. (b) Problem domain and setup.

Comanche Helicopter in forward flight was carried out in 1999 by Behr and Tezduyar<sup>119</sup>, taking into account the interaction between the rotor and fuselage. It was computed with the Shear-Slip Mesh Update Method (SSMUM), which was introduced originally for two high-speed trains passing each other in a tunnel<sup>120,121</sup> and later refined and applied to the simulations of a propeller<sup>122</sup> and a helicopter.<sup>119</sup>

### 5.3. Problem setup

The fluid domain is a box with dimensions  $x \in [-40 \text{ m}, 40 \text{ m}]$ ,  $y \in [-30 \text{ m}, 30 \text{ m}]$ , and  $z \in [-40 \text{ m}, 15 \text{ m}]$ . The helicopter is positioned such that the center of the main rotor is located at  $(0 \text{ m}, 0 \text{ m}, 0.35 \text{ m})$ . We assume the helicopter flies forward in the  $-x$  direction, while the  $z$ -axis points upward. Boundary conditions are set as follows. To model forward flight, we hold the helicopter fuselage fixed and prescribe uniform inflow velocity of the same magnitude and opposite direction of the helicopter in

forward flight. We consider two flight speeds, 30 m/s and 60 m/s. We set the far field air pressure and temperature to 100 kPa and 300 K, respectively. The far-field air density is set to  $1.16 \text{ kg/m}^3$  and the ideal gas constant is taken as  $R = 287.06 \text{ J/(kg}\cdot\text{K)}$ . Under these conditions, the 60 m/s case corresponds to the Mach number of 0.17 and is close to the maximum forward flight speed of the UH-60 helicopter.<sup>123</sup> The main rotor spins counter-clockwise in the  $x$ - $y$  plane with a speed of 27 rad/s if observed from the top. The problem setup is illustrated in Figure 14b.

The rotor model defined in Section 5.1 can be easily generated as a clean, water-tight surface. As a result, we use the body-fitted approach to generate a high-quality boundary-layer mesh around the rotor. The no-slip boundary conditions on the rotor surface are set weakly using the symmetric Nitsche formulation. The fuselage model in this work is comprised of complex non-watertight surfaces. To avoid the geometry cleanup, we directly immerse the fuselage into a background fluid mesh and solve the fuselage aerodynamics using IMGA. The boundary conditions of the fuselage surface, which are the no-slip velocity and stagnation temperature boundary conditions, are set weakly using the non-symmetric Nitsche formulation.

## 5.4. Computational results

### 5.4.1. Mesh refinement study

To find a mesh resolution that yields a good balance between solution accuracy and computation costs, we perform a mesh refinement study on three sets of meshes. Note that the emphasis of this study is about the performance of the IMGA method, therefore we focus on the aerodynamic forces acting on the immersed fuselage. We keep only the fuselage in this setup for simplicity, and immerse it in the fluid domain meshed with tetrahedral elements. The boundary-layer region near the fuselage is locally refined. We then smoothly grow the mesh size from the fuselage to the fluid-domain outer boundary by constructing a refinement box defined by ( $x \in [-10 \text{ m}, 20 \text{ m}]$ ,  $y \in [-10 \text{ m}, 10 \text{ m}]$ , and  $z \in [-8 \text{ m}, 10 \text{ m}]$ ). The mesh size near the fuselage and inside the refinement zone are varied according to the values in Table 2. The rest of the domain is filled with tetrahedral elements that have a maximum element size of 2.0 m. The three meshes with different refinement levels are denoted as M1, M2 and M3, and the mesh statistics are shown in Table 2. The immersed fuselage models are provided as non-watertight STL surfaces, and the element size of the triangles is half of the near-fuselage mesh size used in the background mesh.

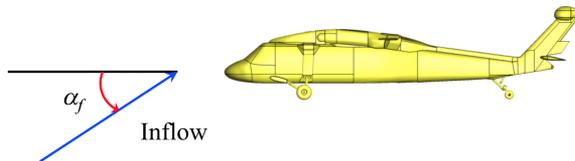
The simulations are carried out using a time step size of  $1.0 \times 10^{-3} \text{ s}$ . The time-averaged fuselage drag-force data for forward-flight speeds of 30 m/s and 60 m/s is reported in Table 3. The relative error in the drag force between M2 and M3 is smaller than 2% for both 30 m/s and 60 m/s cases. We conclude that M2 gives sufficiently accurate results, and use a similar mesh resolution for the remainder of the forward-flight simulations.

Table 2: Statistics of the fuselage-only meshes used in the mesh refinement study.

	Number of nodes	Number of elements	Near-fuselage mesh size (m)	Refinement-zone mesh size (m)
M1	1,409,011	8,265,993	0.045	0.45
M2	2,267,546	13,312,620	0.03	0.3
M3	3,396,768	19,930,505	0.02	0.2

Table 3: Fuselage drag ( $F_D$ ) for the two forward-flight speeds computed on meshes M1-M3.

	M1	M2	M3
$F_D$ , 30 m/s	759.2 N	830.2 N	846.1 N
(relative error)	(11.3 %)	(1.8 %)	–
$F_D$ , 60 m/s	2,701.1 N	2866.1 N	2,924.0 N
(relative error)	(7.6 %)	(1.9 %)	–

Fig. 15: Definition of the angle of attack ( $\alpha_f$ ).

#### 5.4.2. Experimental validation

To validate our IMGA framework using wind-tunnel test data, we carry out the UH-60 *fuselage-only* computations at different angles of attack. Extensive wind tunnel test data are presented for this case in a 1981 NASA technical report,<sup>123</sup> comprehensively covering different flight conditions in the low-speed flight regime. In this reference, normalized drag forces are reported as a function of the flow angle of attack  $\alpha_f$  (see Figure 15 for a definition).

Figure 16 shows the simulation results of IMGA compared to the wind-tunnel drag-force test data. The wind tunnel data span the range of angles from  $-90^\circ$  to  $90^\circ$ . Since the test data are fairly symmetric with respect to  $\alpha_f = 0^\circ$ , we only look at the range of angles from  $0^\circ$  to  $90^\circ$ . In the window of  $[0^\circ, 30^\circ]$ , there is a data point every  $5^\circ$ , and in the window  $[30^\circ, 90^\circ]$ , a data point is present every  $10^\circ$ . We simulate the same cases as reported in the experiments. The inflow velocity in the simulation is set to 30 m/s to better reflect the low-speed regime studied in the

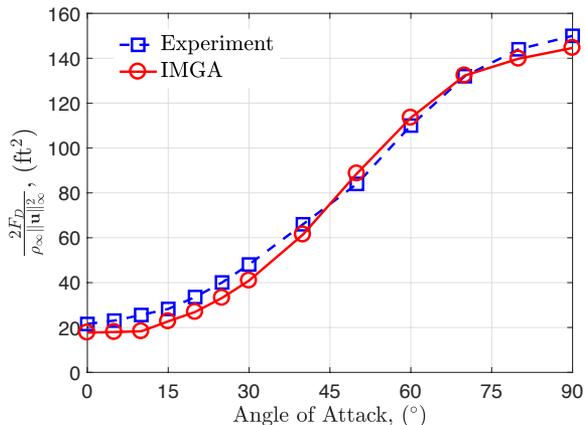


Fig. 16: Fuselage normalized drag force (in  $\text{ft}^2$ ) as a function of the angle of attack. Comparison of the IMGGA simulations with the wind-tunnel test data from Ref. 123.

wind-tunnel tests. Overall, a very good agreement between the IMGGA results and the wind-tunnel data is observed. The trend of the normalized drag force changing with  $\alpha_f$  is captured particularly well.

**Remark 5.2.** The wind-tunnel tests in Ref. 123 were performed using a quarter-scale model, and forces were scaled up to the full-scale helicopter so that the data can be used to design mathematical models for real-time piloted simulation.<sup>124</sup> The IMGGA calculations in the present work are carried out on a full-scale model.

#### 5.4.3. *Simulating the full machine*

In this section, we build the mesh for the full helicopter to model the interaction between rotor and fuselage. We assume the UH-60 helicopter is doing a straight and level flight, in which the roll, pitch and yaw angles are all  $0^\circ$ . Figure 17a illustrates the mesh design of the fluid domain. The mesh design can be separated in two parts by the non-matching sliding interface: the body-fitted mesh inside the sliding interface, and the non-body-fitted tetrahedral mesh outside. As shown in Figure 14, the domain inside the grey sliding interface contains the main rotor. We mesh this domain by first tessellating the main rotor surface into triangles with a mesh size of 0.04 m in the axial direction and 0.02 m in the chord-wise direction. We then generate a total of 15 layers of prismatic elements around the main rotor, with the first layer height of 0.005 m and a growth ratio of 1.1 (see Figure 17b). The rest of the domain inside the sliding interface is filled with tetrahedral elements with a maximum size of 0.1 m. The domain outside the sliding disk and inside the outer fluid domain is meshed using the same strategy as M2. Finally, the mesh contains 15,237,905 elements and is denoted as FM2. In this setup, the 27 rad/s rotor speed and the 8.179 m length of the blade result in a linear speed of 220.83 m/s at the

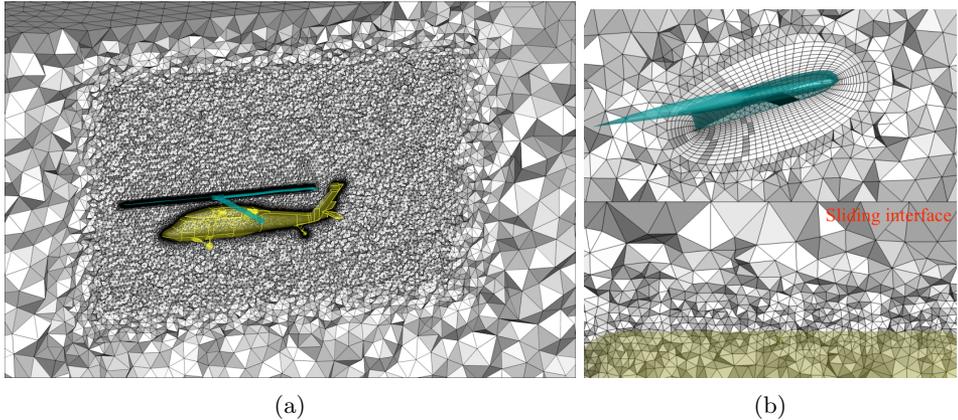


Fig. 17: (a) Mesh for the full-machine simulation; (b) Zoom on the prismatic boundary-layer mesh around the blade, sliding interface, and fuselage cutting through the background mesh.

blade tip. The latter figure corresponds to a Mach number of 0.63, and clearly shows the necessity of modeling aerodynamics using the compressible-flow assumption. The time step size is reduced to  $2 \times 10^{-5}$  s to achieve a similar CFL number with fuselage-only simulations.

Figure 18 shows the vortical structures around the helicopter and clearly illustrates a highly unsteady and turbulent flow field. From the side view, more vortex shedding is observed around the fuselage in the 60 m/s case, likely due to the higher forward-flight velocity. The top view clearly shows vortices generated by the rotor-blade tips. It is interesting to note that the vortices generated by the rotor tips are propagated downstream on both sides of the helicopter for the 30 m/s case, and only on the right side (assuming top view) of the helicopter for the 60 m/s case. This can be explained by the difference in the relative air flow velocities at the rotor tip in both cases. For the right side, since the rotor tip moves in the direction opposite to the air flow, the relative velocity is larger, suggesting a larger velocity gradient and stronger vortex propagating downstream of the rotor. For the left side, however, the rotor tip moves in the same direction as the incoming flow, leading to a smaller relative velocity and a weaker tip vortex, especially for the 60 m/s case. This also explains why the vortex core region is thinner on the left side than the right side for the case with a 30 m/s inflow. In addition, from the plots of the top view, we emphasize that the vortex structures, which are associated with the velocity gradient, are continuous and smooth across the non-matching sliding interfaces.

Figure 19 shows instantaneous pressure distribution on the fuselage surface at two different rotor positions. When the axis passing through the lead rotor blade is at a  $45^\circ$  angle to the fuselage axis, the pressure at the fuselage nose is lower than when the two axes are aligned (see Figure 19). We report the mean drag force

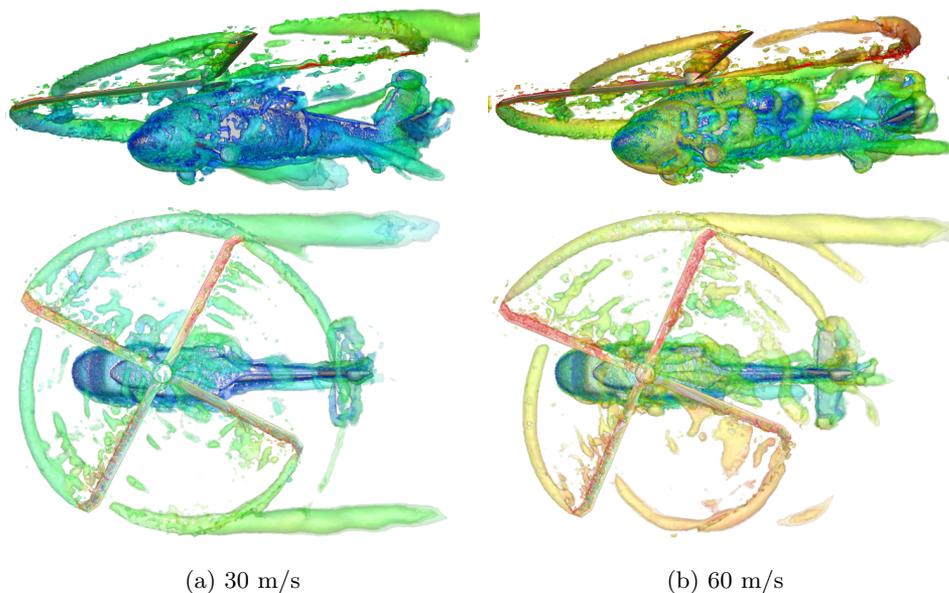


Fig. 18: Vorticity isosurfaces colored by the flow speed plotted using a color scale ranging from 0 m/s (blue) to  $\geq 80$  m/s (red).

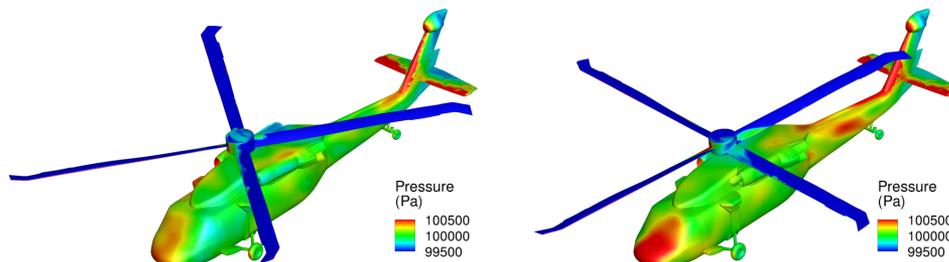


Fig. 19: Fuselage pressure contours for two different rotor positions.

on the fuselage obtained from the full-rotorcraft simulations in Table 4. Compared with the fuselage-only computations, the drag forces are higher when we consider the spinning rotor. As shown in Figure 19, the presence of the rotor causes the pressure on the fuselage nose to be higher than the stagnation pressure due to forward motion alone, which explains the difference in the drag force.

For the two cases of 30 m/s and 60 m/s flight speeds, the lift forces are noticeably different. For the 30 m/s case, the lift force of 110,974.5 N is obtained; while for the 60 m/s case, the lift force is predicted to be 136,117.9 N. The maximum take-off weight of the UH-60 helicopter is reported in Ref. 123 to be 20,250 lbs or 90,076.5 N.

Table 4: Fuselage drag at two different flight speeds obtained from fuselage-only and full-rotorcraft simulations.

	Fuselage only	Full rotorcraft
$F_D$ , 30 m/s	830.2 N	1407.2 N
$F_D$ , 60 m/s	2866.1 N	3378.3 N

Therefore, the simulations suggest that the rotor spinning at 27 rad/s certainly generates enough force to lift the helicopter.

## 6. Conclusion

We developed an IMGA formulation for the simulation of compressible flows. Weak enforcement of boundary conditions using a non-symmetric Nitsche approach is explored in the context of non-body-fitted discretizations. It is felt to be more robust than the more standard symmetric Nitsche approach because there is no formal lower bound on the penalty parameter, which allows us more flexibility in its choice. The IMGA formulation is validated extensively for a wide range of Mach and Reynolds numbers using benchmark tests. A significant portion of this paper is devoted to the study of a full-scale UH-60 Black Hawk helicopter in forward flight. All the computational results demonstrate very good accuracy and robustness of the proposed IMGA formulation. In fact, the formulation gives surprisingly accurate and smooth prediction of pressure pressure distribution, which is unexpected from unresolved or marginally resolved simulations of flows on non-body-fitted meshes. We attribute this enhanced accuracy, among other factors, to the good design of the weak-boundary-condition operator.

In addition, we highlight the exceptional flexibility offered by the IMGA framework since it can take various commonly used geometry representation formats directly into the computational analysis environment. In the examples we show that NURBS and analytic surfaces, which are heavily utilized in industrial design, can be directly used in the analysis without the labor-intensive meshing step. Traditional formats, such as tessellated surfaces, can also be used with greater flexibility since IMGA relaxes the requirement of a water-tight surface representation.

Finally, the aerodynamic simulations of the full-scale UH-60 Black Hawk helicopter clearly demonstrates the potential of the IMGA framework to greatly simplify analysis of geometrically-complex configurations without sacrificing the accuracy of simulation results.

## 7. Acknowledgments

This work was partially supported by the ARO Grant No. W911NF-14-1-0296. We also thank NAVAIR for providing funding for this work. Some computations

presented in this paper were carried at the Texas Advanced Computing Center (TACC). All this support is gratefully acknowledged.

## References

1. 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.
2. 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(1-3):689–723, 1982.
3. 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.
4. 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.
5. 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*, 65:207–235, 2011.
6. 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*, 65:236–253, 2011.
7. K. Takizawa, B. Henicke, T. E. Tezduyar, M.-C. Hsu, and Y. Bazilevs. Stabilized space–time computation of wind-turbine rotor aerodynamics. *Computational Mechanics*, 48:333–344, 2011.
8. 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: 647–657, 2011.
9. 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.
10. 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.

11. M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.
12. 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(3):461–481, 2014.
13. 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.
14. 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.
15. A. Korobenko, J. Yan, S. M. I. Gohari, S. Sarkar, and Y. Bazilevs. FSI Simulation of two back-to-back wind turbines in atmospheric boundary layer flow. *Computers & Fluids*, 158:167–175, 2017.
16. A. Korobenko, Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis. *Archives of Computational Methods in Engineering*, 2018. <https://doi.org/10.1007/s11831-018-9292-1>.
17. 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*, 45:77–89, 2009.
18. 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*, 46:3–16, 2010.
19. 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*, 9:481–498, 2010.
20. M.-C. Hsu and Y. Bazilevs. Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. *Finite Elements in Analysis and Design*, 47:593–599, 2011.
21. 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.
22. K. Takizawa, T. E. Tezduyar, T. Terahara, and T. Sasaki. Heart valve flow computation with the Space–Time Slip Interface Topology Change (ST-SITC) 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, 2018.
23. 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.
24. 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. In T. E. Tezduyar, editor, *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation: Research from Lead Investigators under Forty – 2018*, Modeling and Simulation in Science, Engineering and Technology, pages 29–89. Springer, 2018.
  25. K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19: 125–169, 2012.
  26. 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*, 50:835–854, 2012.
  27. 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.
  28. K. Takizawa, T. E. Tezduyar, C. Boswell, Y. Tsutsui, and K. Montel. Special methods for aerodynamic-moment calculations from parachute FSI modeling. *Computational Mechanics*, 55:1059–1069, 2015.
  29. 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(1):010903–10, 2012.
  30. 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*, 50:743–760, 2012.
  31. K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T. E. Tezduyar. Computer modeling techniques for flapping-wing aerodynamics of a locust. *Computers & Fluids*, 85:125–134, 2013.
  32. 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.
  33. Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. John Wiley & Sons, Chichester, 2013.
  34. 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*, 21:481–508, 2014.
  35. T. E. Tezduyar and K. Takizawa. Space–time computations in practical engineering applications: a summary of the 25-year history. *Computational Mechanics*, 63(4):747–753, 2018.
  36. 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.
37. 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.
  38. A. A. Johnson and T. E. Tezduyar. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics*, 23:130–143, 1999.
  39. 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.
  40. 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.
  41. 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.
  42. 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.
  43. Y. Bazilevs and T. J. R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43:143–150, 2008.
  44. 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.
  45. 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*, 141:124–134, 2016.
  46. D. L. Marcum and J. A. Gaither. Unstructured grid generation for aerospace applications. In M. D. Salas and W. K. Anderson, editors, *Computational Aerosciences in the 21st Century*, volume 8, pages 189–209. Springer Netherlands, 2000.
  47. Z. J. Wang and K. Srinivasan. An adaptive Cartesian grid generation method for ‘Dirty’ geometry. *International Journal for Numerical Methods in Fluids*, 39:703–717, 2002.
  48. M. W. Beall, J. Walsh, and M. S. Shephard. A comparison of techniques for geometry access related to mesh generation. *Engineering with Computers*, 20: 210–221, 2004.
  49. Y. K. Lee, C. K. Lim, H. Ghazialam, H. Vardhan, and E. Eklund. Surface mesh generation for dirty geometries by the Cartesian shrink-wrapping technique. *Engineering with Computers*, 26:377–390, 2010.
  50. C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

51. A. Gilmanov and F. Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457–492, 2005.
52. J.-I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224(2):757–784, 2007.
53. R. Mittal, H. Dong, M. Bozkurtas, F. M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825–4852, 2008.
54. 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.
55. F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M.-C. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 141:135–154, 2016.
56. M.-C. Hsu, C. Wang, F. Xu, A. J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.
57. C. Wang, F. Xu, M.-C. Hsu, and A. Krishnamurthy. Rapid B-rep model preprocessing for immersogeometric analysis using analytic surfaces. *Computer Aided Geometric Design*, 52–53:190–204, 2017.
58. J. A. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1970.
59. Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.
60. 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.
61. 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.
62. F. Xu, G. Moutsanidis, D. Kamensky, M.-C. Hsu, M. Murugan, A. Ghoshal, and Y. Bazilevs. Compressible flows on moving domains: Stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling. *Computers & Fluids*, 158:201–220, 2017.
63. A. Embar, J. Dolbow, and I. Harari. Imposing Dirichlet boundary conditions

- with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83(7):877–898, 2010.
64. J. T. Oden, I. Babuška, and C. E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics*, 146:491–519, 1998.
  65. C. E. Baumann and J. T. Oden. A discontinuous *hp* finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175:311–341, 1999.
  66. C. E. Baumann and J. T. Oden. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 31(1):79–95, 1999.
  67. D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S. K. F. Stoter, Y. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.
  68. H. Atkins and C.-W. Shu. Analysis of the discontinuous Galerkin method applied to the diffusion operator. In *AIAA 14th Computational Fluid Dynamics Conference*, Norfolk, Virginia, 1999. AIAA-99-3306.
  69. R. M. Kirby and G. E. Karniadakis. Selecting the numerical flux in discontinuous galerkin methods for diffusion problems. *Journal of Scientific Computing*, 22(1-3):385–411, 2005.
  70. F. Heimann, C. Engwer, O. Ippisch, and P. Bastian. An unfitted interior penalty discontinuous galerkin method for incompressible navier–stokes two-phase flow. *International Journal for Numerical Methods in Fluids*, 71(3):269–293, 2013.
  71. W. G. Dettmer, C. Kadapa, and D. Perić. A stabilised immersed boundary method on hierarchical b-spline grids. *Computer Methods in Applied Mechanics and Engineering*, 311:415–437, 2016.
  72. G. Hauke and T. J. R. Hughes. A comparative study of different sets of variables for solving compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 153:1–44, 1998.
  73. 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*, 59:307–325, 1986.
  74. T. J. R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.
  75. T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58:329–339, 1986.
  76. 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. *Comput. Methods Appl. Mech. and Engrg.*, 89:141–219, 1991.
77. R. C. Almeida and A. C. Galeão. An adaptive Petrov–Galerkin formulation for the compressible Euler and Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 129(1):157–176, 1996.
  78. T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. In E. Stein, R. De Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 17. John Wiley & Sons, 2004.
  79. T. E. Tezduyar and M. Senga. Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195:1621–1632, 2006.
  80. T. E. Tezduyar, M. Senga, and D. Vicker. Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and  $YZ\beta$  shock-capturing. *Computational Mechanics*, 38:469–481, 2006.
  81. T. E. Tezduyar and M. Senga. SUPG finite element computation of inviscid supersonic flows with  $YZ\beta$  shock-capturing. *Computers & Fluids*, 36:147–159, 2007.
  82. F. Rispoli, R. Saavedra, A. Corsini, and T. E. Tezduyar. Computation of inviscid compressible flows with the V-SGS stabilization and  $YZ\beta$  shock-capturing. *International Journal for Numerical Methods in Fluids*, 54:695–706, 2007.
  83. 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  $YZ\beta$  shock-capturing. *Journal of Applied Mechanics*, 76:021209, 2009.
  84. 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  $YZ\beta$  shock-capturing. *Computational Mechanics*, 55:1201–1209, 2015.
  85. 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*, 62:1169–1186, 2018.
  86. 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*, 45:217–284, 1984.
  87. T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54:223–234, 1986.
  88. 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*, 58:305–328, 1986.

89. T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multi-dimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 63:97–112, 1987.
90. 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*, 104:397–422, 1993.
91. S. K. Aliabadi and T. E. Tezduyar. Space–time finite element computation of compressible flows involving moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 107(1–2):209–223, 1993.
92. G. Hauke and T. J. R. Hughes. A unified approach to compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 113:389–396, 1994.
93. G. Hauke. Simple stabilizing matrices for the computation of compressible flows in primitive variables. *Computer Methods in Applied Mechanics and Engineering*, 190:6881–6893, 2001.
94. T. J. R. Hughes, G. Scovazzi, and T. E. Tezduyar. Stabilized methods for compressible flows. *Journal of Scientific Computing*, 43:343–368, 2010.
95. 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(04):771–806, 2017.
96. T. Kanai, K. Takizawa, T. E. Tezduyar, T. Tanaka, and A. Hartmann. Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization. *Computational Mechanics*, published online, DOI: 10.1007/s00466-018-1595-4, July 2018.
97. S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 3rd ed.* Springer-Verlag, New York, 2008.
98. I. Harari and E. Grosu. A unified approach for embedded boundary conditions for fourth-order elliptic problems. *International Journal for Numerical Methods in Engineering*, 104(7):655–675, 2015.
99. W. Jiang, C. Annavarapu, J. E. Dolbow, and I. Harari. A robust nitsche’s formulation for interface problems with spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 104(7):676–696, 2015.
100. 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.
101. M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal of Numerical Analysis*, 15:152–161, 1978.

102. B. Riviere, M. F. Wheeler, and V. Girault. A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM Journal of Numerical Analysis*, 39(3):902–931, 2001.
103. J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- $\alpha$  method. *Journal of Applied Mechanics*, 60:371–75, 1993.
104. K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized- $\alpha$  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.
105. 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.
106. F. Shakib, T. J. R. Hughes, and Z. Johan. A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 75:415–456, 1989.
107. A. Düster, J. Parvzian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197:3768–3782, 2008.
108. I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, and P. Shirley. State of the art in ray tracing animated scenes. *Computer Graphics Forum*, 28(6):1691–1722, 2009.
109. G. Young and A. Krishnamurthy. GPU-accelerated generation and rendering of multi-level voxel representations of solid models. *Computers & Graphics*, 75:11–24, 2018.
110. A. Düster, J. Parvzian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45–48):3768–3782, 2008.
111. M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.
112. G. Karypis and V. Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 1998.
113. O. Rodriguez. The circular cylinder in subsonic and transonic flow. *AIAA journal*, 22(12):1713–1718, 1984.
114. C.-Y. Xu, L.-W. Chen, and X.-Y. Lu. Large-eddy simulation of the compressible flow past a wavy cylinder. *Journal of Fluid Mechanics*, 665:238–273, 2010.
115. M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Physics of Fluids A*, 2(5):765–777, 1990.

116. J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.
117. W. G. Bousman. Aerodynamic characteristics of SC1095 and SC1094 R8 airfoils. NASA Technical Report NASA/TP-2003-212265, NASA, 2003.
118. 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*, 39:1749–1779, 2002.
119. M. Behr and T. Tezduyar. Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components. *Computer Methods in Applied Mechanics and Engineering*, 190:3189–3200, 2001.
120. V. Kalro and T. E. Tezduyar. Parallel finite element computation of 3D incompressible flows on MPPs. In W. G. Habashi, editor, *Solution Techniques for Large-Scale CFD Problems*. John Wiley & Sons, 1995.
121. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke. Flow simulation and high performance computing. *Computational Mechanics*, 18:397–412, 1996.
122. M. Behr and T. Tezduyar. The Shear-Slip Mesh Update Method. *Computer Methods in Applied Mechanics and Engineering*, 174:261–274, 1999.
123. J. J. Howlett. UH-60A Black Hawk engineering simulation program. Volume 1: Mathematical model. NASA Technical Report NASA-CR-16630, NASA, 1981.
124. K. B. Hilbert. A mathematical model of the UH-60 helicopter. NASA Technical Report NASA-TM-85890, NASA, 1984.