Compressible flows on moving domains: Stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling

Fei Xu[a], George Moutsanidis[b], David Kamensky[b], Ming-Chen Hsu[a], Muthuvel Murugan[c], Anindya Ghoshal[c], Yuri Bazilevs[b,*]

[a]*Department of Mechanical Engineering, Iowa State University, 2025 Black Engineering, Ames, IA 50011, USA*
[b]*Department of Structural Engineering, University of California, San Diego, La Jolla, CA 92037, USA*
[c]*Vehicle Technology Directorate, US Army Research Laboratory, Aberdeen Proving Ground, MD 21005, USA*

Abstract

A novel stabilized formulation for 3D compressible viscous flows on moving domains is developed. New weak imposition of essential boundary conditions and sliding-interface formulations are also proposed in the context of moving-domain compressible flows. The new formulation is successfully tested on a set of examples spanning a wide range of Reynolds and Mach numbers showing its superior robustness. Experimental validation of the new formulation is also carried out with good success. In addition, the formulation is applied to simulate flow inside a gas turbine stage, illustrating its potential to support design of real engineering systems through high-fidelity aerodynamic analysis.

Keywords: Compressible flow, Stabilized methods, Discontinuity capturing, Weak essential boundary conditions, Sliding-interface formulation, Gas turbine

Contents

*Corresponding author
Email address: yuri@ucsd.edu (Yuri Bazilevs)

The final publication is available at Elsevier via https://doi.org/10.1016/j.compfluid.2017.02.006

1. Introduction

The success of finite element methods in solid and structural mechanics, heat conduction, and other areas in 1970s encouraged its development and use to simulate flow problems. Stabilized finite element methods for fluid mechanics were introduced, and the first of them was the streamline upwind/Petrov–Galerkin (SUPG) method [1] for incompressible flows. The key idea of SUPG was to add a residual-based stabilization term to the Galerkin form of the governing equations in order to enhance the stability for higher Reynolds number flows while retaining consistency of the formulation. SUPG was extended to compressible flows using conservation variables in [2–4]. The concept of SUPG was further refined and studied for entropy variables in [5–7], and then generalized to arbitrary variable sets in [8, 9]. Over the years, significant progress was made in stabilized methods for compressible flows. The one perhaps most relevant to this paper was combining a

new version [10, 11] of the compressible-flow SUPG method [2–4] with the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method [12–14]. The DSD/SST method (now also called the "ST method") was introduced for flow problems with moving boundaries/interfaces, including fluid–structure interaction (FSI). The method resulting from this straightforward mixture of the DSD/SST concept and the compressible-flow SUPG method, which is now called compressible-flow ST SUPG method, was first tested in [15]. This was followed by computations for air intake of a jet engine with adjustable spool at supersonic speeds [16], aerodynamics of two high-speed trains in a tunnel [14], liquid propellant guns [17, 18], and compressible-flow FSI [19, 20]. Other progress included large-scale parallel computations [16, 21–24], unified formulations of incompressible and compressible flows [8, 25], and the development of stabilization parameters [10, 11, 26–32].

It was observed early on that when stabilized methods were applied to compressible flow analysis, oscillations occurred in the vicinity of shocks and other sharp solution features. Hughes et al. [33, 34] proposed a class of shock- or discontinuity-capturing methods that provide additional dissipation by adding mesh- and solution-dependent artificial viscosity terms to a stabilized formulation. These viscosities are often residual-based, and thus preserve consistency of the formulation. These shock-capturing methods were in the context of entropy variables. In a 1991 ASME paper [10], the original compressible-flow SUPG method, now called "$(\text{SUPG})_{82}$", was supplemented with a very similar shock-capturing term, which included a shock-capturing parameter that is now called "δ_{91}". The shock-capturing parameter was derived from the one given in [7] for the entropy variables. It was shown in that, with the added shock-capturing term, $(\text{SUPG})_{82}$ was very comparable in accuracy to $(\text{SUPG})_{82}$ recast in entropy variables. In the 2D inviscid-flow test computations reported in [11] soon after that, $(\text{SUPG})_{82}$ and $(\text{SUPG})_{82}$ recast in entropy variables yielded indistinguishable results. Following these works, references [9, 35] generalized discontinuity-capturing methods to arbitrary solution-variable sets. Further developments include the discontinuity-capturing directional dissipation (DCDD) stabilization for incompressible flows [28, 36] and the YZβ shock capturing [28–32, 37–41], which is based on a scaled residual and has a parameter (β) that controls the degree of shock smoothness. Numerical experiments in [30–32] demonstrated that these new discontinuity capturing techniques are relatively simple to implement and give results of comparable or even improved accuracy relative to earlier approaches. A concise summary of stabilized methods and discontinuity-capturing techniques for compressible flows may be found in a recent review article [42] and references therein.

In this paper, we make use of SUPG stabilization and discontinuity capturing to develop a novel numerical formulation for the Navier–Stokes equations of compressible flows in the Arbitrary Lagrangian–Eulerian (ALE) frame [43] suitable for moving-domain simulations. Early developments in stabilized ALE-based finite-element methods for compressible flows may be found

in [44–46]. In the present effort, we introduce several improvements to the existing formulations, as well develop new techniques, such as weakly enforced essential boundary condition and sliding-interface formulations, that enlarge the scope and applicability of moving-domain, finite-element-based compressible flow formulations.

Weakly enforced no-slip boundary conditions [47] are imposed on solid surfaces in order to avoid excessive resolution of thin, and often turbulent, boundary layers. Weak imposition of essential boundary conditions in the sense of Nitsche's method [48] for incompressible flows was first introduced in [47], and further refined in [49, 50]. The most distinguishing feature of this method is the added flexibility to allow the flow to slip on the solid surface in the case when the wall-normal mesh size is relatively large [50–52]. This feature allows one to achieve good accuracy on relatively coarse boundary-layer meshes. Weakly enforced boundary conditions have been successfully applied to simulations of wall-bounded turbulent flows [49, 50] and wind turbines [52–55]. More recently, weak enforcement of no-slip conditions was developed and applied in the context of immersogeometric analysis [56–59], which led to solutions of higher-order accuracy on non-boundary-fitted meshes. In the present work, we propose an extension of weakly enforced essential boundary conditions in the context of compressible flows, which brings the aforementioned advantages to this important area of computational fluid mechanics.

The sliding-interface formulation for incompressible flows was introduced in [60] for simulating flows with objects in relative motion. The formulation was comprehensively studied and refined in [54, 55], mostly with application to wind turbines. The sliding-interface formulation may be interpreted as a Discontinuous Galerkin method [61], where the basis functions are continuous inside the interior of subdomains but not at the sliding interface. In the incompressible-flow regime, the sliding-interface formulation was recently extended to the space–time (ST) variational multiscale (VMS) method [62–69], and the extension is called the "ST Slip Interface (ST-SI)" method [70–76]. In this work, we develop a compressible-flow counterpart of the sliding-interface formulation.

This paper is organized as follows. In Section 2, we develop a complete numerical formulation of the Navier–Stokes equations of compressible flows. In Section 3, we compute several 2D and 3D examples to verify and validate the different constituents of our compressible-flow numerical methodology. We focus on a broad range of Reynolds and Mach numbers to illustrate the robustness of the numerical formulation. In Section 4, we apply the methods developed to simulate flow inside a gas turbine stage, illustrating the potential of our methods to support design for real engineering systems through high-fidelity aerodynamic analysis. In Section 5, we draw conclusions.

2. Numerical methodology

2.1. Governing equations of compressible flows

2.1.1. Preliminaries

The Navier–Stokes equations of compressible flows are often expressed using a vector of conservation variables $\tilde{\mathbf{U}}$ defined as

$$\tilde{\mathbf{U}} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{\text{tot}} \end{bmatrix}, \tag{1}$$

where ρ is the density, u_i is the i^{th} velocity component, $i = 1, ..., d$, where $d = 2$ or 3 is the space dimension, and $e_{\text{tot}} = e + \|\mathbf{u}\|^2/2$ is the fluid total energy density, where e is the fluid internal energy density and $\|\mathbf{u}\|$ is the velocity magnitude.

We also introduce a vector of primitive variables based on pressure or the pressure-primitive variables $\mathbf{Y}$ defined as

$$\mathbf{Y} = \begin{bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{bmatrix}, \tag{2}$$

where p is the pressure and T is the temperature. Pressure, density, and temperature are related through an equation of state. Here we make use of the ideal gas equation of state, which may be written as

$$p = \rho R T, \tag{3}$$

where R is the ideal gas constant. Furthermore, we assume a calorically perfect gas and define the fluid internal energy density as

$$e = c_{\text{v}} T, \tag{4}$$

where $c_{\text{v}} = R/(\gamma - 1)$ is the specific heat at constant volume and γ is the heat capacity ratio.

Throughout the paper we use $(\cdot)_{,t}$ to denote a partial time derivative holding the spatial coordinates $\mathbf{x}$ fixed, and we use $(\cdot)_{,i}$ to denote the spatial gradient.

2.1.2. Strong form

The Navier–Stokes equations of compressible flows, which express pointwise balance of mass, linear momentum, and energy, may be written in terms of $\tilde{\mathbf{U}}$ as

$$\tilde{\mathbf{U}}_{,t} + \tilde{\mathbf{F}}^{\text{adv}}_{i,i} = \tilde{\mathbf{F}}^{\text{diff}}_{i,i} + \tilde{\mathbf{S}}, \tag{5}$$

where $\tilde{\mathbf{F}}^{\text{adv}}_i$ and $\tilde{\mathbf{F}}^{\text{diff}}_i$ are the vectors of advective and diffusive fluxes, respectively, and $\tilde{\mathbf{S}}$ is the source term. The residual of the compressible-flow equations may be defined as

$$\widetilde{\mathbf{Res}} = \tilde{\mathbf{U}}_{,t} + \tilde{\mathbf{F}}^{\text{adv}}_{i,i} - \tilde{\mathbf{F}}^{\text{diff}}_{i,i} - \tilde{\mathbf{S}}. \tag{6}$$

We further split the advective flux into $\tilde{\mathbf{F}}^{\text{adv}}_i = \tilde{\mathbf{F}}^{\text{adv}\backslash p}_i + \tilde{\mathbf{F}}^{p}_i$. The aforementioned fluxes are defined as

$$\tilde{\mathbf{F}}^{\text{adv}\backslash p}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \\ \rho u_i \left(e + \|\mathbf{u}\|^2/2\right) \end{bmatrix}, \tag{7}$$

$$\tilde{\mathbf{F}}^{p}_i = \begin{bmatrix} 0 \\ p\delta_{1i} \\ p\delta_{2i} \\ p\delta_{3i} \\ p u_i \end{bmatrix}, \tag{8}$$

$$\tilde{\mathbf{F}}^{\text{diff}}_i = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} u_j - q_i \end{bmatrix}, \tag{9}$$

where δ_{ij} is the Kronecker delta. The viscous stress τ_{ij} and heat flux q_i are given by

$$\tau_{ij} = \lambda u_{k,k} \delta_{ij} + \mu \left(u_{i,j} + u_{j,i}\right), \tag{10}$$

$$q_i = -\kappa T_{,i}, \tag{11}$$

where μ is the dynamic viscosity, $\lambda = -2\mu/3$ based on Stokes' hypothesis, and κ is the thermal conductivity.

2.1.3. Reduced form of the energy equation

Introducing the mass and momentum balance into the energy equation, we can simplify the compressible-flow equation system. The balance laws given by Eq. (5) become

$$\mathbf{U}_{,t} + \mathbf{F}^{\text{adv}}_{i,i} + \mathbf{F}^{\text{sp}} = \mathbf{F}^{\text{diff}}_{i,i} + \mathbf{S}, \tag{12}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e \end{bmatrix}, \tag{13}$$

$$\mathbf{F}^{\text{adv}}_i = \mathbf{F}^{\text{adv}\backslash p}_i + \mathbf{F}^{p}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \\ \rho u_i e \end{bmatrix} + \begin{bmatrix} 0 \\ p\delta_{1i} \\ p\delta_{2i} \\ p\delta_{3i} \\ 0 \end{bmatrix}, \tag{14}$$

$$\mathbf{F}^{\text{diff}}_i = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ -q_i \end{bmatrix}, \tag{15}$$

and the term $\mathbf{F}^{\text{sp}}$ is the contribution of stress power in the energy equation, which has the form

$$\mathbf{F}^{\text{sp}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ pu_{i,i} - \tau_{ij}u_{j,i} \end{bmatrix}. \tag{16}$$

A modified residual function is now defined as

$$\mathbf{Res} = \mathbf{U}_{,t} + \mathbf{F}^{\text{adv}}_{i,i} + \mathbf{F}^{\text{sp}} - \mathbf{F}^{\text{diff}}_{i,i} - \mathbf{S}. \tag{17}$$

2.1.4. Weak form

Let $\Omega \in \mathbb{R}^d$ denote the fluid domain and let Γ be its boundary. Considering $\mathbf{U}, \mathbf{F}_i^{\text{adv}\backslash p}, \mathbf{F}_i^p, \mathbf{F}^{\text{sp}}$, and $\mathbf{F}_i^{\text{diff}}$ as functions of $\mathbf{Y}$, i.e., $\mathbf{U}(\mathbf{Y})$, $\mathbf{F}_i^{\text{adv}\backslash p}(\mathbf{Y})$, $\mathbf{F}_i^p(\mathbf{Y})$, $\mathbf{F}^{\text{sp}}(\mathbf{Y})$, and $\mathbf{F}_i^{\text{diff}}(\mathbf{Y})$, the weak form of Eq. (12) may be stated as: Find $\mathbf{Y} \in \mathcal{S}$, such that $\forall \mathbf{W} \in \mathcal{V}$,

$$B(\mathbf{W}, \mathbf{Y}) - F(\mathbf{W}) = 0, \tag{18}$$

where

$$\begin{aligned} B(\mathbf{W}, \mathbf{Y}) = &\int_\Omega \mathbf{W} \cdot \left(\mathbf{U}_{,t}(\mathbf{Y}) + \mathbf{F}_{i,i}^{\text{adv}\backslash p}(\mathbf{Y}) + \mathbf{F}^{\text{sp}}(\mathbf{Y}) \right) d\Omega \\ &- \int_\Omega \mathbf{W}_{,i} \cdot \left(\mathbf{F}_i^p(\mathbf{Y}) - \mathbf{F}_i^{\text{diff}}(\mathbf{Y}) \right) d\Omega, \end{aligned} \tag{19}$$

and

$$F(\mathbf{W}) = \int_\Omega \mathbf{W} \cdot \mathbf{S} \, d\Omega + \int_{\Gamma_\text{h}} \mathbf{W} \cdot \mathbf{H} \, d\Gamma. \tag{20}$$

$\mathcal{S}$ and $\mathcal{V}$ are the trial-function space for the pressure-primitive variables and test-function space for the compressible-flow equation system, respectively, and Γ_h is the subset of Γ where the traction and heat flux boundary conditions $\mathbf{H}$ are enforced. The vector $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 0 \\ -pn_1 + \tau_{1i} n_i \\ -pn_2 + \tau_{2i} n_i \\ -pn_3 + \tau_{3i} n_i \\ -q_i n_i \end{bmatrix}, \tag{21}$$

where n_i is the i^{th} component of outward unit surface normal vector $\mathbf{n}$. The vector $\mathbf{H}$ contains the fluid traction in the momentum-equation slot and heat flux in the energy-equation slot.

Remark 1. Note that the choice of the variable set does not change the balance laws. In the present work, rather than using conservation variables as the problem unknowns, pressure-primitive variables are chosen. One reason for this choice is that, unlike for conservation variables, the incompressible limit of the compressible-flow equations is well defined for the pressure-primitive variables [9]. Pressure-primitive variables are also convenient for setting boundary conditions and implementing fluid–structure interaction (FSI) coupling. In addition, as will become evident in the later sections, pressure-primitive variables lead to a natural extension of the weakly enforced essential boundary condition and sliding-interface formulations, originally defined for incompressible

flows, to the compressible flow regime.

Remark 2. The use of reduced-energy equation is likewise convenient for FSI modeling involving thermally-coupled solids, because only the heat flux appears in the energy-equation slot of vector $\mathbf{H}$.

2.2. Constituents of the discrete formulation

2.2.1. Quasi-linear form

The Navier–Stokes equations of compressible flows given by Eq. (12) may be expressed in the following quasi-linear form involving the conservation variables $\mathbf{U}$:

$$\mathbf{U}_{,t} + \hat{\mathbf{A}}_i \mathbf{U}_{,i} + \hat{\mathbf{A}}_i^{\text{sp}} \mathbf{U}_{,i} - \left(\hat{\mathbf{K}}_{ij} \mathbf{U}_{,j}\right)_{,i} - \mathbf{S} = \mathbf{0}, \tag{22}$$

where $\hat{\mathbf{A}}_i = \dfrac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{U}}$, $\hat{\mathbf{A}}_i^{\text{sp}}$ is such that $\hat{\mathbf{A}}_i^{\text{sp}} \mathbf{U}_{,i} = \mathbf{F}^{\text{sp}}$, and $\hat{\mathbf{K}}_{ij}$ is such that $\hat{\mathbf{K}}_{ij} \mathbf{U}_{,j} = \mathbf{F}_i^{\text{diff}}$.

Analogously, using the pressure-primitive variables $\mathbf{Y}$, the quasi-linear form of Eq. (22) may be written as

$$\mathbf{A}_0 \mathbf{Y}_{,t} + \mathbf{A}_i \mathbf{Y}_{,i} + \mathbf{A}_i^{\text{sp}} \mathbf{Y}_{,i} - \left(\mathbf{K}_{ij} \mathbf{Y}_{,j}\right)_{,i} - \mathbf{S} = \mathbf{0}, \tag{23}$$

where $\mathbf{A}_0 = \dfrac{\partial \mathbf{U}}{\partial \mathbf{Y}}$, $\mathbf{A}_i = \dfrac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{Y}} = \dfrac{\partial \mathbf{F}_i^{\text{adv}}}{\partial \mathbf{U}} \dfrac{\partial \mathbf{U}}{\partial \mathbf{Y}} = \hat{\mathbf{A}}_i \mathbf{A}_0$, $\mathbf{A}_i^{\text{sp}}$ is such that $\mathbf{A}_i^{\text{sp}} \mathbf{Y}_{,i} = \mathbf{F}^{\text{sp}}$, and $\mathbf{K}_{ij}$ is such that $\mathbf{K}_{ij} \mathbf{Y}_{,j} = \mathbf{F}_i^{\text{diff}}$. Explicit expressions for the matrices appearing in the quasi-linear forms are provided in Appendix A.

2.2.2. Moving-domain formulation

Using the space–time Piola transformation and following the steps in [77], the convective ALE formulation of the Navier–Stokes equations of compressible flows may be stated as

$$\mathbf{U}_{,t}\big|_{\hat{\mathbf{x}}} + \mathbf{F}_{i,i}^{\text{adv}} + \mathbf{F}^{\text{sp}} - \hat{u}_i \mathbf{U}_{,i} - \mathbf{F}_{i,i}^{\text{diff}} - \mathbf{S} = \mathbf{0}, \tag{24}$$

where $\hat{u}_i$ is the i^{th} component of the domain velocity $\hat{\mathbf{u}}$ and $(\cdot)_{,t}\big|_{\hat{\mathbf{x}}}$ denotes a partial time derivative holding the referential coordinates $\hat{\mathbf{x}}$ fixed. The quasi-linear form of Eq. (24) may be written for the conservation variables as

$$\mathbf{U}_{,t}\big|_{\hat{\mathbf{x}}} + \left(\hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^{\text{sp}} - \hat{u}_i \mathbf{I}\right) \mathbf{U}_{,i} - \left(\hat{\mathbf{K}}_{ij} \mathbf{U}_{,j}\right)_{,i} - \mathbf{S} = \mathbf{0}. \tag{25}$$

In the case of the pressure-primitive variables, Eq. (24) becomes

$$\mathbf{A}_0 \mathbf{Y}_{,t}\big|_{\hat{\mathbf{x}}} + \left(\mathbf{A}_i + \mathbf{A}_i^{\text{sp}} - \hat{u}_i \mathbf{A}_0\right) \mathbf{Y}_{,i} - \left(\mathbf{K}_{ij} \mathbf{Y}_{,j}\right)_{,i} - \mathbf{S} = \mathbf{0}. \tag{26}$$

The residual for the ALE form of the compressible-flow equations may be expressed as

$$\mathbf{Res} = \mathbf{A}_0 \mathbf{Y}_{,t}\big|_{\hat{\mathbf{x}}} + \left(\mathbf{A}_i + \mathbf{A}_i^{\mathrm{sp}} - \hat{u}_i \mathbf{A}_0\right) \mathbf{Y}_{,i} - \left(\mathbf{K}_{ij} \mathbf{Y}_{,j}\right)_{,i} - \mathbf{S}. \tag{27}$$

We also introduce the matrices $\hat{\mathbf{A}}_i^{\mathrm{ALE}} = \hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^{\mathrm{sp}} - \hat{u}_i \mathbf{I}$ for the conservation variables, and $\mathbf{A}_i^{\mathrm{ALE}} = \mathbf{A}_i + \mathbf{A}_i^{\mathrm{sp}} - \hat{u}_i \mathbf{A}_0$ for the pressure-primitive variables.

2.2.3. SUPG operator

We assume the time-dependent fluid domain Ω is divided into N_{el} spatial finite elements each denoted by Ω^e, and define the SUPG operator as follows:

$$B_{\mathrm{SUPG}}(\mathbf{W}, \mathbf{Y}) = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\mathbf{A}_i^{\mathrm{ALE}} \right)^{\mathrm{T}} \mathbf{W}_{,i} \right) \cdot \boldsymbol{\tau}_{\mathrm{SUPG}} \mathbf{Res}(\mathbf{Y}) \, d\Omega, \tag{28}$$

where the stabilization matrix $\boldsymbol{\tau}_{\mathrm{SUPG}}$ to this day remains a subject of active research [28–31]. In the present work we adopt a philosophy of designing $\boldsymbol{\tau}_{\mathrm{SUPG}}$ for the conservation variables and transforming to the pressure-primitive-variable formulation. For this, we employ the following design condition [27]:

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\mathbf{A}_i^{\mathrm{ALE}} \right)^{\mathrm{T}} \mathbf{W}_{,i} \right) \cdot \boldsymbol{\tau}_{\mathrm{SUPG}} \mathbf{Res} \, d\Omega = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\hat{\mathbf{A}}_i^{\mathrm{ALE}} \right)^{\mathrm{T}} \mathbf{W}_{,i} \right) \cdot \hat{\boldsymbol{\tau}}_{\mathrm{SUPG}} \mathbf{Res} \, d\Omega, \tag{29}$$

which yields

$$\boldsymbol{\tau}_{\mathrm{SUPG}} = \mathbf{A}_0^{-1} \hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}, \tag{30}$$

where $\mathbf{A}_0^{-1} = \mathbf{Y}_{,\mathbf{U}}$ is the transformation matrix between the two variable sets. The stabilization matrix $\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}$ may be defined as [26]

$$\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}} = \left(\frac{4}{\Delta t^2} \mathbf{I} + G_{ij} \hat{\mathbf{A}}_i^{\mathrm{ALE}} \hat{\mathbf{A}}_j^{\mathrm{ALE}} + C_I G_{ij} G_{kl} \hat{\mathbf{K}}_{ik} \hat{\mathbf{K}}_{lj} \right)^{-\frac{1}{2}}, \tag{31}$$

where Δt is the time step size, C_I is a positive constant derived from an appropriate element-wise inverse estimate [78–80], and G_{ij} is the element metric tensor computed as

$$G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}, \tag{32}$$

where $\mathbf{x}(\xi)$ denotes the mapping from the parametric element to its physical-domain counterpart.

The definition of $\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}$ in Eq. (31) requires the computation of the square-root-inverse of a

5×5 matrix in 3D. For this purpose, the Denman–Beavers method [81, 82] is employed. The Denman–Beavers method is an iterative technique that derives from the Newton–Raphson approach to the computation of the matrix square-root-inverse. To improve the convergence of the Denman–Beavers iterations for the simulations we consider in this work, we propose to modify the algorithm by setting

$$\mathbf{P}_0 = \frac{\Delta t^2}{4}\left(\frac{4}{\Delta t^2}\mathbf{I} + \mathbf{G}_{ij}\hat{\mathbf{A}}_i^{\mathrm{ALE}}\hat{\mathbf{A}}_j^{\mathrm{ALE}} + C_I\mathbf{G}_{ij}\mathbf{G}_{kl}\hat{\mathbf{K}}_{ik}\hat{\mathbf{K}}_{lj}\right), \tag{33}$$

$$\mathbf{Q}_0 = \mathbf{I}, \tag{34}$$

as the initial guess, and carrying out the Denman–Beavers iterations as

$$\mathbf{P}_{k+1} = \frac{1}{2}\left(\mathbf{P}_k + \mathbf{Q}_k^{-1}\right), \tag{35}$$

$$\mathbf{Q}_{k+1} = \frac{1}{2}\left(\mathbf{Q}_k + \mathbf{P}_k^{-1}\right), \tag{36}$$

where k is the iteration index. In the above equations, $\mathbf{P}_{k+1}$ is a successive approximation of the matrix square-root of $\mathbf{P}_0$ and $\mathbf{Q}_{k+1}$ is a successive approximation of the matrix square-root-inverse of $\mathbf{P}_0$. As a result, when the Denman–Beavers iteration converges, $\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}$ in Eq. (31) can be computed as $\frac{\Delta t}{2}\mathbf{Q}_{k+1}$.

Remark 3. Note that the initial guess of $\mathbf{P}_0$ is scaled by $\frac{\Delta t^2}{4}$. For the simulations considered in this work, Δt is usually small and as a result $\mathbf{P}_0$ defined in Eq. (33) is close to an identity matrix. This presents an excellent initial guess for the Denman–Beavers algorithm.

Remark 4. Defining $\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}$ as in Eq. (31) and taking the square-root-inverse using the iterative Denman–Beavers algorithm amounts to what we call the "direct" approach, which is somewhat more computationally expensive than adopting simplified expressions for $\hat{\boldsymbol{\tau}}_{\mathrm{SUPG}}$. Nevertheless, we feel the proposed methodology justifies the extra cost due to the increased accuracy and robustness it exhibits for a wide range of Reynolds and Mach numbers.

2.2.4. Discontinuity-capturing operator

Following the philosophy for designing the SUPG operator, we first define the discontinuity-capturing (DC) operator for conservation variables as

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}_{,i} \cdot \hat{\mathbf{K}}_{\mathrm{DC}}\mathbf{U}_{,i}\, d\Omega, \tag{37}$$

where $\hat{\mathbf{K}}_{\mathrm{DC}}$ is the diffusivity matrix for the DC operator. Changing variables from $\mathbf{U}$ to $\mathbf{Y}$ gives

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}_{,i} \cdot \hat{\mathbf{K}}_{\mathrm{DC}} \mathbf{A}_0 \mathbf{Y}_{,i} \, d\Omega, \tag{38}$$

which, in turn, defines the DC-operator diffusivity matrix for the pressure-primitive variables as

$$\mathbf{K}_{\mathrm{DC}} = \hat{\mathbf{K}}_{\mathrm{DC}} \mathbf{A}_0. \tag{39}$$

We assume a diagonal form of the diffusivity matrix for the conservation variables, namely,

$$\hat{\mathbf{K}}_{\mathrm{DC}} = \mathrm{diag}\left(\hat{\kappa}_{\mathrm{C}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{E}}\right), \tag{40}$$

where the diagonal entries are given by

$$\hat{\kappa}_{\mathrm{C}} = C_{\mathrm{C}} \frac{h \left|\mathrm{Res}_1\right|}{\left|\nabla U_1\right|}, \tag{41}$$

$$\hat{\kappa}_{\mathrm{M}} = C_{\mathrm{M}} \frac{h \left|\mathrm{Res}_{2:d+1}\right|}{\left|\nabla U_{2:d+1}\right|}, \tag{42}$$

$$\hat{\kappa}_{\mathrm{E}} = C_{\mathrm{E}} \frac{h \left|\mathrm{Res}_{d+2}\right|}{\left|\nabla U_{d+2}\right|}. \tag{43}$$

In the above, h is the element size, and C_{C}, C_{M}, and C_{E} are the $O(1)$ positive constants corresponding to the continuity, momentum, and energy equations, respectively. The above equations are inspired by the so-called CAU discontinuity capturing technique [83], and may also be viewed as YZβ discontinuity capturing with $\beta = 1$. Note that the definition of $\mathbf{K}_{\mathrm{DC}}$ is residual-based, and thus does not upset consistency of the formulation. The DC operator for the pressure-primitive variable formulation is now defined as

$$B_{\mathrm{DC}}\left(\mathbf{W}, \mathbf{Y}\right) = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}_{,i} \cdot \mathbf{K}_{\mathrm{DC}} \mathbf{Y}_{,i} \, d\Omega. \tag{44}$$

Remark 5. We note that the CAU discontinuity-capturing technique is an extension of the "δ_{91}" shock-capturing technique [10, 11] to unsteady flows. The CAU is residual-based in the context of unsteady problems, while "δ_{91}" was residual-based in the context of steady, inviscid flows.

2.2.5. Weak-boundary-condition operator

The idea of weak imposition of essential boundary conditions for incompressible flows was first proposed in [47]. The motivation for this development was to relax the boundary-layer resolution requirements for wall-bounded turbulent flows without sacrificing the overall solution ac-

curacy [47, 49–52]. Weakly enforced essential boundary conditions act as near-wall models for underresolved boundary-layer flows while converging to their strongly-enforced counterparts at optimal rate with mesh refinement. Here we extend the weakly enforced essential boundary condition formulation to compressible flows.

We assume that essential boundary conditions on the velocity and temperature fields are enforced on $\Gamma_{\mathrm{D}} \in \Gamma$, and the fluid domain boundary Γ is decomposed into N_{eb} surface elements each denoted by Γ^b. Let $\mathbf{W} = [q\ \mathbf{w}\ w_\theta]^{\mathrm{T}}$ be the vector of test functions with $\mathbf{w} = [w_1\ w_2\ w_3]^{\mathrm{T}}$ being the test functions for the linear-momentum balance equations. The weak-boundary-condition operator for compressible flows that is consistent with the pressure-primitive variable formulation is given by

$$
\begin{aligned}
B_{\mathrm{WBC}}(\mathbf{W},\mathbf{Y}) = &-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} \mathbf{w}\cdot(\boldsymbol{\sigma}(\mathbf{u},p)\mathbf{n})\ d\Gamma \\
&-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} (\delta\boldsymbol{\sigma}(\mathbf{w},q)\mathbf{n})\cdot(\mathbf{u}-\mathbf{g})\ d\Gamma \\
&-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}^-} \mathbf{w}\cdot\rho\left((\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{n}\right)(\mathbf{u}-\mathbf{g})\ d\Gamma \\
&+\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} \mathbf{w}\cdot\tau_\mu(\mathbf{u}-\mathbf{g})\ d\Gamma \\
&+\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} (\mathbf{w}\cdot\mathbf{n})\,\tau_\lambda(\mathbf{u}-\mathbf{g})\cdot\mathbf{n}\ d\Gamma \\
&-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} w_\theta\kappa\nabla T\cdot\mathbf{n}\ d\Gamma \\
&-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} \kappa\nabla w_\theta\cdot\mathbf{n}\,(T-T_{\mathrm{b}})\ d\Gamma \\
&-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}^-} w_\theta\rho c_{\mathrm{v}}\left((\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{n}\right)(T-T_{\mathrm{b}})\ d\Gamma \\
&+\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_{\mathrm{D}}} w_\theta\tau_\kappa(T-T_{\mathrm{b}})\ d\Gamma,
\end{aligned}
\tag{45}
$$

where $\boldsymbol{\sigma}(\mathbf{u},p) = -p\mathbf{I} + (\lambda\nabla\cdot\mathbf{u})\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}\right)$, $\delta\boldsymbol{\sigma}(\mathbf{w},q) = \rho q\mathbf{I} + (\lambda\nabla\cdot\mathbf{w})\mathbf{I} + \mu\left(\nabla\mathbf{w} + \nabla^{\mathrm{T}}\mathbf{w}\right)$, $\mathbf{g}$ is the prescribed velocity on the no-slip boundary, T_{b} is the prescribed temperature on the boundary, and Γ_{D}^- is the inflow part of Γ_{D} where $(\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{n} < 0$.

Remark 6. The first five terms on the right-hand side of Eq. (45) correspond to the weak en-

forcement of the velocity boundary conditions, while the last four terms ensure weak enforcement of temperature boundary conditions. The formulation is essentially a direct extension of weak boundary-condition enforcement for incompressible-flow and advection-diffusion equations.

Remark 7. The penalty parameters in the weak-boundary-condition operator are defined as follows: $\tau_\mu = C_I^B \mu/h_n$, $\tau_\lambda = C_I^B |\lambda|/h_n$, and $\tau_\kappa = C_I^B \kappa/h_n$. Here h_n is the element size in the wall-normal direction and C_I^B is a positive constant, which needs to be sufficiently large for the overall stability of the formulation. It emanates from an appropriate element-level trace inequality [78–80], and, for low-order elements, it is sufficient to take $4 \leq C_I^B \leq 8$. We advise the readers to avoid taking C_I^B to be too large, because in that case the penalty terms will dominate the weak-boundary-condition formulation and overshadow its variational consistency responsible for the good performance of the method.

2.2.6. Sliding-interface operator

Here we extend the sliding-interface formulation, developed for incompressible flows in [60] and applied to wind-turbine simulation in [54, 55], to compressible flows. We are motivated by applications such as gas turbines, where the sliding interfaces arise due to the need to handle mechanical and structural components that are in relative motion.

To present the method, we consider two subdomains that are in relative motion and share a sliding interface, denoted by Γ_I. We use subscripts 1 and 2 to distinguish the quantities (e.g., test and trial functions, state variables, etc.) defined on the subdomains. To weakly enforce the compatibility of the flow variables, tractions, and heat fluxes at the sliding interface, we define the following sliding-interface operator:

$$
\begin{aligned}
B_{\mathrm{SI}}(\mathbf{W},\mathbf{Y}) = &-\sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_\mathrm{I}} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \frac{1}{2} \left(\boldsymbol{\sigma}(\mathbf{u}_1, p_1)\mathbf{n}_1 - \boldsymbol{\sigma}(\mathbf{u}_2, p_2)\mathbf{n}_2\right) \, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_\mathrm{I}} \frac{1}{2} \left(\delta\boldsymbol{\sigma}(\mathbf{w}_1, q_1)\mathbf{n}_1 - \delta\boldsymbol{\sigma}(\mathbf{w}_2, q_2)\mathbf{n}_2\right) \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_\mathrm{I}^-)_1} q_1 \cdot \rho_1 \beta_{T_1} \left((\mathbf{u}_1 - \hat{\mathbf{u}}_1) \cdot \mathbf{n}_1\right) (p_1 - p_2) \, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_\mathrm{I}^-)_2} q_2 \cdot \rho_2 \beta_{T_2} \left((\mathbf{u}_2 - \hat{\mathbf{u}}_2) \cdot \mathbf{n}_2\right) (p_2 - p_1) \, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_\mathrm{I}^-)_1} \mathbf{w}_1 \cdot \rho_1 \left((\mathbf{u}_1 - \hat{\mathbf{u}}_1) \cdot \mathbf{n}_1\right) (\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_\mathrm{I}^-)_2} \mathbf{w}_2 \cdot \rho_2 \left((\mathbf{u}_2 - \hat{\mathbf{u}}_2) \cdot \mathbf{n}_2\right) (\mathbf{u}_2 - \mathbf{u}_1) \, d\Gamma
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_I} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \tau_\mu (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \\
&+ \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_I} (\mathbf{w}_1 \cdot \mathbf{n}_1 + \mathbf{w}_2 \cdot \mathbf{n}_2)\, \tau_\lambda (\mathbf{u}_1 \cdot \mathbf{n}_1 + \mathbf{u}_2 \cdot \mathbf{n}_2)\, d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_I} (w_{\theta_1} - w_{\theta_2})\, \frac{\kappa}{2} (\nabla T_1 \cdot \mathbf{n}_1 - \nabla T_2 \cdot \mathbf{n}_2)\; d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_I} \frac{\kappa}{2} (\nabla w_{\theta_1} \cdot \mathbf{n}_1 - \nabla w_{\theta_2} \cdot \mathbf{n}_2)\, (T_1 - T_2)\; d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_I^-)_1} w_{\theta_1} \rho_1 c_v\, ((\mathbf{u}_1 - \hat{\mathbf{u}}_1) \cdot \mathbf{n}_1)\, (T_1 - T_2)\; d\Gamma \\
&- \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap (\Gamma_I^-)_2} w_{\theta_2} \rho_2 c_v\, ((\mathbf{u}_2 - \hat{\mathbf{u}}_2) \cdot \mathbf{n}_2)\, (T_2 - T_1)\; d\Gamma \\
&+ \sum_{b=1}^{N_{eb}} \int_{\Gamma^b \cap \Gamma_I} (w_{\theta_1} - w_{\theta_2})\, \tau_\kappa (T_1 - T_2)\, d\Gamma,
\end{aligned} \tag{46}
$$

where $\beta_T = 1/p$, and the remaining terms are defined as in Eq. (45). For each of $i = 1$ and $i = 2$, the surface $\left(\Gamma_I^-\right)_i$ is the portion of Γ_I on which $(\mathbf{u}_i - \hat{\mathbf{u}}_i) \cdot \mathbf{n}_i < 0$. Note that $\mathbf{n}_1 = -\mathbf{n}_2$.

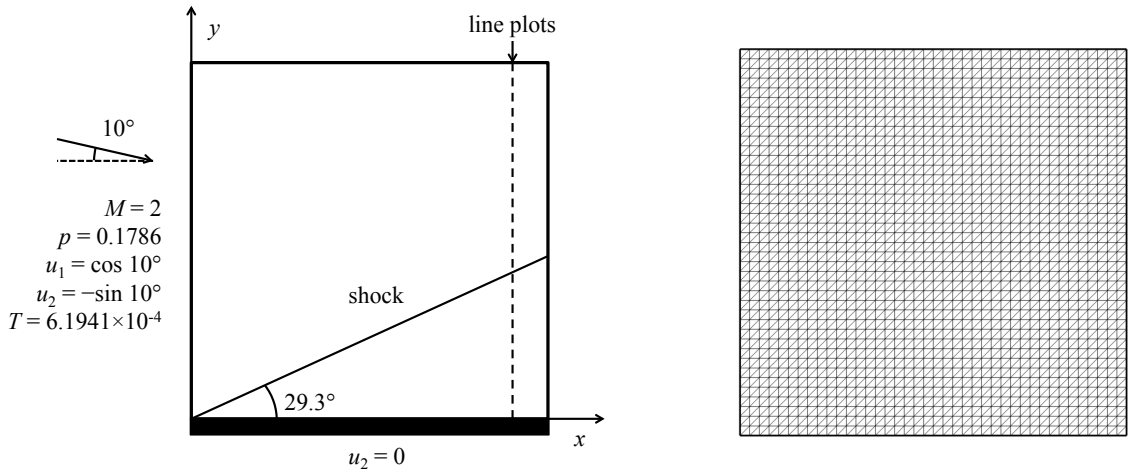
Remark 8. The above formulation is, for the most part, a direct extension of the sliding-interface formulation for incompressible-flow and advection-diffusion equations. The terms that are not present in the incompressible-flow or advection-diffusion formulations are the third and fourth terms on the right-hand side of Eq. (46). These terms provide additional convective stabilization for the pressure variable and are important for the overall stability of the sliding-interface formulation.

2.3. Semi-discrete formulation and time integration

Putting all the numerical constituents into a single framework, the 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 $\forall \mathbf{W}^h \in \mathcal{V}^h$,

$$
\begin{aligned}
&B\left(\mathbf{W}^h, \mathbf{Y}^h\right) - F\left(\mathbf{W}^h\right) \\
&+ B_{\text{SUPG}}\left(\mathbf{W}^h, \mathbf{Y}^h\right) + B_{\text{DC}}\left(\mathbf{W}^h, \mathbf{Y}^h\right) \\
&+ B_{\text{WBC}}\left(\mathbf{W}^h, \mathbf{Y}^h\right) + B_{\text{SI}}\left(\mathbf{W}^h, \mathbf{Y}^h\right) = 0,
\end{aligned} \tag{47}
$$

where $\mathcal{S}^h$ and $\mathcal{V}^h$ are the discrete counterparts of $\mathcal{S}$ and $\mathcal{V}$. To integrate the semi-discrete compressible-flow equations in time we employ the Generalized-α method, which was first introduced in [84] for structural dynamics and later extended to fluid dynamics in [85]. Generalized-α



(a) Problem setup (b) Mesh

Figure 1: Oblique shock. Problem setup, boundary conditions, and mesh.

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-diagonal-preconditioned GMRES technique [86, 87].

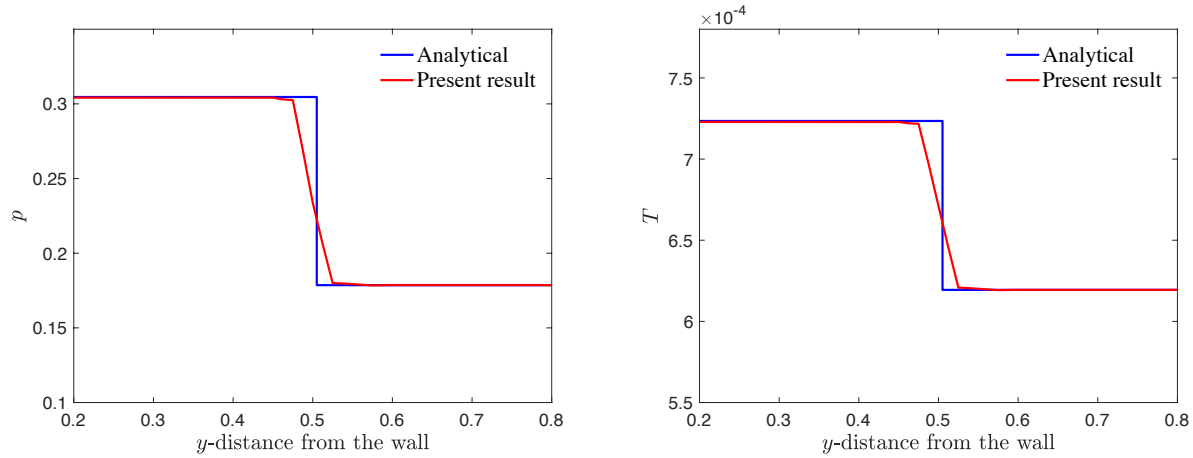
3. Numerical examples

In this section, we present a series of numerical examples that cover a wide range of Reynolds and Mach numbers to illustrate the general applicability of the proposed compressible-flow formulation. Comparison with experimental data, with computational results obtained by other researchers, and with analytical solutions are presented where applicable.

3.1. Oblique shock

In this 2D example a uniform inviscid flow at $M = 2$ impacts a wall at a 10° angle. The analytical solution of the steady Euler equations predicts formation of an oblique shock at 29.3° relative to the wall at the leading edge. The problem setup is shown in Figure 1a. The computational domain is a bi-unit square discretized using a uniform 41×41-node mesh of triangular elements shown in Figure 1b. The values of p, u_1, u_2, and T are prescribed on the left and top boundaries, and no-penetration boundary conditions are set at the bottom wall. The right boundary is left open.

In Figure 2, we plot the pressure and temperature solution along a dashed line shown in Figure 1a (at $x = 0.9$) together with the analytical result. The computed pressure and temperature profiles match their analytical counterparts very well. The shock is in the correct location and is resolved within four elements without oscillation, illustrating a good combination of accuracy and robustness of the proposed methodology.



(a) Pressure (b) Temperature

Figure 2: Oblique shock. Pressure and temperature solutions along a vertical line corresponding $x = 0.9$. Exact solution is plotted for comparison.

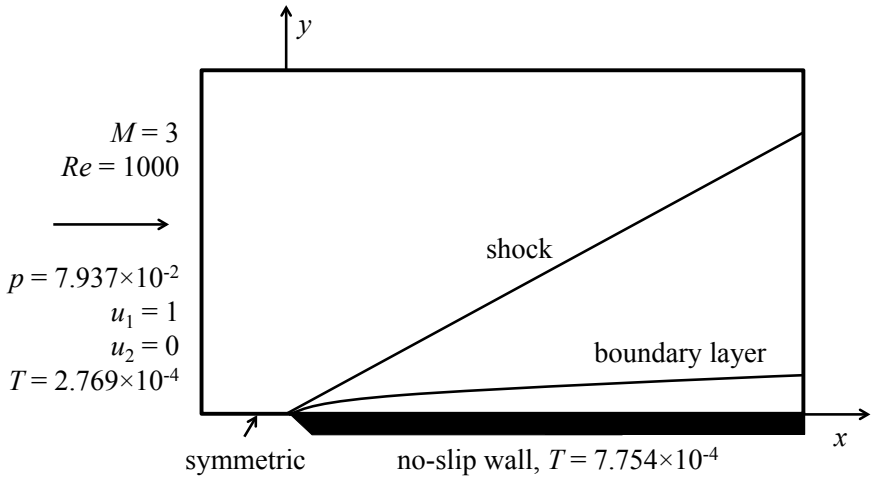


Figure 3: Supersonic flow over a flat plate. Problem setup and boundary conditions.

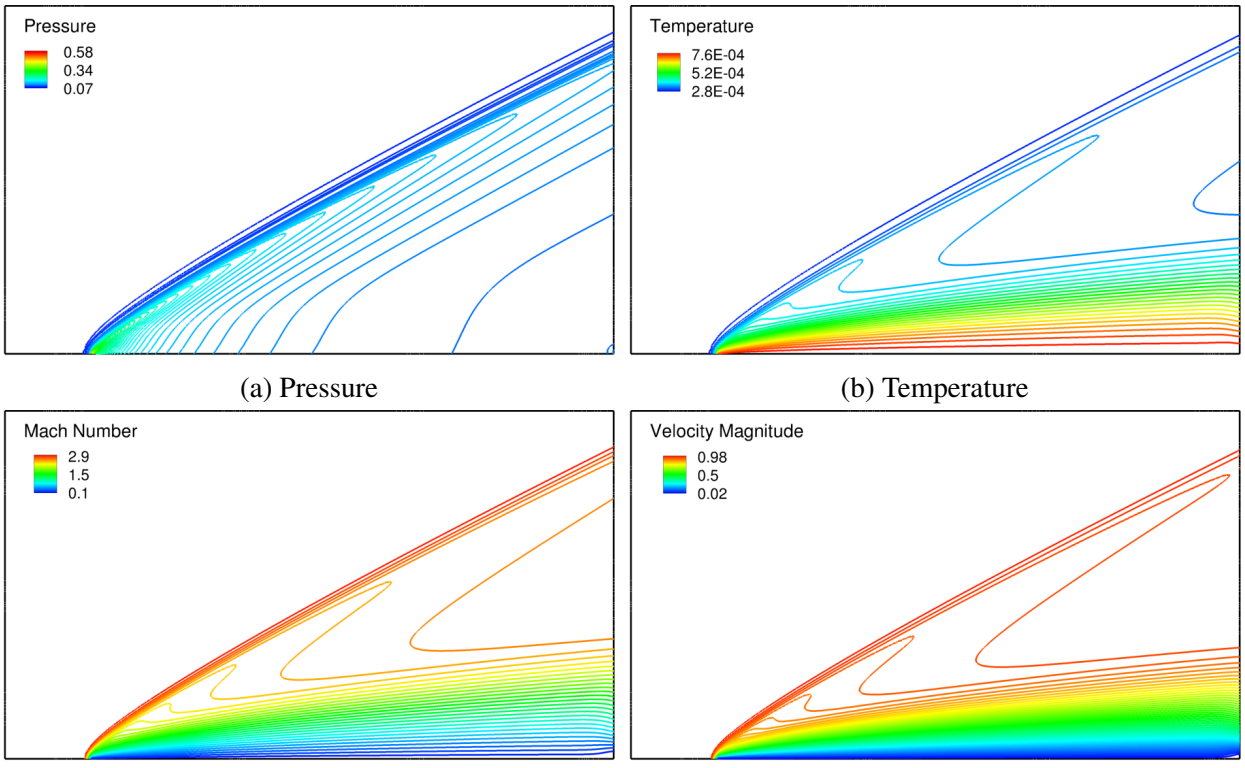
3.2. Supersonic flow over a flat plate

A 2D viscous supersonic flow over a flat plate at free-stream $Re = 1000$ and $M = 3$ is simulated. Figure 3 shows the problem setup wherein a shock and boundary layer are developed from the leading edge. The temperature-dependent viscosity is set according to Sutherland's law as

$$\mu = C_1 T^{1.5} / (T + S), \tag{48}$$

where $S = 0.0001406$ is a constant taken from literature (see, e.g., [9]), and $C_1 = 0.0906$ is a scaling factor chosen to yield the desired free-stream Reynolds number.

The computational domain is a rectangle with $-0.2 \leq x \leq 1.2$ and $0 \leq y \leq 0.8$. The wall begins at $x = 0$ and continues along the lower boundary all the way to the outflow. All the unknowns are



(a) Pressure (b) Temperature

(c) Mach number (d) Velocity magnitude

Figure 4: Supersonic flow over a flat plate. Isocontours of pressure, temperature, Mach number, and velocity magnitude.

set at the inlet and top boundaries, and the outlet is assumed to have zero traction and heat flux. On the portion of the bottom boundary prior to the wall (i.e., the "symmetric" boundary), normal velocity, tangential traction, and heat flux are all set to zero. No-slip boundary condition is enforced strongly at the solid wall. Wall temperature is prescribed as the stagnation temperature given by

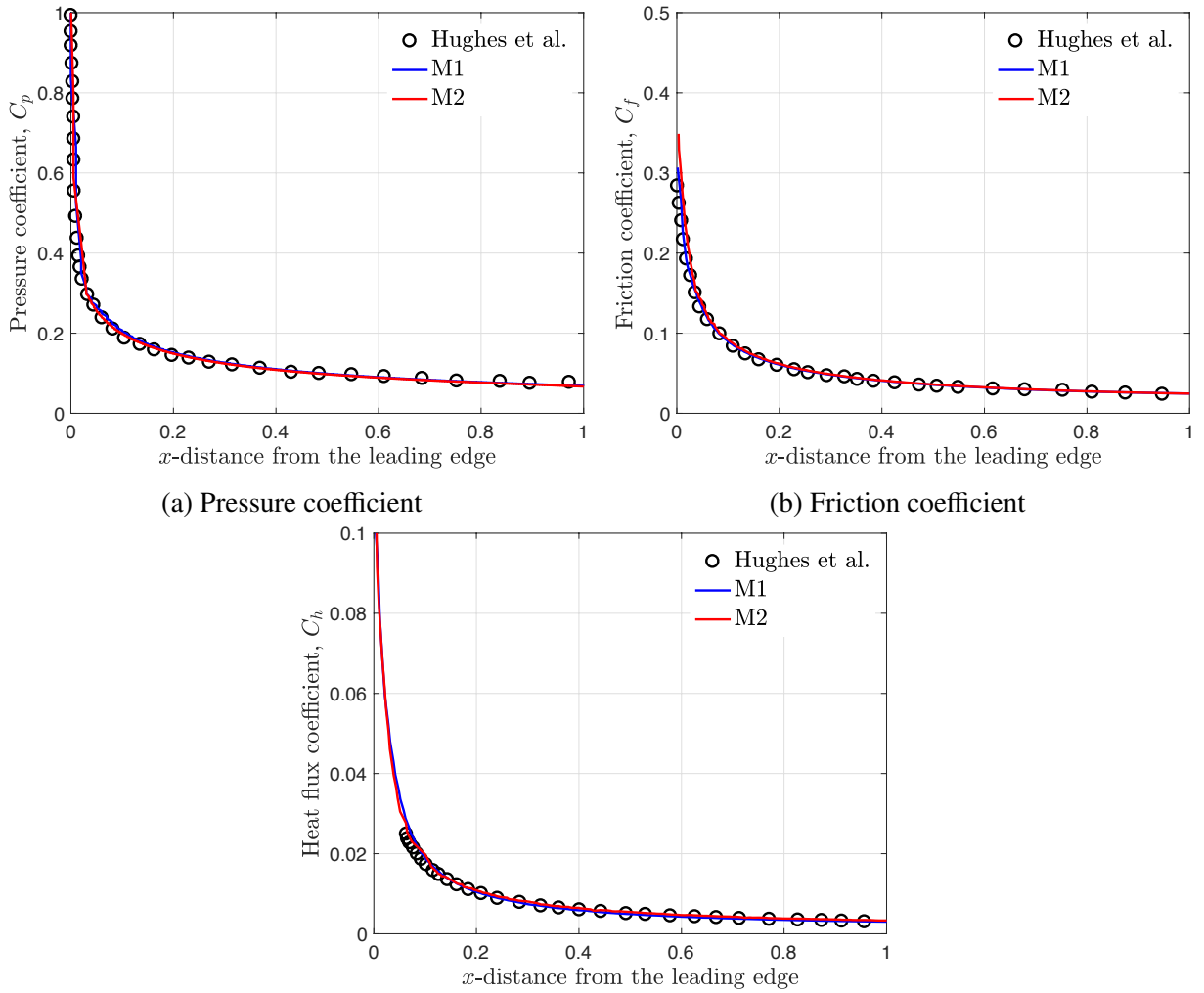
$$T_{\mathrm{w}} = T_{\infty}\left(1 + \frac{\gamma - 1}{2} M_{\infty}^2\right). \tag{49}$$

Two uniform meshes with 22,400 and 89,600 triangular elements, denoted by M1 and M2, respectively, are employed in the computations. Figure 4 shows the solution contours obtained on M1. The data is in good qualitative and quantitative agreement with the results presented in [9].

Figure 5 shows the computed pressure (C_p), friction (C_f), and heat-flux (C_h) coefficients plotted along the wall. The coefficients are defined as

$$C_p = \frac{2\,(p - p_{\infty})}{\rho_{\infty}\|\mathbf{u}_{\infty}\|^2}, \tag{50}$$

$$C_f = \frac{2\tau_{\mathrm{w}}}{\rho_{\infty}\|\mathbf{u}_{\infty}\|^2}, \tag{51}$$



(a) Pressure coefficient

(b) Friction coefficient

(c) Heat flux coefficient

Figure 5: Supersonic flow over a flat plate. Pressure (C_p), friction (C_f), and heat-flux (C_h) coefficients plotted along the wall. Numerical solutions from [9] are plotted for reference.

$$C_h = \frac{2q_{\mathrm{w}}}{\rho_\infty \|\mathbf{u}_\infty\|^3}, \tag{52}$$

where, p_∞ is the inflow static pressure, ρ_∞ is the inflow density, $\|\mathbf{u}_\infty\|$ is the magnitude of inflow velocity, τ_{w} is the wall shear stress, and q_{w} is the heat flux. Meshes M1 and M2 produce nearly identical results, suggesting the mesh is sufficiently refined to obtain accurate wall quantities in this example. Excellent agreement with the reference results of [9] is also attained.

3.3. Flow around NASA's delta wings

Flow around delta wings across a wide range of Reynolds and Mach numbers have been experimentally investigated and extensively documented by NASA researchers [88, 89]. We use this data to validate the compressible-flow formulation proposed in this work. Among the available

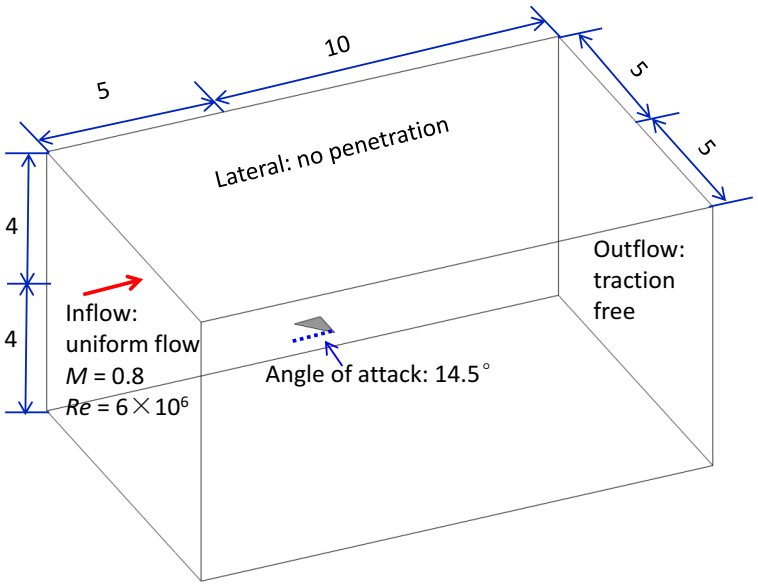


Figure 6: Flow around NASA's delta wing. Problem setup.

data, we choose a subsonic case and a supersonic case, both have a high angle of attack leading to flow separation. This example partially serves to validate the newly proposed weakly enforced boundary conditions for wall-bounded turbulent compressible flows with separation.

3.3.1. Subsonic case

Chu and Luckring [88] conducted experimental tests of the flow around delta wings with a leading edge sweep angle of 65°, and documented the pressure distributions on the delta wing surfaces. The experiment was performed for Mach number ranging from $M = 0.4$ to $M = 0.9$, and Reynolds number ranging from $Re = 6 \times 10^6$ to $Re = 36 \times 10^6$. Here we choose the case $M = 0.8$, $Re = 6 \times 10^6$, and the angle of attack of 14.5°. In [88], four types of leading edge profiles were employed. Here we choose the delta wing with a sharp leading edge. The detailed geometry description of the delta wing may be found in [88, Appendix A]. In the present computations we normalize the delta-wing geometry by scaling the chord to a non-dimensional length of $c = 1$.

The delta wing is placed in a large rectangular-box computational domain (see Figure 6). At the inflow we set $\|\mathbf{u}\| = 1$, $p = 1.1161$, and $T = 3.8713 \times 10^{-3}$, which yields $M = 0.8$. The viscosity is set to $\mu = 1.1111 \times 10^{-7}$ to obtain the desired Reynolds number based on the non-dimensional mean aerodynamic chord length of $\bar{c} = 2/3$. On the wing surface, no-slip velocity and stagnation temperature of $T = 4.3368 \times 10^{-3}$ boundary conditions are enforced weakly. At the outlet, zero traction and heat-flux boundary conditions are specified. On the lateral boundaries, no-penetration and zero tangential-traction and heat-flux boundary conditions are set.

The problem mesh is designed as follows. We first use an element size of 0.004 to discretize the delta wing surface uniformly into triangles. Starting from the surface mesh, we define a total

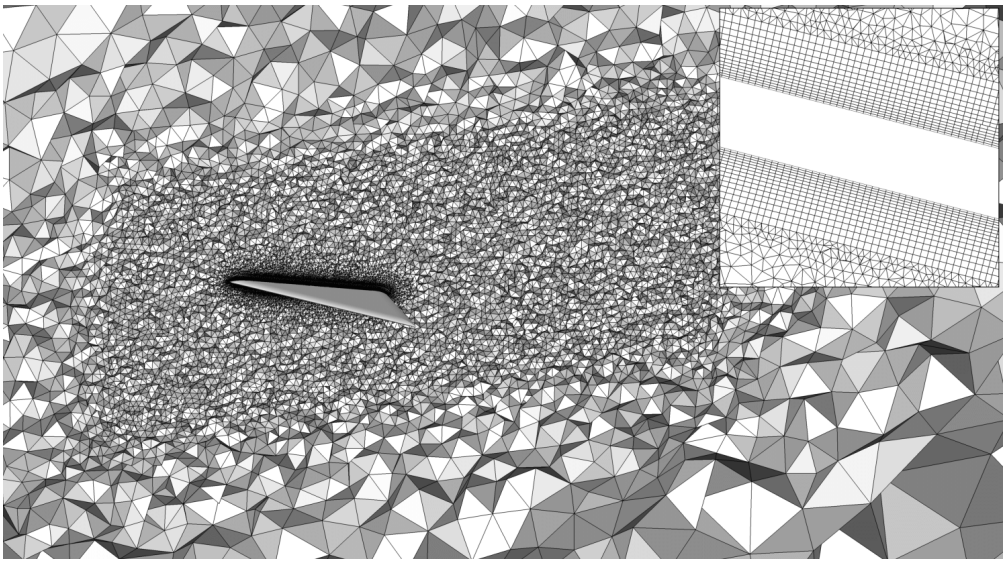


Figure 7: Subsonic flow around NASA's delta wing. Cut through the mesh interior and zoom on the boundary-layer discretization.

of 15 layers of prismatic elements, with the first layer height of 0.001 and a growth ratio of 1.1. The first layer height gives $y^+ \approx 225$. A mesh refinement zone with the element growth ratio of 1.2 and a maximum element size of 0.04 is defined near and downstream of the delta wing to better resolve the downstream flow. The remainder of the fluid domain is filled with tetrahedral elements with a maximum mesh size of 1.0. This design gives the problem mesh of 6,551,827 elements. A cut through the mesh interior is shown in Figure 7 along with the zoom-in to the boundary-layer discretization. The time step size in the computation is set to $\Delta t = 4 \times 10^{-4}$.

Figure 8 shows the highly turbulent, separated flow around the delta wing through the visualization of vortex structures (see, e.g., [90, 91] for definition). The flow separates along the leading edge and forms a region of rotational flow referred to as the "primary vortex" [89]. This highly rotational flow induces surface velocities that create regions of low pressure, as can be seen in Figure 9. These regions of low pressure are coincident with the locations of the primary vortices shown in Figure 8.

Figure 10 shows the mean pressure coefficient along the spanwise direction (η) at different chord locations (x/c). The definition of x/c and η are shown in Figure 10f. Very good agreement between the numerical results and experimental data is observed. In particular, sharp pressure gradients at $x/c = 0.6$, $x/c = 0.8$, and $x/c = 0.95$ are very well captured in the simulation. This demonstrates the superior coarse-mesh (here $y^+ \approx 225$) accuracy of weakly enforced boundary conditions in handling compressible, turbulent flow separation and reattachment.

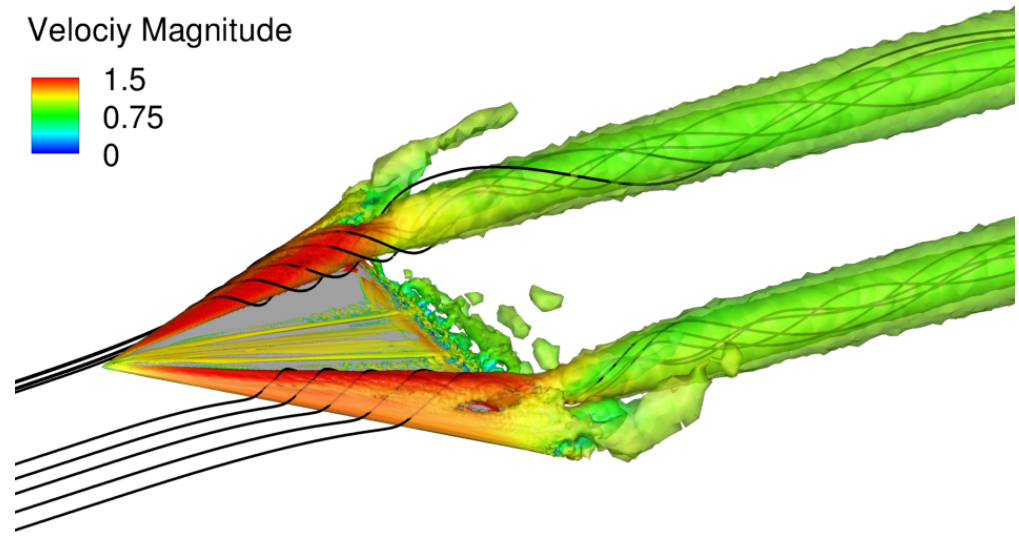
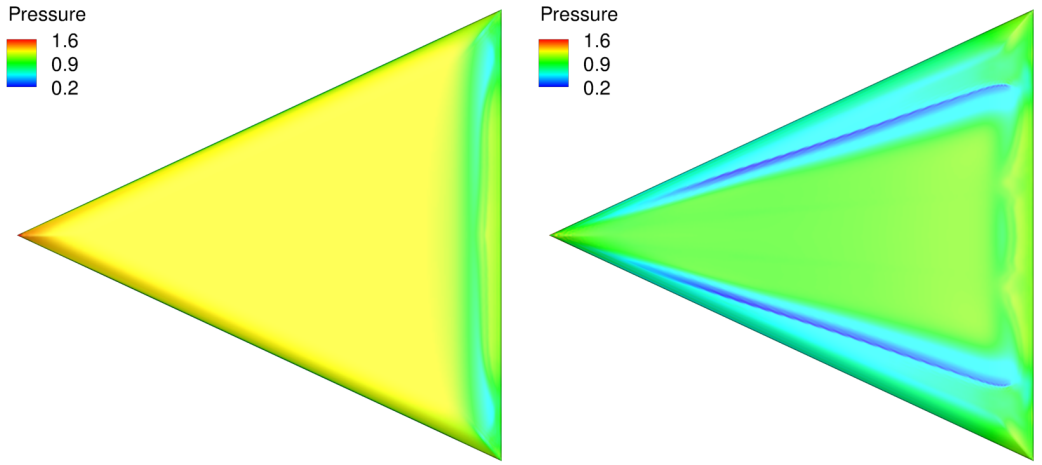


Figure 8: Subsonic flow around NASA's delta wing. Vorticity isosurfaces colored by velocity magnitude. Streamlines are also plotted in order to better illustrate the structure of the primary vortex.



(a) Lower surface (b) Upper surface

Figure 9: Subsonic flow around NASA's delta wing. Pressure distribution over the delta-wing surfaces.

3.3.2. Supersonic case

We compute the delta-wing flow in the supersonic regime, and refer the reader to [89] for the experimental investigations of this case. We chose the case at $M = 2$, $Re = 2 \times 10^6$, and the angle of attack of 12°, in which the shock-induced flow separation appears. We note that the geometry in this case is slightly different from the subsonic case. The geometry details may be found in [89, Figure 1]. As in the subsonic case, we normalize the geometry by scaling the chord length to a non-dimensional unit length.

The problem setup is similar to the subsonic case. At the inflow we set $\|\mathbf{u}\| = 1$, $p = 0.1786$, and $T = 6.1941 \times 10^{-4}$, which yields $M = 2$. On the wing surface, no-slip velocity and stagnation

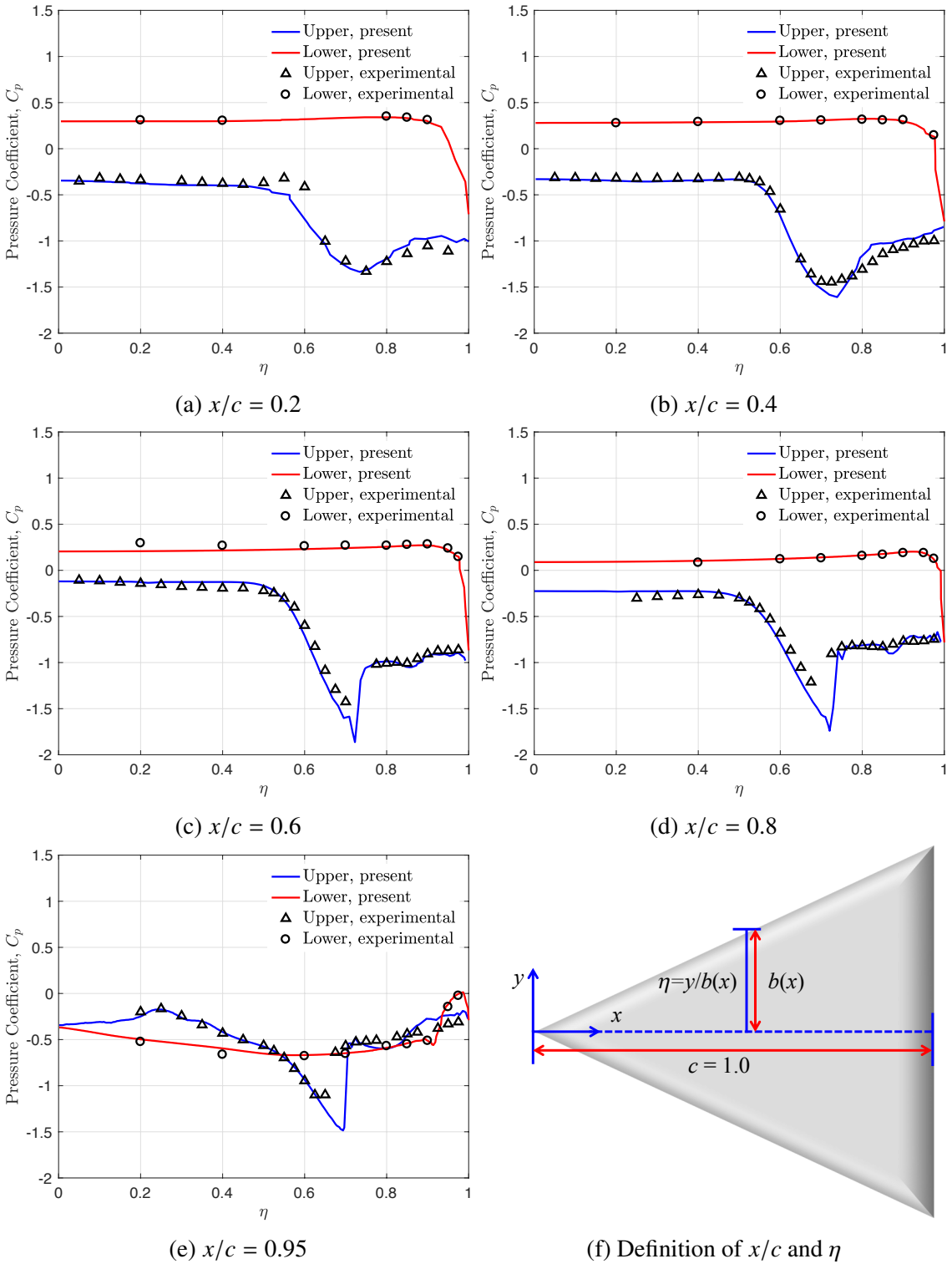


Figure 10: Subsonic flow around NASA's delta wing. Pressure coefficient plotted along the spanwise direction at different chord locations on the wing surface. Experimental data from [88] are plotted for comparison.

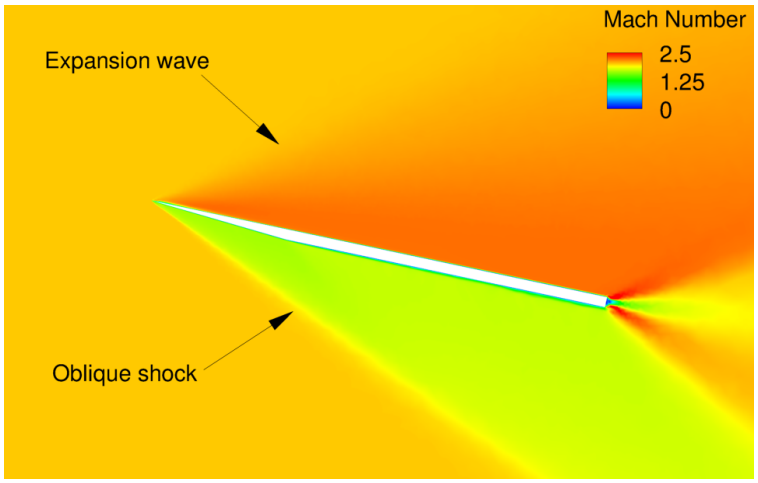


Figure 11: Supersonic flow around NASA's delta wing. Mach number contours on the vertical symmetry plane.

temperature of $T = 1.1149 \times 10^{-3}$ boundary conditions are enforced weakly. At the outlet, zero traction and heat-flux boundary conditions are specified. On the lateral boundaries, no-penetration and zero tangential-traction and heat-flux boundary conditions are set. The temperature dependent viscosity is determined by the Sutherland's law given by Eq. (48) with $C_1 = 1.6434 \times 10^{-5}$ and $S = 0.0001406$, which yields the desired inflow Reynolds number based on the non-dimensional mean aerodynamic chord length of $\bar{c} = 2/3$. A similar mesh design as in the subsonic case is employed, which yields a total of 6,149,717 elements. The time step size used in this case is 4×10^{-4}.

Figure 11 shows the Mach number contours on the vertical symmetry plane. In the figure one can clearly observe the formation of an oblique shock below the wing and the well-known "Prandtl–Meyer expansion fan" above the wing. In Figure 12, the time-averaged pressure is plotted along the $x/c = 0.96$ line on the upper surface (x/c and η are defined in the same way as in Figure 10f) together with the experimental measurements, showing very good agreement between the two. In particular, a sudden drop in the pressure due to flow separation is accurately captured in the simulation.

3.4. Turbulent flow around a sphere at Re = 10,000

With this example we pursue a dual purpose: a. To demonstrate the effectiveness of our formulation in the nearly incompressible, low-Mach-number regime, thus extending the applicability of our methodology to cover a broad range of flows; b. To validate the sliding-interface formulation by solving the same problem with and without the sliding interface, and expecting to obtain solutions with the same level of accuracy.

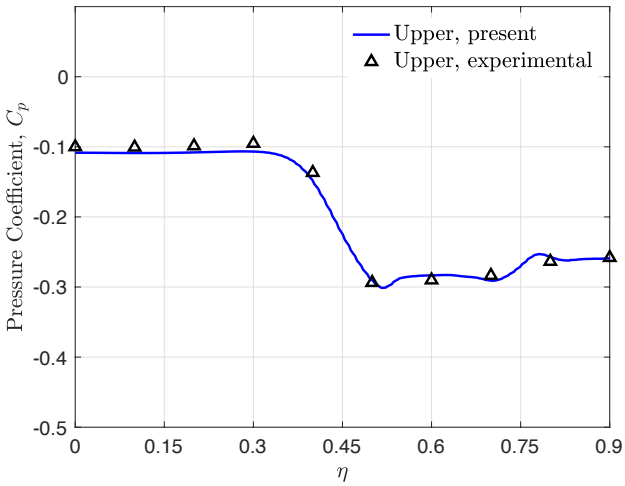


Figure 12: Supersonic flow around NASA's delta wing. Pressure coefficient on the upper surface along the spanwise direction at $x/c = 0.96$. Experimental data from [89] are plotted for comparison.

3.4.1. Problem setup

Turbulent flow around a sphere at $Re = 10{,}000$ and $M = 0.1$ is considered. The inflow consists of uniform flow with unit speed, $p = 71.4286$, and $T = 0.2478$. The sphere diameter and fluid viscosity are set to $D = 1$ and $\mu = 0.0001$, respectively, yielding the desired free-stream Reynolds number. No-slip conditions and stagnation temperature of $T = 0.2588$ are imposed weakly on the sphere. As mentioned earlier, the problem is solved with and without the sliding interface. The problem setup, including the sliding interface, is shown in Figure 13. In the case of sliding-interface computation, the inner domain is assumed to rotate with angular velocity of one radian per non-dimensional time unit with zero-velocity boundary condition weakly enforced on the sphere surface. This setup is expected to produce the same solution as the stationary-domain case without the sliding interface.

We use a similar mesh design as in [92]. We first discretize the sphere into uniform triangles. Seven layers of prismatic elements with a growth ratio of 1.2 are then constructed, resulting in the first element size of $y^+ = 2.36$ in non-dimensional wall units. Subsequently, the sliding interface is introduced, and two cylindrical refinement zones around and downstream of the sphere are constructed to better capture the wake flow. The remainder of the fluid domain is filled with tetrahedral elements. The mesh statistics are shown in Table 1. A cut through the mesh is shown in Figure 14 to illustrate the discretization in the fluid-domain interior. The mesh with the sliding interface has a total of 6,378,930 elements, while the mesh without the sliding interface has a total of 6,395,378 elements. In both computations the time step is set to $\Delta t = 0.005$, which yields a CFL number of $O(1)$.

Table 1: Turbulent flow around a sphere. Mesh statistics.

First layer height	Sliding interface	Refinement cylinder 1	Refinement cylinder 2	Outer box
0.004	0.04	0.04	$0.16/\sqrt{2}$	0.8

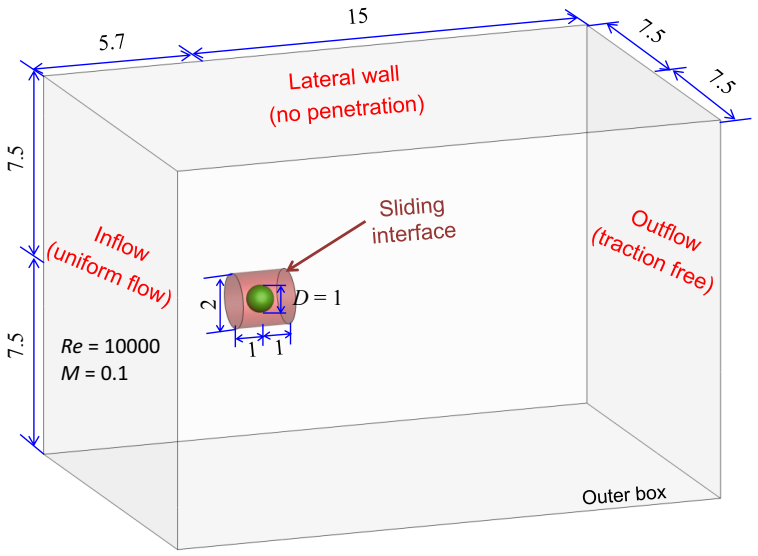


Figure 13: Turbulent flow around a sphere. Problem setup.

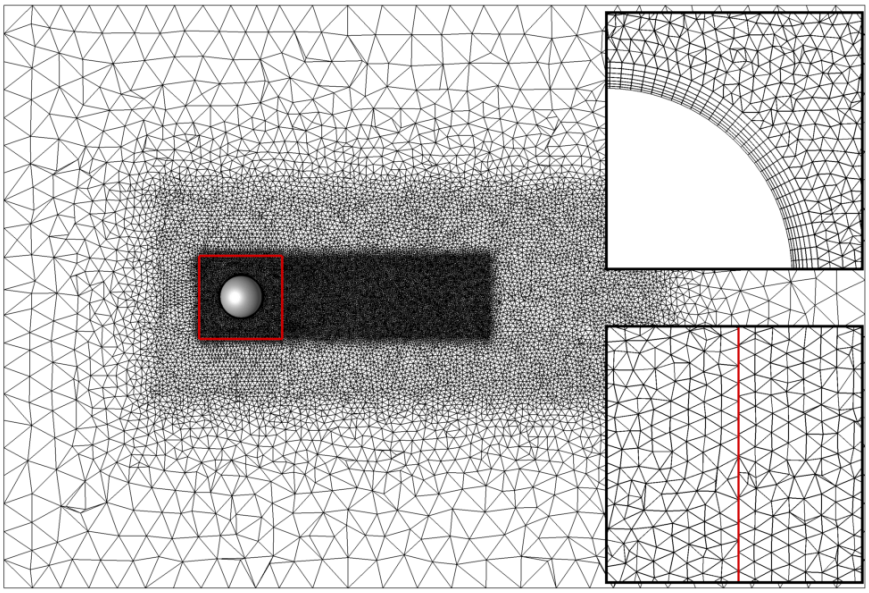


Figure 14: Turbulent flow around a sphere. Cut through the mesh and zoom on the boundary layer and sliding interface (marked in red).

3.4.2. Simulation results

Figure 15 shows instantaneous vortical structures, which illustrate several features of this flow including a thin, laminar boundary layer in the front of the sphere, flow separation at the sphere

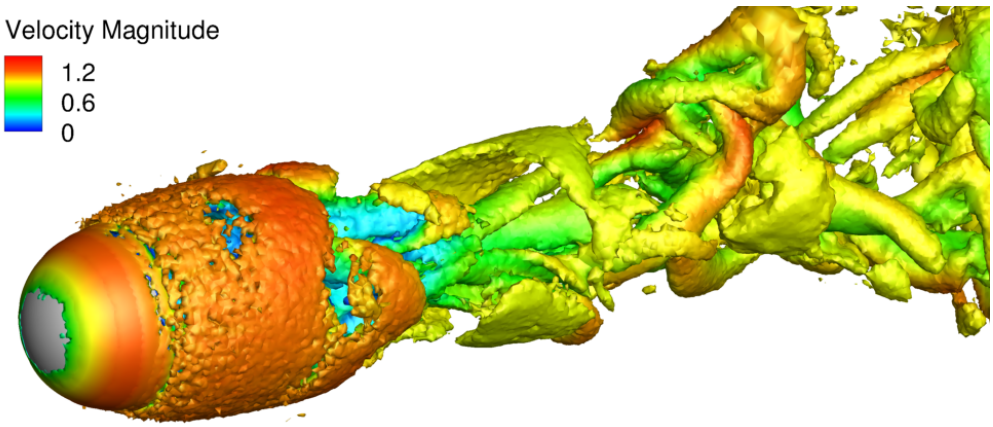


Figure 15: Turbulent flow around a sphere. Vorticity isosurfaces colored by velocity magnitude.

apex, and complex flow in the sphere wake.

We compare our results, in terms of the key quantities of interest, for the simulations with and without the sliding interface with data obtained from the Direct Numerical Simulation (DNS) reported in [93] and experimental data from [94, 95]. Note that the DNS results in [93] are obtained by solving the Navier–Stokes equations of incompressible flows. As a result, a perfect match between our results and the DNS data is not expected. Table 2 presents a comparison of the time-averaged $\overline{C}_{\mathrm{D}}$, the drag coefficient, $\overline{L}/D$, the non-dimensional length of the recirculation bubble, and $\overline{C}_{pb}$, the pressure coefficient at an azimuthal angle of 180°, which corresponds to the rearmost point on the sphere in the flow direction. The distribution of mean pressure and skin-friction coefficient as function of the azimuthal angle is shown in Figure 16. Here the mean skin-friction coefficient is defined as $C_f = \bar{\tau}_w/(\rho_\infty \|\mathbf{u}_\infty\|^2 Re^{0.5})$, where τ_w is the wall friction.

In all comparisons we observe a very good agreement between the results with or without the sliding interface. Among all the quantities of interest, we emphasize that the predictions of the non-dimensional length of the circulation bubble are very close in these two cases. This shows that the sliding-interface formulation gives good compatibility of the kinematic and thermodynamic variables as well as tractions and heat fluxes across the non-matching interface.

Remark 9. In the computation of the skin-friction coefficient, the traction vector on the sphere surface is computed using a conservation definition [51, 96], which takes on the following form:

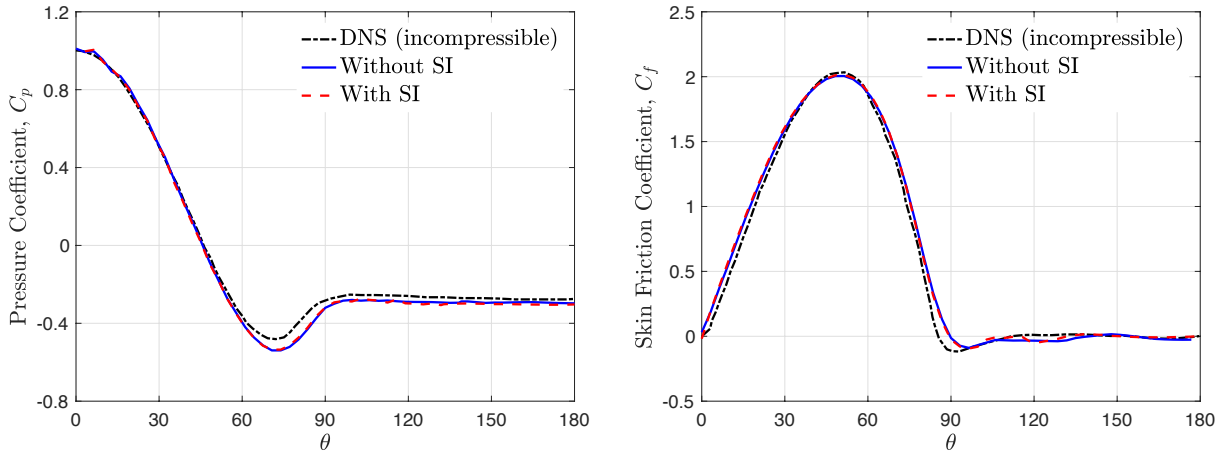
$$\mathbf{t} = -\boldsymbol{\sigma}(\mathbf{u}, p)\mathbf{n} - \rho\{(\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n}\}_- (\mathbf{u} - \mathbf{g}) + \tau_\mu(\mathbf{u} - \mathbf{g}) + \tau_\lambda ((\mathbf{u} - \mathbf{g}) \cdot \mathbf{n})\, \mathbf{n}\,, \tag{53}$$

where $\{\ \cdot\ \}_-$ denotes the negative part of the bracketed quantity.

Remark 10. The data in Table 2 indicates that a slightly higher drag is predicted in the $M = 0.1$ compressible-flow computations relative to incompressible DNS, which corresponds to $M = 0$.

Table 2: Turbulent flow around a sphere. Comparison of time-averaged quantities of interest. Simulation results with and without the sliding interface (SI) are reported. The drag coefficient for $Re \approx 10{,}500$ and $M = 0.11$ obtained experimentally in [94, 95] is also provided for comparison.

	$\overline{C}_D$	$\overline{L}/D$	$\overline{C}_{p_b}$
Results without SI	0.409	1.423	-0.297
Results with SI	0.411	1.421	-0.301
DNS [93] (incompressible)	0.402	1.657	-0.272
Experimental data [94, 95]	0.413	–	–



(a) Pressure coefficient

(b) Skin-friction coefficient

Figure 16: Turbulent flow around a sphere. Time-averaged pressure and skin-friction coefficient plotted as a function of azimuthal angle.

This is not surprising since the data in [94] shows that for a flow in the subsonic regime at $Re > 200$ the sphere drag increases with the increasing Mach number. The higher drag value emanates from higher suction pressure predicted in the compressible-flow simulation (see Figure 16).

4. Flow inside a gas turbine stage

In this section, we simulate the flow inside a gas turbine stage. This stage contains a row of stator blades and a row of rotor blades. The flow accelerates inside the stator-blade channels and then enters the rotor-blade channels.

4.1. Problem setup

We model the fluid domain by a stationary subdomain containing the stator and a rotational subdomain containing the rotor. The two domains are coupled using the sliding-interface formulation. A uniform axial inflow with velocity of 82.3 m/s, pressure of 2,012,790 Pa and temperature of 1669.78 K is applied at the inlet boundary. The traction-free boundary condition is set at the outlet.

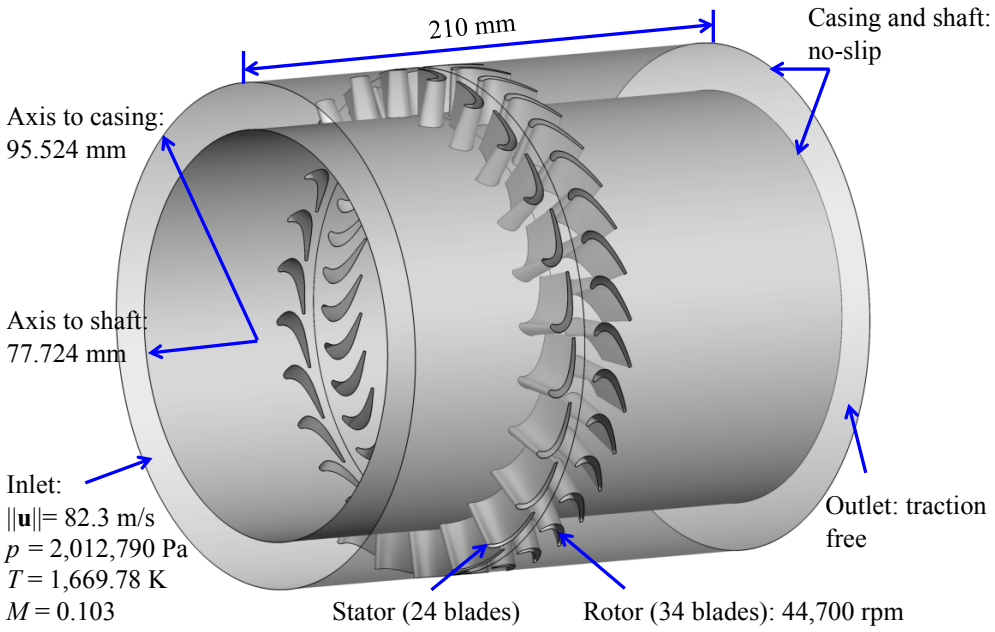


Figure 17: Flow inside a gas turbine stage. Problem setup, geometry and dimensions.

The no-slip velocity boundary conditions and the stagnation temperature of $T = 1,673.15$ K are enforced weakly on the rotor and stator blades. On the shaft and casing surfaces, the zero normal heat flux condition is applied and the no-slip velocity boundary conditions are enforced strongly. The dynamic viscosity is set to $\mu = 5.551 \times 10^{-5}$ kg/(m·s). The gas-turbine geometry, dimensions, and problem-setup details are shown in Figure 17. We note that this is a smaller gas turbine design, similar to the one used as part of a turboshaft for Black Hawk or Apache helicopters.

Gas turbine engines are generally optimized to operate at nearly constant speed with fixed blade geometry for the design operating conditions. When the operating conditions of the engine changes, the flow incidence angles may not be optimal with the blade geometry, resulting in reduced off-design performance. Articulating the pitch angle of gas turbine blades can improve performance by maintaining flow incidence angles within the optimum range. To optimize the pitch angles of the stator and rotor blades and perform a series of simulations with different pitch angles requires a capability to change the blade pitch angles parametrically. Following the idea of the interactive geometry modeling platform proposed in [97, 98], we build a parametric design tool based on Rhinoceros 3D [99] and Grasshopper [100]. The user interface of the design tool is shown in Figure 18. Note that we have two input parameters, "Rotor pitching" (with blue background) and "Stator pitching" (with red background). By changing these two input parameters, we can directly change the pitching angles of the rotor blades (blue) and stator blades (red) parametrically in the 3D gas turbine stage model.

After the geometry is parametrically designed, we mesh the fluid domain using tetrahedral

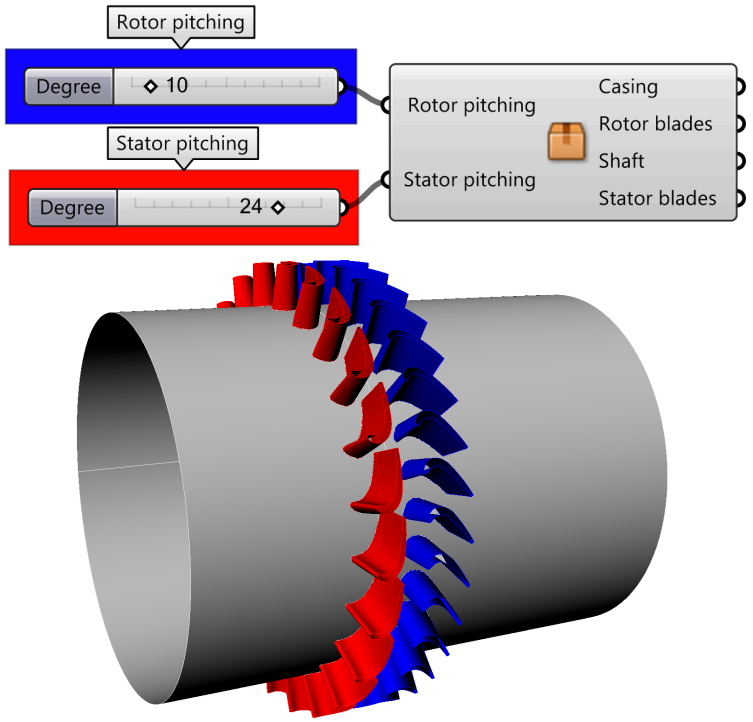


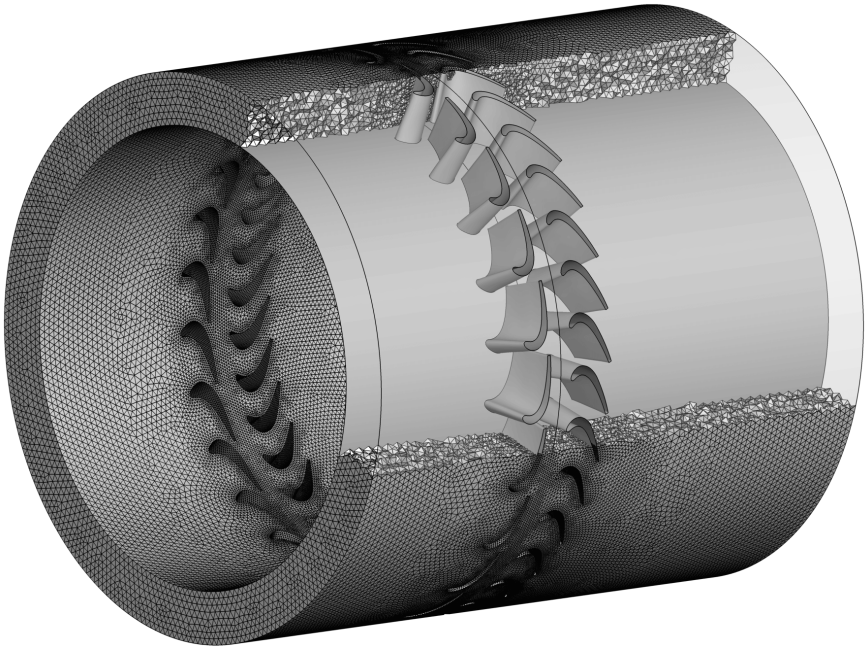
Figure 18: Flow inside a gas turbine stage. Parametric geometry model of the gas turbine stage built by the Rhino–Grasshopper-based design platform.

elements. The mesh is locally refined in the region near the blades, as shown in Figure 19. The sliding-interface meshes are also shown in Figure 19.

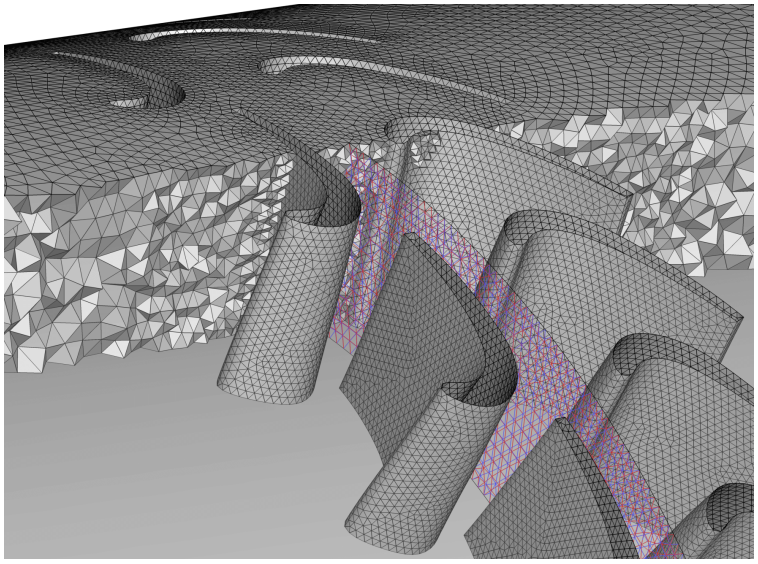
4.2. Simulation results

For a gas turbine operating under off-design conditions, flow phenomena such as flow separation may occur. These will increase the flow losses and thus negatively influence the gas-turbine performance. Pitching the blade angle to match the flow angle can help improve the performance. To illustrate this, we first pick an off-design case and simulate the flow. We then look at the difference between the flow angle and the blade angle, and pitch the rotor blades to match these angles accordingly. Finally, we run a simulation on the new design to see how much the efficiency can be improved by articulating the rotor blades. In the simulations, the time step size of $\Delta t = 3 \times 10^{-7}$ s is employed, which yields a CFL number of around 2. For the case before pitching, the mesh contains 9,454,324 tetrahedral elements. Figure 20 shows a visualization of the highly turbulent 3D flow structures.

We expect the flow variables to be continuous across the sliding interface. To show that, we make a cylindrical slice cutting through the fluid domain, from inlet to outlet. Contours of flow speed, pressure, temperature, and Mach number are shown in Figure 21, and appear to be continuous across the sliding interface.



(a) Mesh of the domain interior



(b) Zoom on the sliding interface

Figure 19: Flow inside a gas turbine stage. Problem mesh.

The relative velocity magnitude inside the rotor passages is shown in Figure 22. Before pitching the rotor blades, since the flow inlet angle is smaller than the blade inlet angle, the flow is not fully attached on the pressure surface (see Figure 22a). By pitching the rotor to decrease the blade inlet angle, we are able to recover a better flow field. The flow is fully attached to the blades, on both the pressure and suction surfaces, as shown in Figure 22b.

Gas-turbine performance may be assessed by computing the adiabatic efficiency of the turbine

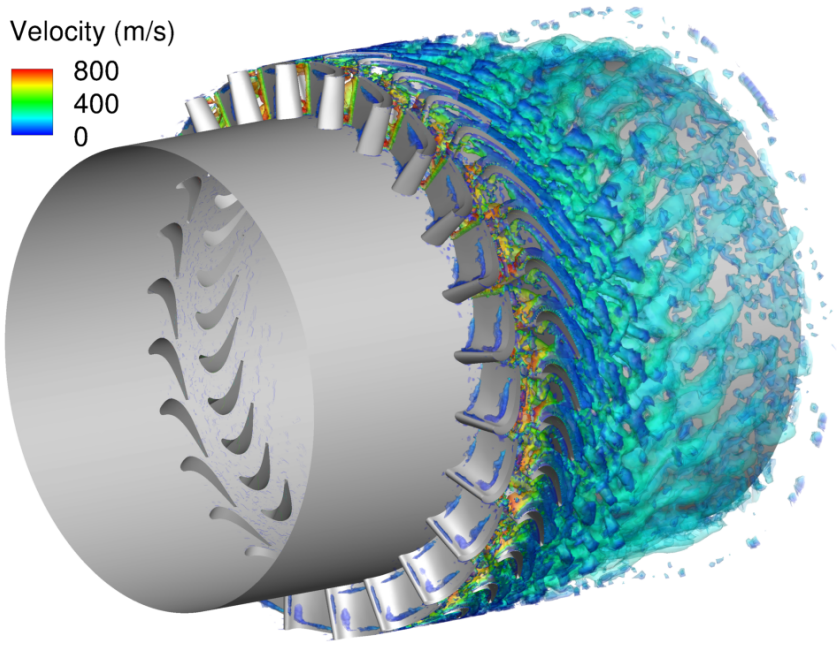
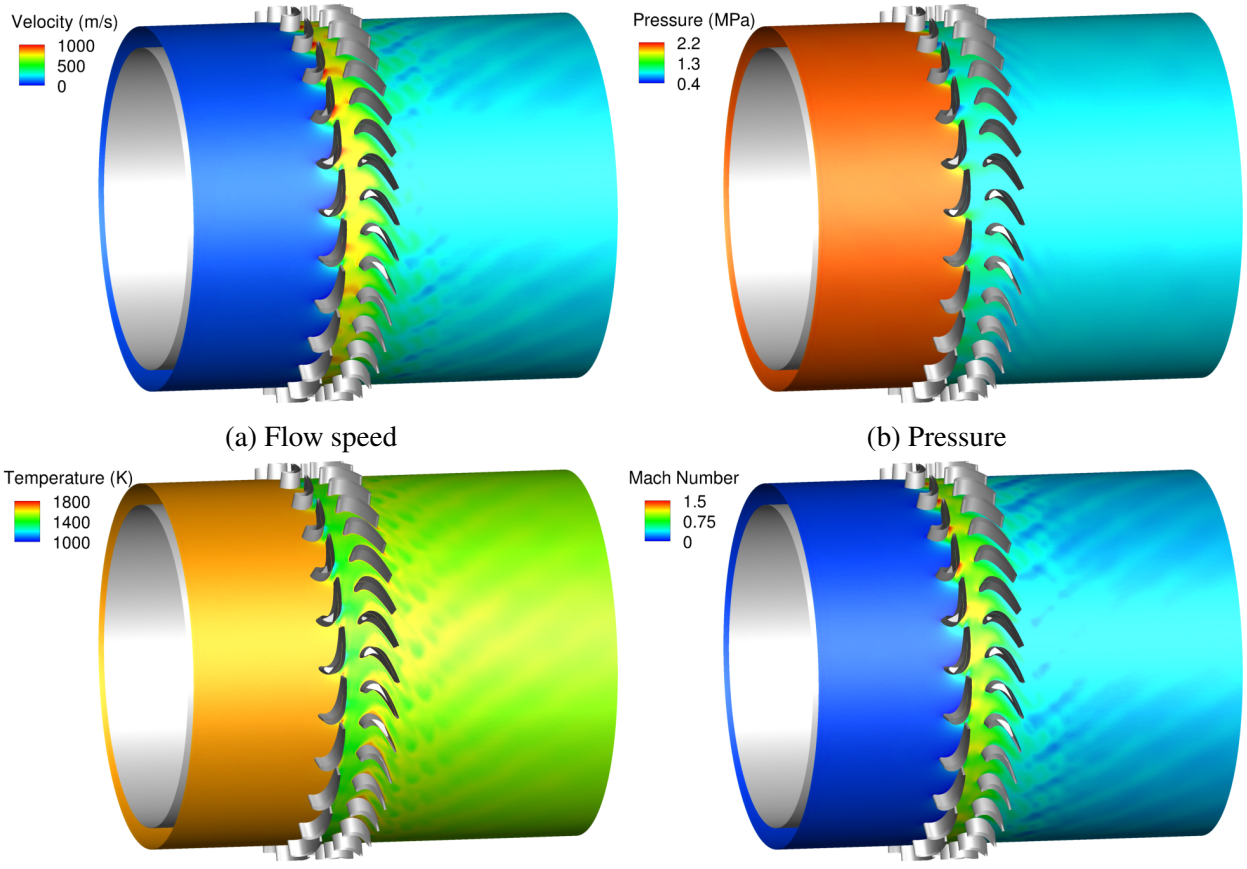


Figure 20: Flow inside a gas turbine stage. Vorticity colored by velocity magnitude.



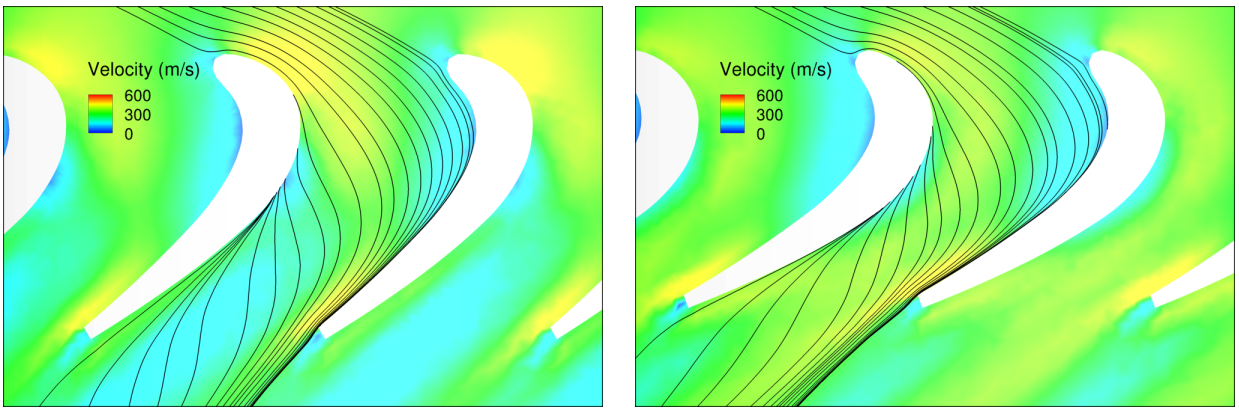
(a) Flow speed

(b) Pressure

(c) Temperature

(d) Mach number

Figure 21: Flow inside a gas turbine stage. Flow variables on a cylindrical cut, which appear to be continuous across the sliding interface.



(a) Before pitching (b) After pitching

Figure 22: Flow inside a gas turbine stage. Relative velocity magnitude inside the rotor passages. Flow streamlines are also plotted to show improvements in the flow characteristics after pitching the rotor blades.

stage (see [101]). The adiabatic efficiency is defined as the ratio between the actual and isentropic (ideal) power output. With subscripts 0 and 2 denoting quantities at the stator inlet and rotor exit, respectively, the adiabatic efficiency η_{ad} is given by

$$\eta_{\text{ad}} = \frac{1 - \dfrac{T_2}{T_0}}{1 - \left(\dfrac{p_2}{p_0}\right)^{\frac{\gamma-1}{\gamma}}}. \tag{54}$$

Using the above formula in the postprocessing of our simulation results, we find that before pitching $\eta_{\text{ad}} = 0.468$ while after pitching $\eta_{\text{ad}} = 0.494$, which presents a 5% increase and shows that rotor-blade pitching can help improve gas-turbine efficiency under off-design conditions.

5. Conclusions

In this paper we developed a novel stabilized formulation for compressible flows on moving domains using the ALE approach. While stabilized methods for compressible flows have a long history of development, the current work presents the following innovations relative to the earlier research:

- A direct approach to the construction of the parameter $\boldsymbol{\tau}_{\text{SUPG}}$, which consistently accounts for the stabilization of all the modes in the compressible-flow equations system.
- A new formulation for weakly enforced essential boundary conditions.
- A new formulation for sliding interfaces.

We extensively validated our formulation and demonstrated its robustness using test cases spanning a wide range of Reynolds and Mach numbers. We also demonstrated excellent accuracy of wall quantities, such as pressure and skin friction, that can be achieved on relatively coarse boundary-layer meshes using weak imposition of essential boundary conditions. The aerodynamic analysis of a gas turbine stage indicates that the methods developed in this work may be used directly to support engineering design at industrial scale.

Acknowledgments

This work was supported by the ARO Grant No. W911NF-14-1-0296. The HPC resources that have contributed to the research results reported in this paper were provided by the Texas Advanced Computing Center (TACC) at the University of Texas at Austin. This support is gratefully acknowledged.

Appendix A.

For the Navier–Stokes equations with reduced energy equation (12), the matrices corresponding to pressure-primitive variables are as follows:

The matrix $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{Y}}$ is given by

$$\mathbf{A}_0 = \begin{bmatrix} \rho\beta_T & 0 & 0 & 0 & -\rho\alpha_p \\ \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T e & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.1}$$

where $\beta_T = 1/p$, $\alpha_p = 1/T$.

It's inverse $\mathbf{A}_0^{-1} = \mathbf{Y}_{,\mathbf{U}}$ is given by

$$\mathbf{A}_0^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{\alpha_p}{\rho\beta_T c_v} \\ -\dfrac{u_1}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0 \\ -\dfrac{u_2}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{u_3}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ -\dfrac{T}{\rho} & 0 & 0 & 0 & \dfrac{1}{\rho c_v} \end{bmatrix}. \tag{A.2}$$

The Euler–Jacobian matrices are given by

$$\mathbf{A}_1^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_1^2 & 2\rho u_1 & 0 & 0 & -\rho\alpha_p u_1^2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ \rho\beta_T e u_1 & \rho e & 0 & 0 & 0 \end{bmatrix}, \tag{A.3}$$

$$\mathbf{A}_2^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_2^2 & 0 & 2\rho u_2 & 0 & -\rho\alpha_p u_2^2 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ \rho\beta_T e u_2 & 0 & \rho e & 0 & 0 \end{bmatrix}, \tag{A.4}$$

$$\mathbf{A}_3^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ \rho\beta_T u_3^2 & 0 & 0 & 2\rho u_3 & -\rho\alpha_p u_3^2 \\ \rho\beta_T e u_3 & 0 & 0 & \rho e & 0 \end{bmatrix}, \tag{A.5}$$

$$\mathbf{A}_1^{p} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.6}$$

$$\mathbf{A}_2^{p} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.7}$$

$$\mathbf{A}_3^p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.8}$$

$$\mathbf{A}_1^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p-\tau_{11} & -\tau_{12} & -\tau_{13} & 0 \end{bmatrix}. \tag{A.9}$$

$$\mathbf{A}_2^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\tau_{21} & p-\tau_{22} & -\tau_{23} & 0 \end{bmatrix}. \tag{A.10}$$

$$\mathbf{A}_3^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\tau_{31} & -\tau_{32} & p-\tau_{33} & 0 \end{bmatrix}. \tag{A.11}$$

The diffusivity matrices are given by

$$\mathbf{K}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}, \tag{A.12}$$

$$\mathbf{K}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.13}$$

$$\mathbf{K}_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.14}$$

$$\mathbf{K}_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.15}$$

$$\mathbf{K}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 2\mu + \lambda & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}, \tag{A.16}$$

$$\mathbf{K}_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.17}$$

$$\mathbf{K}_{31} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.18}$$

$$\mathbf{K}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.19}$$

$$\mathbf{K}_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 2\mu + \lambda & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}. \tag{A.20}$$

The matrices for the conservation variables may be obtained from the corresponding matrices for the pressure-primitive variables using the following transformations: $\hat{\mathbf{A}}_i = \mathbf{A}_i\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{\text{adv}\backslash p} = \mathbf{A}_i^{\text{adv}\backslash p}\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{p} = \mathbf{A}_i^{p}\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{\text{sp}} = \mathbf{A}_i^{\text{sp}}\mathbf{A}_0^{-1}$, and $\hat{\mathbf{K}}_{ij} = \mathbf{K}_{ij}\mathbf{A}_0^{-1}$

References

[1] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[2] T. E. Tezduyar and T. J. R. Hughes. Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. NASA Technical Report NASA-CR-204772, NASA, 1982.

[3] T. E. Tezduyar and T. J. R. Hughes. Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In *Proceedings of AIAA 21st Aerospace Sciences Meeting*, AIAA Paper 83-0125, Reno, Nevada, 1983.

[4] T. J. R. Hughes and T. E. Tezduyar. Finite element methods for first-order hyperbolic sys-

tems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] G. J. Le Beau and T. E. Tezduyar. Finite element computation of compressible flows with the SUPG formulation. In *Advances in Finite Element Analysis in Fluid Dynamics*, FED-Vol.123, pages 21–27, New York, 1991. ASME.

[11] 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.

[12] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[13] 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.

[14] 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.

[15] 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.

[16] T. E. Tezduyar, S. K. Aliabadi, M. Behr, and S. Mittal. Massively parallel finite element simulation of compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 119:157–177, 1994.

[17] G. P. Wren, S. E. Ray, S. K. Aliabadi, and T. E. Tezduyar. Space–time finite element computation of compressible flows between moving components. *International Journal for Numerical Methods in Fluids*, 21:981–991, 1995.

[18] G. P. Wren, S. E. Ray, S. K. Aliabadi, and T. E. Tezduyar. Simulation of flow problems with moving mechanical components, fluid–structure interactions and two-fluid interfaces. *International Journal for Numerical Methods in Fluids*, 24:1433–1448, 1997.

[19] S. E. Ray, G. P. Wren, and T. E. Tezduyar. Parallel implementations of a finite element formulation for fluid–structure interactions in interior flows. *Parallel Computing*, 23:1279–1292, 1997.

[20] S. E. Ray and T. E. Tezduyar. Fluid–object interactions in interior ballistics. *Computer Methods in Applied Mechanics and Engineering*, 190:363–372, 2000.

[21] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.

[22] Z. Johan, K. K. Mathur, S. L. Johnsson, and T. J. R. Hughes. An efficient communications strategy for finite element methods on the Connection Machine CM-5 system. *Computer Methods in Applied Mechanics and Engineering*, 113:363–387, 1994.

[23] Z. Johan, K. K. Mathur, S. L. Johnsson, and T. J. R. Hughes. Scalability of finite element applications on distributed-memory parallel computers. *Computer Methods in Applied Mechanics and Engineering*, 119:61–72, 1994.

[24] 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.

[25] S. Mittal and T. Tezduyar. A unified finite element formulation for compressible and incompressible flows using augumented conservation variables. *Computer Methods in Applied Mechanics and Engineering*, 161:229–243, 1998.

[26] 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.

[27] G. Hauke. Simple sabilizing matrices for the computation of compressible flows in primitive variables. *Computer Methods in Applied Mechanics and Engineering*, 190:6881–6893,

2001.

[28] T. E. Tezduyar. Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004 (CD-ROM)*, Jyvaskyla, Finland, 2004.

[29] 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.

[30] 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.

[31] T. E. Tezduyar, M. Senga, and D. Vicker. Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. *Computational Mechanics*, 38:469–481, 2006.

[32] T. E. Tezduyar and M. Senga. SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing. *Computers & Fluids*, 36:147–159, 2007.

[33] 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.

[34] 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.

[35] G. Hauke. *A unified approach to compressible and incompressible flows and a new entropy-consistent formulation of the k-ϵ model*. PhD thesis, Stanford University, 1995.

[36] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[37] F. Rispoli, A. Corsini, and T. E. Tezduyar. Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD). *Computers & Fluids*, 36:121–126, 2007.

[38] Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. YZβ discontinuity-capturing for advection-dominated processes with application to arterial drug delivery. *International Journal for Numerical Methods in Fluids*, 54:593–608, 2007.

[39] F. Rispoli, R. Saavedra, A. Corsini, and T. E. Tezduyar. Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing. *International Journal for Numerical Methods in Fluids*, 54:695–706, 2007.

[40] F. Rispoli, R. Saavedra, F. Menichini, and T. E. Tezduyar. Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and YZβ shock-capturing. *Journal of Applied Mechanics*, 76:021209, 2009.

[41] F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T. E. Tezduyar. Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing. *Computational Mechanics*, 55:1201–1209, 2015.

[42] T. J. R. Hughes, G. Scovazzi, and T. E. Tezduyar. Stabilized methods for compressible flows. *Journal of Scientific Computing*, 43:343–368, 2010.

[43] 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.

[44] S. M. Rifai, Z. Johan, W.-P. Wang, J.-P. Grisval, T. J. R. Hughes, and R. M. Ferencz. Multiphysics simulation of flow-induced vibrations and aeroelasticity on parallel computing platforms. *Computer Methods in Applied Mechanics and Engineering*, 174(3):393–417, 1999.

[45] S. M. Rifai, J. C. Buell, Z. Johan, and T. J. R. Hughes. Automotive design applications of fluid flow simulation on parallel computing platforms. *Computer Methods in Applied Mechanics and Engineering*, 184(2):449–466, 2000.

[46] A. Masud. Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows. *Computational Mechanics*, 38:430–439, 2006.

[47] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.

[48] 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.

[49] 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.

[50] 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.

[51] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[52] 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.

[53] 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.

[54] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] K. Takizawa and T. E. Tezduyar. Multiscale space–time fluid–structure interaction techniques. *Computational Mechanics*, 48:247–267, 2011.

[63] K. Takizawa and T. E. Tezduyar. Space–time fluid–structure interaction methods. *Mathematical Models and Methods in Applied Sciences*, 22(supp02):1230001, 2012.

[64] K. Takizawa, D. Montes, S. McIntyre, and T. E. Tezduyar. Space–time VMS methods for

modeling of incompressible flows at high Reynolds numbers. *Mathematical Models and Methods in Applied Sciences*, 23:223–248, 2013.

[65] 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.

[66] K. Takizawa, T. E. Tezduyar, and N. Kostov. Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV. *Computational Mechanics*, 54:213–233, 2014.

[67] 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.

[68] 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.

[69] K. Takizawa, T. E. Tezduyar, and T. Kuraishi. Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. *Mathematical Models and Methods in Applied Sciences*, 25:2227–2255, 2015.

[70] 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.

[71] K. Takizawa, T. E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi. Computational thermo-fluid analysis of a disk brake. *Computational Mechanics*, 57:965–977, 2016.

[72] K. Takizawa, T. E. Tezduyar, and H. Hattori. Computational analysis of flow-driven string dynamics in turbomachinery. *Computers & Fluids*, 142:109–117, 2017.

[73] K. Takizawa, T. E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori. Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA). *Computers & Fluids*, 142:15–20, 2017.

[74] 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.

[75] K. Takizawa, T. E. Tezduyar, and T. Terahara. Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA). *Computers & Fluids*, 141:191–200, 2016.

[76] 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*, published online, DOI:

10.1016/j.compfluid.2016.11.012, November 2016.

[77] 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.

[78] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Sweden, 1987.

[79] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, Berlin, 2002.

[80] A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*. Springer, Berlin, 2004.

[81] E. D. Denman and A. N. Beavers. The matrix sign function and computations in systems. *Applied mathematics and Computation*, 2(1):63–94, 1976.

[82] A. E. Tejada-Martínez, I. Akkerman, and Y. Bazilevs. Large-eddy simulation of shallow water Langmuir turbulence using isogeometric analysis and the residual–based variational multiscale method. *Journal of Applied Mechanics*, 79(1):010909, 2012.

[83] 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.

[84] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60:371–75, 1993.

[85] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[86] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[87] 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.

[88] J. Chu and J. M. Luckring. Experimental surface pressure data obtained on 65 delta wing across Reynolds number and Mach number ranges, volume 1 – sharp leading edge. NASA Technical Report NASA-TM-4645-Vol-1, NASA, 1996.

[89] D. S. Miller and R. M. Wood. Lee–side flows over delta wings at supersonic speeds. NASA Technical Report NASA-TP-2430, NASA, 1985.

[90] 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.

[91] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[92] Y. Bazilevs, J. Yan, M. de Stadler, and S. Sarkar. Computation of the flow over a sphere at *Re* = 3700: A comparison of uniform and turbulent inflow conditions. *Journal of Applied Mechanics*, 81(12):121003, 2014.

[93] I. Rodriguez, O. Lehmkuhl, R. Borrell, and A. Oliva. Flow dynamics in the turbulent wake of a sphere at sub–critical Reynolds numbers. *Computers & Fluids*, 80:233–243, 2013.

[94] A. B. Bailey and J. Hiatt. Sphere drag coefficients for a broad range of Mach and Reynolds numbers. *AIAA Journal*, 10(11):1436–1440, 1972.

[95] C. Wieselsberger. Wietere feststellungen über die gesetze des flüssigkeits–und luft widerstandes. *Zeitschrift für Physik*, 23:219–224, 1922.

[96] E. H. van Brummelen, K. G. van der Zee, V. V. Garg, and S. Prudhomme. Flux evaluation in primal and dual boundary-coupled problems. *Journal of Applied Mechanics*, 79:010904, 2011.

[97] M.-C. Hsu, C. Wang, A. J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers and Mathematics with Applications*, 70:1481–1500, 2015.

[98] A. J. Herrema, N. M. Wiese, C. N. Darling, B. Ganapathysubramanian, A. Krishnamurthy, and M.-C. Hsu. A framework for parametric design optimization using isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:944–965, 2017.

[99] Rhinoceros. http://www.rhino3d.com/. [Accessed 5 February 2016].

[100] Grasshopper. http://www.grasshopper3d.com/. [Accessed 5 February 2016].

[101] M. P. Boyce. *Gas turbine engineering handbook*. Elsevier, 2011.