AIAA AVIATION Forum
June 25-29, 2018, Atlanta, Georgia
2018 Multidisciplinary Analysis and Optimization Conference

10.2514/6.2018-3111

Simulation and Adjoint-based Design for Variable Density Incompressible Flows with Heat Transfer

Thomas D. Economon[*]

Bosch Research and Technology Center, Sunnyvale, CA, 94085, U.S.A.

This article details the development and implementation of an incompressible solver for simulation and design in variable density incompressible flows with heat transfer. In the low-Mach approximation of the Navier-Stokes equations, density can vary as a function of transported scalars, and in this case, density varies with temperature from a coupled energy equation. These governing equations are spatially discretized using a finite volume method on unstructured grids and solved in a coupled manner with a custom preconditioning approach. The implementation is within the SU2 suite for multiphysics simulation and design, and it has been algorithmically differentiated to construct a discrete adjoint for efficient sensitivity analysis. Results demonstrating the primal solver on a set of standard verification and validation cases and adjoint-based shape optimization are presented.

I. Introduction

In this article, we pursue the development and implementation of a solver for variable density incompressible flows with heat transfer. Variable density, low-speed flows are of interest for many applications, such as natural (buoyancy-driven) or forced convection problems, environmental flows, fire simulations, or for reacting flows, such as combustion simulations. In these situations, the Mach number can be very small, but the effects of heat transfer and the accompanying variations in density remain important.

One approach for this regime is to apply the fully compressible form of the Navier-Stokes equations for conservation of mass, momentum, and energy. Unfortunately, it is well known that the equations become very stiff at low Mach numbers, resulting in poor convergence behavior for density-based, compressible codes, and the numerical methods applied typically suffer from accuracy issues due to artificial dissipation that is poorly scaled at small Mach numbers (related to disparate scales of convection/acoustics). Preconditioning approaches for the compressible Navier-Stokes equations at low speeds can be a remedy, and they have been successfully demonstrated in literature by many authors.[1–8] These approaches can be an ideal choice for flows with mixed high and low Mach numbers. However, they carry more complexity than necessary for purely low-speed flows, which could lead to convergence or performance issues or more restrictions on the numerics.

On the other hand, the flow can be treated as incompressible. In order to include heat transfer effects in incompressible flows, the energy equation, or a temperature evolution equation, must be solved in addition to the continuity and momentum equations for the fluid. The specific coupling of the energy equation will depend on the situation. For constant density fluids, the energy equation can be solved with a one-way coupling, essentially as a passive scalar, or be two-way coupled through the Boussinesq approximation for problems with suitably small temperature variations. However, for some problems, large density variations are critical even at very small Mach numbers, such as in reacting flows, and a more elaborate model is necessary. Here, the low-Mach number formulation of the equations is an attractive choice.[9–11]

The appeal of the low-Mach Navier-Stokes equations is the ability to treat incompressible fluids that feature large density variations while avoiding the complexity of the fully compressible form of the Navier-Stokes equations. Density is decoupled from pressure and determined from an equation of state that is a function of transported scalars, such as temperature. This decoupling of the thermodynamic pressure removes the acoustics from the equations. Practically speaking, the low-Mach approximation, arrived at

[*] Senior Research Scientist, Multiphysics Modeling and Simulation, AIAA Senior Member.

by asymptotic analysis,[10] results in a simplified form of the energy equation when compared to the fully compressible governing equations. Therefore, there are implications both in terms of the form of the equations themselves and also for numerical implementation and efficiency.

When numerically solving the incompressible governing equations, the main difficulty is the coupling of pressure and velocity. The majority of authors in the literature treat the numerical implementation of the problem with a traditional pressure-projection approach for incompressible flows.[12] A different, and less common, approach is to apply the artificial compressibility method of Chorin,[13] which is an appealing choice for transforming an existing density-based, compressible code into one capable of solving incompressible problems. The trick to the artificial compressibility approach is the introduction of an artificial equation of state that establishes a pressure-velocity coupling by transforming the continuity equation from a density equation to a pressure equation. The result is that the set of fully coupled equations can then be relaxed using much of the existing infrastructure from a density-based, compressible code. While many have authors have considered the artificial compressibility approach for constant density flows, few authors have published works in applying artificial compressibility for variable density flows. Notable works on the topic are by Riedel[14] for stiff reacting flows in 2D, as well as Shapiro and Drikakis[15, 16] on the development of characteristic-based schemes for the artificial compressibility method for variable density flows.

In this article, the primary contribution is the development of a custom preconditioning for the low-Mach Navier-Stokes equations that enables efficient numerical solution in a coupled approach. The proposed preconditioning can be viewed in two ways. First, it could be seen as a generalization of the artificial compressibility method for treating variable density incompressible flows with heat transfer. The method could also be viewed as a simplification of the preconditioning approach developed by Weiss and Smith,[6] who sought to unify artificial compressibility and low-speed preconditioning approaches into one method for all speeds. By limiting the application to incompressible flows alone in this article, instead of mixed high and low Mach numbers, the proposed preconditioning scheme takes a slightly simpler form. It is important to note, however, that the artificial compressibility approach and the aforementioned preconditioning methods are closely related, which has been alluded to repeatedly in the literature.[1, 6, 17]

The preconditioning approach is implemented to solve the governing equations in the context of a general, unstructured finite volume code with vertex-based schemes, and it allows for solution of the incompressible Euler, laminar Navier-Stokes (constant or variable density), or Reynolds-averaged Navier-Stokes (RANS) equations within the same solver framework. It is implemented within the SU2 software suite.[18–20] SU2 is an open-source collection of software tools written in C++ and Python for performing multiphysics simulation and design. Motivating factors for this particular development direction include the opportunity to leverage an existing artificial compressibility solver within SU2 as a starting point,[21, 22] the potential for performing time-accurate calculations in the future (combining the preconditioning with dual time-stepping), and the availability of tools for adjoint-based design in SU2.

Adjoint methods for use in gradient-based optimal shape design have come a long way since early pioneering efforts in incompressible flows and airfoil design.[23, 24] Today, these methods are applied to large-scale problems of interest in the aircraft, automotive, and other engineering industries, and their effectiveness for efficient design optimization has been well documented in a rich body of literature over the past several decades. Thanks to collaboration across the open-source community, SU2 contains multiple implementations of the adjoint methodology with supporting tools for design all within the same codebase, including a continuous adjoint and a discrete adjoint constructed via algorithmic differentiation (AD) of the code.

A secondary contribution of this article is a verification and validation (V&V) of the preconditioning approach as implemented in SU2. A set of common laminar and RANS V&V cases are simulated, and the results are compared to analytical solutions, well-established codes, and experiment in order to build a high degree of confidence in the implementation. This article can be considered a reference to build upon for future users and developers of the incompressible solver in SU2.

This paper is organized as follows. Section II covers the governing equations and presents the development of the preconditioning approach. The numerical implementation of the equations is explained in Section III. Then, in Section IV, a series of V&V cases are detailed. Adjoint-based shape design is also demonstrated for a heated cylinder in cross-flow. Finally, in Section V, we will summarize the major conclusions of the article.

II. Modeling

In this section, the governing equations for variable density, incompressible flow are described, including details on the different fluid models that are available. For completeness, the general approach to forming an adjoint problem from the governing equations, or primal, is also presented.

A. Governing Equations

We are concerned with incompressible, viscous fluid flows involving heat transfer governed by the low-Mach Navier-Stokes equations. The governing equations can be expressed in differential form on a domain $\Omega \subset \mathbb{R}^3$ with a disconnected boundary that is divided into a solid wall S, an inlet boundary Γ_{in}, and an outlet boundary Γ_{out}, as the following:

$$\begin{cases} R(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) - Q = 0, & \text{in } \Omega, \\ \bar{v} = 0, & \text{on } S, \\ T = T_S, & \text{on } S, \\ (W)_{\bar{v}} = W_{in}, & \text{on } \Gamma_{in}, \\ (W)_{\bar{P}} = W_{out}, & \text{on } \Gamma_{out}, \end{cases} \tag{1}$$

where the conservative variables are $U = \{\rho, \rho\bar{v}, \rho c_p T\}^{\mathsf{T}}$. The second line of (1) represents the no-slip condition at a fixed, solid wall, the third line represents an isothermal condition at the wall (a heat flux condition is also possible), and the final two lines represent inlet and outlet boundary conditions on which the inlet velocity and outlet pressure can be imposed, respectively. Q is a generic source term to be discussed later. The convective and viscous fluxes can be written as

$$\bar{F}^c(U) = \left\{ \begin{array}{c} \rho\bar{v} \\ \rho\bar{v} \otimes \bar{v} + \bar{\bar{I}}p \\ \rho c_p \, T\bar{v} \end{array} \right\}, \quad \bar{F}^v(U, \nabla U) = \left\{ \begin{array}{c} \cdot \\ \bar{\bar{\tau}} \\ \kappa \nabla T \end{array} \right\}, \tag{2}$$

where ρ is the fluid density, $\bar{v} = \{u, v, w\}^{\mathsf{T}} \in \mathbb{R}^3$ is the flow speed in Cartesian system of reference, p is the dynamic pressure, $\bar{\bar{\tau}}$ is the viscous stress tensor, c_p is the specific heat at constant pressure, T is the temperature, κ is the thermal conductivity, μ is the viscosity, and the viscous stress tensor can be expressed in vector notation as

$$\bar{\bar{\tau}} = \mu \left(\nabla\bar{v} + \nabla\bar{v}^T \right) - \mu \frac{2}{3} \bar{\bar{I}} \left(\nabla \cdot \bar{v} \right). \tag{3}$$

In the low-Mach approximation of the equations, the pressure is decomposed into thermodynamic and dynamic components. In Eqn. 2, p is interpreted as the dynamic pressure in the governing equations, and p_o is the thermodynamic pressure, which is constant in space. In this article, we will assume the thermodynamic pressure is also constant in time, but it must be computed if not.[11] The system in (1) is closed with an equation of state for the density as a function of temperature alone. Assuming an ideal gas with a ratio of specific heats γ and gas constant R, one can determine the density from $\rho = p_o/RT$. We assume a constant Prandtl number Pr such that the conductivity varies with viscosity as $\kappa = \mu c_p/Pr$. It is also possible to solve additional equations as part of the system in (1) for updating the density or the transport coefficients, such as adding equations for multiple species in reacting flows, but this is not discussed here. The topic of complex fluid modeling within the present framework is left for future work.

For laminar flows, μ is simply the dynamic viscosity μ_{dyn}, which can be constant or assumed to satisfy Sutherland's law[25] (a function of temperature alone), and Pr is the dynamic Prandtl number Pr_d. For turbulent flows, we solve the Reynolds-averaged Navier-Stokes (RANS) equations. In accord with the standard approach to turbulence modeling based upon the Boussinesq hypothesis,[26] which states that the effect of turbulence can be represented as an increased viscosity, the viscosity in Eqn. 3 is divided into dynamic and turbulent components, or μ_{dyn} and μ_{tur}, respectively. Therefore, the effective viscosity in Eqn. 3 becomes

$$\mu = \mu_{dyn} + \mu_{tur}. \tag{4}$$

Similarly, the thermal conductivity in the energy equation becomes an effective thermal conductivity written as

$$\kappa = \frac{\mu_{dyn} \, c_p}{Pr_d} + \frac{\mu_{tur} \, c_p}{Pr_t}, \tag{5}$$

where we have introduced a turbulent Prandtl number Pr_t. The turbulent viscosity μ_{tur} is obtained from a suitable turbulence model involving the mean flow state U and a set of new variables for the turbulence. The Shear Stress Transport (SST) model of Menter[27] and the Spalart-Allmaras (S-A)[28] model are two of the most common and widely used turbulence models and will be used in this work.

B. Preconditioning Approach

In this section, the approach of Weiss and Smith[6] is followed in order to develop a preconditioning strategy for solving the low-Mach equations with a coupled method. Note that the main difference in the following development from previous work is that it starts from the low-Mach equations rather than the fully compressible Navier-Stokes equations. The equation of state for density is assumed to be a function of both pressure and temperature during the manipulations, or $\rho = \rho(p, T)$, to arrive at general expressions that can be simplified later.

We will work in terms of primitive quantities $V = \{p, \bar{v}, T\}^{\mathsf{T}}$, which is a natural choice for an incompressible solver, so we first transform our governing equations from conservative to primitive variables with transformation matrix M:

$$M\frac{\partial V}{\partial t} + \nabla \cdot \bar{F}^c(V) - \nabla \cdot \bar{F}^v(V, \nabla V) - Q = 0, \tag{6}$$

where

$$M = \frac{\partial U}{\partial V} = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & 0 & \rho_T u \\ \rho_p v & 0 & \rho & 0 & \rho_T v \\ \rho_p w & 0 & 0 & \rho & \rho_T w \\ \rho_p c_p T & 0 & 0 & 0 & \rho_T c_p T + \rho c_p \end{bmatrix} \tag{7}$$

transforms the conservative variable vector to the primitive vector and $\rho_p = \frac{\partial \rho}{\partial p}$ and $\rho_T = \frac{\partial \rho}{\partial T}$ are the partial derivatives of the equation of state with respect to pressure and temperature, respectively. We next transform the equations into non-conservative form through premultiplication of an additional transformation matrix K, which can be written as

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ -c_p T & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

to give

$$KM\frac{\partial V}{\partial t} + K\,\nabla \cdot \bar{F}^c(V) - K\,\nabla \cdot \bar{F}^v(V, \nabla V) - KQ = 0. \tag{9}$$

It is important to note the difference in the matrix K here as compared to that of Weiss and Smith (Eqn. 3 in the original article[6]). Due to the simplifications to the energy equation inherent in the low-Mach formulation of the equations, the bottom row of K contains fewer entries, and this simplification carries through the remainder of the manipulations. Ultimately, this will lead to a slightly simpler preconditioning overall. Inspecting KM, we see

$$KM = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ 0 & \rho & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & \rho c_p \end{bmatrix}, \tag{10}$$

and from here, we form the preconditioning approach. In incompressible flow, $\rho_p = 0$, and since we have limited ourselves to operating in the incompressible regime by construction, we will replace it with a value proportional to the magnitude of the velocity in the domain, which will control the eigenvalues of the system.

Therefore, we form the preconditioning matrix Γ_{nc} to replace KM as

$$\Gamma_{nc} = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 & \rho_T \\ 0 & \rho & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & \rho c_p \end{bmatrix}, \tag{11}$$

where β is our preconditioning factor related to the maximum velocity in the domain. To arrive at the final form of the preconditioned equations in conservation form, we introduce Γ_{nc} and premultiply by K^{-1} to give

$$R(V) = \Gamma \frac{\partial V}{\partial t} + \nabla \cdot \bar{F}^c(V) - \nabla \cdot \bar{F}^v(V, \nabla V) - Q = 0, \tag{12}$$

where

$$\Gamma = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 & \rho_T \\ \frac{u}{\beta^2} & \rho & 0 & 0 & \rho_T u \\ \frac{v}{\beta^2} & 0 & \rho & 0 & \rho_T v \\ \frac{w}{\beta^2} & 0 & 0 & \rho & \rho_T w \\ \frac{c_p T}{\beta^2} & 0 & 0 & 0 & \rho c_p + \rho_T c_p T \end{bmatrix}. \tag{13}$$

The choice of nomenclature for β is not arbitrary, rather it is chosen to draw the clear link between the present preconditioning approach and the artificial compressibility method, where β is typically used. However, the matrix Γ in Eqn. 13 represents a preconditioning that treats all equations simultaneously with a pressure time derivative term, rather than only the density equation treatment in the classic artificial compressibility approach. In the author's experience, treating all equations with the preconditioning is critical for the stability and robustness of the method. Early attempts to simply introduce an energy equation with the classic artificial compressibility method were challenging and often met with divergence, especially when fully coupling the energy equation for variable density laminar or RANS problems. As noted previously, the preconditioning here can be seen as a generalization of the artificial compressibility approach to all equations in the governing system, which has been discussed by Turkel[1] for several types of flows.

For steady problems, Eqn. 12 is relaxed in pseudo-time until the time derivative term vanishes and the divergence-free condition is satisfied for isothermal flows, providing an incompressible solution. Therefore, the β parameter is disposable, and in practice, it affects the convergence behavior of the solution more than the accuracy. Several authors have investigated β and its impact.[1,8,22] Following Riedel,[14] we choose the following for a problem independent scaling of β:

$$\beta^2 = \epsilon^2 (\bar{v} \cdot \bar{v})_{max}, \tag{14}$$

where ϵ is a user-provided value (default value of 2.0) and $(\bar{v} \cdot \bar{v})_{max}$ is the maximum squared velocity in the domain, which is updated with each time step for an adaptive β value. More advanced tunings of β are possible, such as those by Colin et al.,[8] but Eqn. 14 performs robustly for all test cases considered in this article without modification. As noted in the literature, the preconditioning in Eqn. 12 destroys the time-accuracy in this form, but time-accurate flows can be computed in a straightforward manner by combining the approach with a dual time-stepping method.[29]

Lastly, we can obtain the eigenvalues Λ_Γ of the preconditioned system from

$$\Lambda_\Gamma = \lambda \left(\Gamma^{-1} \frac{\partial \bar{F}^c}{\partial V} \right) = \lambda \left(\bar{A}^c_\Gamma \right) = \begin{bmatrix} \bar{v} \cdot \bar{n} & 0 & 0 & 0 & 0 \\ 0 & \bar{v} \cdot \bar{n} & 0 & 0 & 0 \\ 0 & 0 & \bar{v} \cdot \bar{n} & 0 & 0 \\ 0 & 0 & 0 & \bar{v} \cdot \bar{n} - \beta & 0 \\ 0 & 0 & 0 & 0 & \bar{v} \cdot \bar{n} + \beta \end{bmatrix}, \tag{15}$$

where $\bar{n} = \{n_x, n_y, n_z\}$ is the local normal vector, $\bar{A}^c_\Gamma$ is the preconditioned convective flux Jacobian, and we see that, as usual with an artificial compressibility-type approach, β acts as a pseudo speed of sound.

C. Variations on the Fluid Modeling

The governing equation set in (1) in the preconditioned form of Eqn. 12 is very flexible for handling a number of variations in the modeling assumptions, from constant density inviscid flows up to variable density turbulent flows with a two-way coupled energy equation and temperature-dependent transport coefficients. In this section, we will briefly cover several different modeling approaches that one can apply.

1. Constant Density Flows

The preconditioned system can easily be applied to constant density flows by removing the coupling from the energy equation back to the mass / momentum equations through the equation of state. Here, we assume a constant density that is no longer dependent on temperature, or $\rho = \rho_o$, and in this case, the preconditioning matrix becomes

$$\Gamma = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 & 0 \\ \frac{u}{\beta^2} & \rho & 0 & 0 & 0 \\ \frac{v}{\beta^2} & 0 & \rho & 0 & 0 \\ \frac{w}{\beta^2} & 0 & 0 & \rho & 0 \\ \frac{c_p T}{\beta^2} & 0 & 0 & 0 & \rho c_p \end{bmatrix}, \tag{16}$$

as $\rho_T = 0$. With the preconditioning in Eqn. 16 and a constant density assumption, the energy equation is simply one-way coupled from mass / momentum to energy and can be optionally computed. For constant density problems without heat transfer (isothermal), the energy equation is unnecessary, and the final row and column of Eqn. 16 can be removed:

$$\Gamma = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 \\ \frac{u}{\beta^2} & \rho & 0 & 0 \\ \frac{v}{\beta^2} & 0 & \rho & 0 \\ \frac{w}{\beta^2} & 0 & 0 & \rho \end{bmatrix}. \tag{17}$$

Eqn 17 makes the clearest connection to the artificial compressibility approach and the analysis of Turkel[1] concerning the inclusion of pressure time derivatives for all equations in the system.

If an inviscid assumption is appropriate for a given problem, the incompressible Euler equations can be recovered from the preconditioned system by eliminating all viscous terms and removing the energy equation. Here, we have the simplest subset of the present framework with a constant density assumption, the convective terms only, and the preconditioning in Eqn. 17. With the Euler equations, flow tangency boundary conditions are applied to solid surfaces in place of the no-slip condition.

2. Body Forces

A general body force can be added by including the following source term in (1):

$$\mathcal{Q} = \left\{ \begin{array}{c} \cdot \\ \rho \bar{f} \\ \cdot \end{array} \right\}, \tag{18}$$

where $\bar{f} = \{f_x, f_y, f_z\}^\mathsf{T}$ is a body force vector.

3. Natural Convection

For natural convection-type problems with large temperature variations, we assume a variable density fluid with a suitable equation of state $\rho = \rho(T)$ along with gravity. In this case, Eqn. 18 takes the form of a buoyancy term and can be expressed as

$$\mathcal{Q} = \left\{ \begin{array}{c} \cdot \\ (\rho - \rho_o)\bar{g} \\ \cdot \end{array} \right\}, \tag{19}$$

where we subtract the constant operating density ρ_o multiplied by gravity $\bar{g}$ in order to remove the hydrostatic pressure component, which can be important from a numerical implementation point of view, especially for boundary conditions. If the temperature variations are small, we can make a final simplification and apply the Boussinesq approximation for the source term instead as

$$\mathcal{Q} = \left\{ \begin{array}{c} \cdot \\ -\rho_o \beta (T - T_o) \bar{g} \\ \cdot \end{array} \right\}, \tag{20}$$

where we assume a constant density fluid throughout the calculation ($\rho = \rho_o$) but approximate the difference ($\rho - \rho_o$) from Eqn. 19 with variations in temperature from a one-way coupled energy equation. β is the thermal expansion coefficient, and T_o is the operating temperature.

4. Axisymmetric Flows

For axisymmetric flows, it is possible to rewrite the equations in a cylindrical coordinate system such that one obtains a 2D formulation of the equations that can be solved numerically for both 2D planar flows or axisymmetric flows with an added source term.[30,31] It is also possible to introduce swirl terms for velocity in the circumferential direction for axisymmetric flows, but that is not considered here. The following source term should be added to (1) for axisymmetric flows:

$$\mathcal{Q} = \left\{ \begin{array}{c} -\rho \frac{v}{y} \\ -\rho u \frac{v}{y} + \frac{\mu}{y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - \frac{2}{3}\frac{\partial}{\partial x}\left(\mu \frac{v}{y}\right) \\ -\rho v \frac{v}{y} + 2\frac{\mu}{y}\left(\frac{\partial v}{\partial y} - \frac{v}{y}\right) - \frac{2}{3}\frac{\partial}{\partial y}\left(\mu \frac{v}{y}\right) \\ -\rho c_p T \frac{v}{y} + \frac{1}{y}\kappa \frac{\partial T}{\partial y} \end{array} \right\}, \tag{21}$$

where y is the y-coordinate in the cartesian reference frame of the 2D grid (the radial direction in cylindrical coordinates) and the axis of symmetry lies along $y = 0$. The first term in each expression in Q in Eqn. 21 is the contribution from the convective terms, while the others are contributions from the viscous terms, which are zero for inviscid axisymmetric flows.

D. Adjoint Method

The adjoint method is a highly efficient sensitivity analysis technique that allows the calculation of the sensitivity of a single objective function (e.g., drag, pressure drop, heat flux, etc.) with respect to an arbitrary number of control parameters (design variables) at a fixed cost that is roughly equivalent to an additional solution of the primal problem. These sensitivities can be used for gradient-based design optimization, uncertainty quantification, or even for simply gaining intuition into a complex system, for example. With the primal problem formally defined, we can now formulate a corresponding adjoint problem, which we will do at a high level in this section before discussing the implementation later.

After our development above, we can now define our governing equations for the analysis of our particular problem as the preconditioned form in Eqn. 12. Next, we define a functional of interest, and this objective function J should be computable from the solution of our governing equations $R(V) = 0$. Lastly, we define our vector of control input parameters, or design variables α, which are used to control our chosen objective function J.

We assume that the governing equations and the objective function are both functions of the flow solution state V as well as the specific design variables α, i.e., $R = R(\alpha, V(\alpha))$ and $J = J(\alpha, V(\alpha))$. Taking the total derivative of J with respect to α, we find that

$$\frac{dJ}{d\alpha}^{\mathsf{T}} = \frac{\partial J}{\partial \alpha}^{\mathsf{T}} + \frac{\partial J}{\partial V}^{\mathsf{T}} \frac{\partial V}{\partial \alpha}. \tag{22}$$

The fluid state is constrained to satisfy the governing equations, or $R(\alpha, V(\alpha)) = 0$, so we can also take the total derivative of R to give

$$\frac{dR}{d\alpha} = \frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial V}\frac{\partial V}{\partial \alpha} = 0. \tag{23}$$

The direct approach to evaluating $\frac{dJ}{d\alpha}$ would to solve Eqn. 23 for $\frac{\partial V}{\partial \alpha}$ due to a perturbation for each entry in α and to use it to compute the derivative of J with Eqn. 22. The partial derivatives of J and R can be relatively inexpensive to calculate, but the partial derivative of the fluid state V can be extremely expensive to evaluate. Unfortunately, this makes the direct method prohibitively expensive for a large vector of design variables α.

However, there is another option for computing $\frac{dJ}{d\alpha}$ that is highly efficient when the design space is large: the adjoint approach. We introduce a Lagrange multiplier Ψ multiplied by Eqn. 23, subtract the result from Eqn. 22, and rearrange to give

$$\frac{dJ}{d\alpha}^{\mathsf{T}} = \left(\frac{\partial J}{\partial V}^{\mathsf{T}} - \Psi^{\mathsf{T}} \frac{\partial R}{\partial V} \right) \frac{\partial V}{\partial \alpha} + \frac{\partial J}{\partial \alpha}^{\mathsf{T}} - \Psi^{\mathsf{T}} \frac{\partial R}{\partial \alpha}, \tag{24}$$

where $\frac{dJ}{d\alpha}^{\mathsf{T}}$ is a row vector. We can choose the Lagrange multiplier, or adjoint state, such that first term vanishes by solving

$$\frac{\partial R}{\partial V}^{\mathsf{T}} \Psi = \frac{\partial J}{\partial V}, \tag{25}$$

which allows us to recover an expression for $\frac{dJ}{d\alpha}$ that is independent of $\frac{\partial V}{\partial \alpha}$:

$$\frac{dJ}{d\alpha}^{\mathsf{T}} = \frac{\partial J}{\partial \alpha}^{\mathsf{T}} - \Psi^{\mathsf{T}} \frac{\partial R}{\partial \alpha}. \tag{26}$$

Therefore, given a solution V to the primal governing equations $\mathcal{R}(\alpha, V(\alpha)) = 0$ and the corresponding adjoint state Ψ (at a cost roughly equivalent to the primal solution), $\frac{dJ}{d\alpha}$ can be computed at a cost independent of the number of parameters in α.

III. Numerical Implementation

In this section, we overview the numerical implementation details of the preconditioning approach and corresponding adjoint within the SU2 software suite. SU2 is an open-source collection of software tools written in C++ and Python for performing multiphysics simulation and design. It is built specifically for the analysis of partial differential equations (PDEs) and PDE-constrained optimization problems on unstructured meshes with state-of-the-art numerical methods, and it is particularly well suited for adjoint-based shape design.

A. Spatial Integration via the Finite Volume Method

We discretize in space using a finite volume method (FVM)[32–39] with a standard edge-based data structure on a dual grid with control volumes constructed using a median-dual, vertex-based scheme. Median-dual control volumes are formed by connecting the centroids, face, and edge midpoints of all primal cells sharing the particular vertex.

After integrating the governing equations in Eqn. 12 over a control volume and applying the divergence theorem, one obtains the following semi-discretized, integral form:

$$0 = \int_{\Omega_i} \Gamma \frac{\partial V}{\partial t} \, d\Omega + \sum_{j \in \mathcal{N}(i)} (\tilde{F}^c_{ij} + \tilde{F}^v_{ij}) \Delta S_{ij} - Q \, |\Omega_i| \tag{27}$$

where $\tilde{F}^c_{ij}$ and $\tilde{F}^v_{ij}$ are the numerical approximations of the convective and viscous fluxes projected along an edge, respectively. ΔS_{ij} is the area of the face associated with the edge ij, $|\Omega_i|$ is the volume of the dual control volume, and $\mathcal{N}(i)$ is the set of neighboring vertices to vertex i.

The convective and viscous fluxes are evaluated at the midpoint of an edge. The convective fluxes can be discretized using either an upwind or a centered scheme. This section will describe the development of two convective schemes for the preconditioned form of the equations, since the dissipation terms must be adjusted accordingly. The first is a flux difference splitting scheme (FDS), and the second is a centered scheme with artificial dissipation, which is a modified version of the Jameson, Schmidt, and Turkel (JST) scheme.[40]

The preconditioning matrix in Eqn. 13 can be introduced into the typical FDS scheme, along with a transformation from conservative to primitive variables, as follows:[6]

$$\begin{aligned}
\tilde{F}^c_{ij} &= \left(\frac{\bar{F}^c_i + \bar{F}^c_j}{2}\right) \cdot \bar{n}_{ij} - \frac{1}{2}\left|\frac{\partial \bar{F}^c}{\partial U}\right| (U_i - U_j) \\
&= \left(\frac{\bar{F}^c_i + \bar{F}^c_j}{2}\right) \cdot \bar{n}_{ij} - \frac{1}{2}\Gamma\Gamma^{-1}\left|\frac{\partial \bar{F}^c}{\partial U}\right| M(V_i - V_j) \\
&= \left(\frac{\bar{F}^c_i + \bar{F}^c_j}{2}\right) \cdot \bar{n}_{ij} - \frac{1}{2}\Gamma\left(\Gamma^{-1}\left|\frac{\partial \bar{F}^c}{\partial V}\right|\right)(V_i - V_j) \\
&= \left(\frac{\bar{F}^c_i + \bar{F}^c_j}{2}\right) \cdot \bar{n}_{ij} - \frac{1}{2}\Gamma\left|\bar{A}^c_\Gamma\right|(V_i - V_j) \\
&= \left(\frac{\bar{F}^c_i + \bar{F}^c_j}{2}\right) \cdot \bar{n}_{ij} - \frac{1}{2}\Gamma P\left|\Lambda_\Gamma\right| P^{-1}(V_i - V_j),
\end{aligned} \tag{28}$$

where $\bar{n}_{ij}$ is the outward unit normal associated with the face between nodes i and j, V_i is the vector of the primitive variables at point i, and $\bar{F}^c_i$ is the convective flux constructed with data from node i. P is the matrix of eigenvectors of the preconditioned convective flux Jacobian matrix $\bar{A}^c_\Gamma$ projected in the $\bar{n}_{ij}$ direction, and $|\Lambda_\Gamma|$ is a diagonal matrix with entries corresponding to the absolute value of the eigenvalues of the flux Jacobian matrix, as given in Eqn. 15. The matrices Γ, P, $|\Lambda_\Gamma|$, and P^{-1} are constructed with mean values computed from data at nodes i and j. For completeness, the matrices P and P^{-1} are given by

$$P = \begin{bmatrix} 0 & 0 & 0 & -\frac{\beta\rho}{n_z} & \frac{\beta\rho}{n_z} \\ -\frac{n_y}{n_x} & -\frac{n_z}{n_x} & 0 & \frac{n_x}{n_z} & \frac{n_x}{n_z} \\ 1 & 0 & 0 & \frac{n_y}{n_z} & \frac{n_y}{n_z} \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} 0 & -n_x n_y & n_x^2 + n_z^2 & -n_y n_z & 0 \\ 0 & -n_x n_z & -n_y n_z & n_x^2 + n_y^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{n_z}{2\beta\rho} & \frac{n_x n_z}{2} & \frac{n_y n_z}{2} & \frac{n_z^2}{2} & 0 \\ \frac{n_z}{2\beta\rho} & \frac{n_x n_z}{2} & \frac{n_y n_z}{2} & \frac{n_z^2}{2} & 0 \end{bmatrix}. \tag{29}$$

As written, Eqn. 28 is first-order accurate in space. However, second-order accuracy can be achieved via reconstruction of the primitive variables on the cell interfaces by a MUSCL approach.[41] In the MUSCL approach, the flow state is reconstructed from the gradients of the variables at each node to project a higher-order approximation of the state at the mid-point of an edge. For second-order accuracy, a linear approximation is applied:

$$V_L \approx V_i + \phi\left(\nabla V|_i \cdot \bar{r}_{ij}\right), \tag{30}$$

$$V_R \approx V_j + \phi\left(\nabla V|_j \cdot \bar{r}_{ji}\right), \tag{31}$$

where $\bar{r}_{ij}$ is the vector pointing from node i to the midpoint of edge ij, $\bar{r}_{ji}$ is the vector pointing from node j to the midpoint of edge ij, and we have also introduced a slope limiter that is represented by ϕ. Slope limiting is applied within upwind schemes in order to preserve monotonicity in the solution by limiting the gradients during reconstruction. It has been shown that limiting is essential for obtaining smooth solutions for compressible flows with shocks, in particular. The slope limiters of Barth and Jesperson[42] and Venkatakrishnan[43] are available, among others. Eqns. 30 and 31 are used to reconstruct the states in place of V_i and V_j in Eqn. 28, respectively, for a second-order flux approximation.

The JST scheme approximates the convective flux using a central difference with a blend of two types of artificial dissipation to maintain numerical stability by preventing even-odd decoupling of the solution at adjacent nodes. It is second-order in space. The artificial dissipation terms are computed using the differences in the undivided Laplacians (higher-order dissipation) of connecting nodes and the difference in the conserved variables (lower-order dissipation) on the connecting nodes on unstructured grids.[44,45] For compressible flows, the two levels of dissipation are blended based on a pressure switch for triggering lower-order dissipation near discontinuities, e.g., shockwaves.

With a similar procedure for introducing the primitive variables and preconditioning matrix into the FDS scheme, we create a modified version of JST scheme for computing the convective flux approximation, which takes the following form:

$$\tilde{F}^c_{ij} = \bar{F}^c\left(\frac{V_i + V_j}{2}\right)\cdot \bar{n}_{ij} - \Gamma_{ij} d_{ij}. \tag{32}$$

The artificial dissipation d_{ij} along the edge connecting nodes i and j can be expressed as

$$d_{ij} = \left(\varepsilon^{(2)}_{ij}(V_j - V_i) - \varepsilon^{(4)}_{ij}(\nabla^2 V_j - \nabla^2 V_i)\right)\varphi_{ij}\lambda_{ij}, \tag{33}$$

where the undivided Laplacians $\nabla^2 V$, local spectral radius, stretching in the grid and pressure switches are computed as

$$\begin{aligned}
\nabla^2 V_i &= \sum_{k\in\mathcal{N}(i)} (V_k - V_i), && (34)\\
\lambda_{ij} &= \left(|\bar{v}_{ij}\cdot\bar{n}_{ij}| + \beta_{ij}\right)\Delta S, \quad \lambda_i = \sum_{k\in\mathcal{N}(i)} \lambda_{ik}, && (35)\\
\varphi_{ij} &= 4\frac{\varphi_i\varphi_j}{\varphi_i + \varphi_j}, \quad \varphi_i = \left(\frac{\lambda_i}{4\lambda_{ij}}\right)^{\alpha}, && (36)\\
\varepsilon^{(2)}_{ij} &= \kappa^{(2)} s_2\left(\left|\sum_{k\in\mathcal{N}(i)}(p_k - p_i)\right| \Big/ \sum_{k\in\mathcal{N}(i)}(p_k + p_i)\right), && (37)\\
\varepsilon^{(4)}_{ij} &= s_4 \max\left(0, \kappa^{(4)} - \varepsilon^{(2)}_{ij}\right), && (38)
\end{aligned}$$

where $\mathcal{N}(i)$ represents the set of neighboring points to node i, p_i is the pressure at node i, s_2 and s_4 are stretching parameters, α is typically set to 0.3, and $\kappa^{(2)}$ and $\kappa^{(4)}$ are adjustable parameters (typical values on unstructured meshes are $\kappa^{(2)} = 0.5$ and $\kappa^{(4)} = 0.02$). For incompressible flow problems, we expect smooth solutions, and therefore, the low-order dissipation switch should be largely inactive. Using the same approach, we can easily formulate a first-order centered scheme as well, such as Lax-Friedrichs.

The viscous flux approximation $\tilde{F}^v_{ij}$ and any source terms Q do not require special treatment due to the preconditioning. In order to evaluate the viscous fluxes, flow quantities and their first derivatives are required at the faces of the control volumes. The spatial gradients of the flow variables are calculated in a pre-processing step at all vertices using a Green-Gauss or Weighted Least-Squares approach[46] and then averaged to obtain these gradients at the control volume faces when evaluating viscous fluxes along the edges of the unstructured mesh. The viscous fluxes are computed using a corrected average-gradient method.[47] Source terms are approximated at each vertex using piece-wise constant reconstruction within each of the dual control volumes. The convective term for any turbulence model variable is discretized using a scalar upwind scheme, and viscous terms in the turbulence model are computed similarly to the mean flow using a corrected average-gradient method.

B. Time Integration

We now consider the techniques for time-marching the coupled system of ordinary differential equations represented by Eqn. 27, which can be rewritten as

$$\frac{d}{dt}\left(\Gamma_i V_i |\Omega_i|\right) + R_i(V) = 0, \tag{39}$$

where $R_i(V)$ is the numerical residual that represents the integrated sum of all spatial terms for the control volume surrounding vertex i. $|\Omega_i|$ is assumed constant in time for now. The governing equations must be satisfied over the entire time interval during integration, so Eqn. 39 can be discretized either explicitly or implicitly by evaluating the residual $R_i(V)$ at the time t^n or t^{n+1}, respectively. Time level n corresponds to the known solution in its current state, while time level $n+1$ represents the new solution state that is being sought after advancing one time step Δt where $\Delta t = t^{n+1} - t^n$.

Both explicit and implicit schemes are described here in order to demonstrate how the preconditioning is applied in both situations. For a simple forward Euler explicit method, the residual is evaluated using the solution in the current state V^n, and the time derivative is discretized with a first-order forward difference. Applying this to Eqn. 39, one has

$$\Gamma_i \frac{\Delta V_i}{\Delta t_i} |\Omega_i| = -R_i(V^n), \tag{40}$$

where $\Delta V_i = V_i^{n+1} - V_i^n$. An expression for the new solution state is obtained by rearranging terms to yield

$$V_i^{n+1} = V_i^n - \Gamma_i^{-1} R_i(V^n) \frac{\Delta t_i}{|\Omega_i|}, \tag{41}$$

where Γ^{-1} is given by

$$\Gamma^{-1} = \begin{bmatrix} \frac{\beta^2 \rho_T T}{\rho} + \beta^2 & 0 & 0 & 0 & -\frac{\beta^2 \rho_T}{\rho c_p} \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ -\frac{T}{\rho} & 0 & 0 & 0 & \frac{1}{\rho c_p} \end{bmatrix}. \tag{42}$$

The application of the preconditioning is straightforward: after computing the terms in the spatial residual $R_i(V^n)$ at each node with current solution data, the inverse preconditioning matrix Γ_i^{-1} can be constructed and multiplied locally at each vertex to complete the solution update.

The solution is relaxed to a steady state iteratively with Eqn. 41 and a suitable time step Δt that respects stability limits. Local time-stepping can be used to accelerate convergence to a steady state. A local time step can be calculated from an estimation of the spectral radii at every node i according to

$$\Delta t_i = N_{CFL} \min \left(\frac{|\Omega_i|}{\lambda_i^{conv}}, C \frac{|\Omega_i|^2}{\lambda_i^{visc}}, C \frac{|\Omega_i|^2}{\lambda_i^{heat}} \right), \tag{43}$$

where N_{CFL} is the Courant-Friedrichs-Lewy (CFL) number and C is a constant (typically $C = \frac{1}{4}$). λ_i^{conv} is the integrated convective spectral radius[48] computed as

$$\lambda_i^{conv} = \sum_{j \in \mathcal{N}(i)} \left(|\bar{v}_{ij} \cdot \bar{n}_{ij}| + \beta_{ij} \right) \Delta S_{ij}, \tag{44}$$

where $\bar{v}_{ij} = (\bar{v}_i + \bar{v}_j)/2$ and $\beta_{ij} = (\beta_i + \beta_j)/2$ denote the velocity and the artificial speed of sound at the cell face as an average of the neighboring nodes, respectively. The viscous spectral radius λ_i^{visc} is computed as

$$\lambda_i^{visc} = \sum_{j \in \mathcal{N}(i)} \frac{4}{3} \frac{\mu_{ij}}{\rho_{ij}} \Delta S_{ij}^2, \tag{45}$$

where μ_{ij} is the sum of the mean laminar and eddy viscosities on the edge in a turbulent calculation and ρ_{ij} is the mean density evaluated at the midpoint of the edge ij. Lastly, the heat conduction spectral radius is estimated as

$$\lambda_i^{heat} = \sum_{j \in \mathcal{N}(i)} \frac{\kappa_{ij}}{c_v \rho_{ij}} \Delta S_{ij}^2, \tag{46}$$

where κ_{ij} is the mean thermal conductivity based on values at nodes i and j and c_v is the specific heat at constant volume.

Implicit methods can be used to improve convergence due to their increased numerical stability. For the backward Euler scheme, the residual is evaluated using the solution state at the new time level V^{n+1}. Applying this to Eqn. (39), one has

$$\Gamma_i \frac{\Delta V_i}{\Delta t_i} |\Omega_i| = -R_i(V^{n+1}). \tag{47}$$

A first-order linearization about time level n provides an approximation to $R_i(V^{n+1})$:

$$R_i(V^{n+1}) = R_i(V^n) + \sum_{j \in \mathcal{N}(i)} \frac{\partial R_i(V^n)}{\partial V_j} \Delta V_j^n + \mathcal{O}(\Delta t^2). \tag{48}$$

Introducing Eqn. (48) into Eqn. (47), we find that the following linear system should be solved to obtain the solution update (ΔV_i^n):

$$\left(\Gamma_i \frac{|\Omega_i|}{\Delta t_i^n} \delta_{ij} + \frac{\partial R_i(V^n)}{\partial V_j} \right) \cdot \Delta V_j^n = -R_i(V^n), \tag{49}$$

where if a flux $\tilde{F}_{ij}$ has a stencil of points $\{i, j\}$, then contributions are made to the Jacobian at four points, or

$$\frac{\partial R}{\partial V} := \frac{\partial R}{\partial V} + \begin{bmatrix} \ddots & & & & \\ & \frac{\partial \tilde{F}_{ij}}{\partial V_i} & \cdots & \frac{\partial \tilde{F}_{ij}}{\partial V_j} & \\ & \vdots & \ddots & \vdots & \\ & -\frac{\partial \tilde{F}_{ij}}{\partial V_i} & \cdots & -\frac{\partial \tilde{F}_{ij}}{\partial V_j} & \\ & & & & \ddots \end{bmatrix}. \tag{50}$$

Similar to the explicit scheme scenario, the preconditioning is relatively straightforward to apply to the implicit method. As seen in the first term of Eqn. 49, we simply multiply the block contribution along the diagonal involving the volume and time step by the preconditioning matrix Γ_i and add it to the Jacobian matrix for node i. This can be completed as a final loop through all local vertices before solving the linear system for the solution update.

C. Adjoint-based Optimal Shape Design

SU2 contains multiple implementations of the adjoint methodology within the same codebase, including a continuous adjoint[49,50] and a discrete adjoint[51,52] constructed via algorithmic differentiation (AD) of the code using the open-source C++ library Code-Differentiation Package (CoDiPack).[53] The discrete adjoint solver has been implemented in a general way within the SU2 suite, such that a consistent discrete adjoint for any evolutions and extensions of the primal analysis capabilities will be available upon recompilation without the need to manually differentiate any new components for inclusion. As such, in this work, we will leverage the availability of AD to construct a discrete adjoint of the preconditioning approach developed in this article and use it for performing optimal shape design in variable density incompressible flows.

Here, we briefly detail the process of constructing the discrete adjoint as implemented within SU2. First, we express the PDE-constrained optimization problem:

$$\begin{aligned} \min_{\alpha} \quad & J(V(\alpha), X(\alpha)) \\ \text{subject to} \quad & V(\alpha) = G(V(\alpha), X(\alpha)) \\ & X(\alpha) = M(\alpha), \end{aligned} \tag{51}$$

where we now define the objective J as a function of both the flow state $V(\alpha)$ and also the mesh state $X(\alpha)$, as we will be solving for discrete solutions of the governing equations on numerical grids. Given that we are interested in shape optimization, both the flow state and the mesh will be implicit functions of our design variables α, i.e., any shape perturbations to the geometry in our simulation will result in changes in both the flow state and the numerical grid (surface and volume). The grid deformations will be governed by the solution of another discretized PDE, typically the linear elasticity equations in a pseudo-structural mesh deformation approach, and we denote this with an operator M, such that $X(\alpha) = M(\alpha)$. Finally, we introduce a fixed-point operator for the governing flow equations defined as:

$$V^{n+1} = G(V^n) := V^n - Q(V^n)R(V^n), \tag{52}$$

where Q is our approximate Jacobian from Eqn. 49, or

$$Q(V^n) := \left(\Gamma \frac{|\Omega|}{\Delta t^n} \delta_{ij} + \frac{\partial R(V^n)}{\partial V} \right)^{-1}. \tag{53}$$

By introducing the fixed-point formulation, an adjoint can be constructed that resembles the iterative flow solver and enables the use of the same approximate Jacobian from the primal solve. The ability to avoid the construction of exact Jacobians for the discrete adjoint has important implications for both performance and robustness of the method.

We transform the constrained optimization problem in Eqn. 51 into an unconstrained problem by forming the Lagrangian:

$$J(\alpha, V, X, \Psi, \Phi) = J(V, X) + \Psi^{\mathsf{T}} \left(G(V, X) - V\right) + \Phi^{\mathsf{T}} \left(M(\alpha) - X\right), \tag{54}$$

where we have introduced our adjoint states for the flow and mesh as Ψ and Φ, respectively. We can now differentiate J with respect to our design parameters α and rearrange as

$$\frac{dJ}{d\alpha}^{\mathsf{T}} = \left(\frac{\partial J}{\partial V}^{\mathsf{T}} + \Psi^{\mathsf{T}}\frac{\partial G}{\partial V} - \Psi^{\mathsf{T}}\right)\frac{\partial V}{\partial \alpha} + \left(\frac{\partial J}{\partial X}^{\mathsf{T}} + \Psi^{\mathsf{T}}\frac{\partial G}{\partial X} - \Phi^{\mathsf{T}}\right)\frac{\partial X}{\partial \alpha} + \Phi^{\mathsf{T}}\frac{dM}{d\alpha}, \tag{55}$$

where $\frac{dJ}{d\alpha}^{\mathsf{T}}$ is a row vector. In this form, we now clearly see that to eliminate the expensive derivatives, in this case both $\frac{\partial V}{\partial \alpha}$ as well as the mesh component $\frac{\partial X}{\partial \alpha}$, we should solve the following adjoint problems:

$$\Psi^{\mathsf{T}} = \frac{\partial J}{\partial V}^{\mathsf{T}} + \Psi^{\mathsf{T}}\frac{\partial G}{\partial V}, \tag{56}$$

$$\Phi^{\mathsf{T}} = \frac{\partial J}{\partial X}^{\mathsf{T}} + \Psi^{\mathsf{T}}\frac{\partial G}{\partial X}. \tag{57}$$

After satisfying Eqns. 56 and 57, the final formula for evaluating the total derivative of the objective function becomes

$$\frac{dJ}{d\alpha}^{\mathsf{T}} = \Phi^{\mathsf{T}}\frac{dM}{d\alpha}. \tag{58}$$

Eqn. 56 is a fixed-point iteration for the adjoint flow state and can be solved in a similar fashion to the primal solver once a solution V is available on the current mesh X. As described in detail by Albring et al.,[51,52] the derivatives and expressions above can be evaluated using AD, which allows for G to be considered as a black box, i.e., the AD tool will enable the construction of a discrete adjoint independent from the specific structure of G. In practice, this enables straightforward construction of consistent discrete adjoints for a rapidly evolving code framework, such as SU2, as new features and capabilities are added.

In addition to embedded adjoint techniques, SU2 ships with all of the tools necessary for optimal shape design, including the modules for the numerical solution of PDEs, additional sensitivity analysis methods, geometry parameterizations, and mesh deformation techniques. Typically, geometry parameterization is accomplished with either Hicks-Henne bumps in 2D[54] or a free-form deformation (FFD) approach in 2D or 3D,[55] mesh deformation with an elasticity-based approach,[56] and the shape design loop is driven by the SciPy SLSQP optimizer.[57]

IV. Results

In this section, we investigate the following series of verification and validation cases: a buoyancy-driven cavity, a heated flat plate in laminar flow, a flat plate in turbulent flow, the NACA 0012 in turbulent flow at two angles of attack, a 3D bump in a channel in turbulent flow, and an axisymmetric laminar pipe flow. For a final case, adjoint-based shape optimization of a heated cylinder in a cross-flow is demonstrated.

All SU2 results within this section leverage the proposed preconditioning approach for incompressible flows. Unless stated otherwise, mean flow convective fluxes are computed with the flux difference splitting upwind scheme developed in Section III with a second-order reconstruction. No slope limiters are applied. Turbulent variables for the S-A and SST models are convected with a first-order scalar upwind method. Viscous fluxes are calculated using a corrected average-gradient method. Boundary conditions for the turbulence models are as given for the standard S-A and SST models on the NASA Turbulence Modeling Resource (TMR)[a]. Implicit, local time-stepping is used to relax problems to steady-state solutions, and the linear systems are solved using GMRES with an ILU preconditioner with zero fill-in. Non-linear geometric multigrid (FAS) is applied when possible for convergence acceleration. RANS cases are typically relaxed with a CFL number of 25, while some laminar problems are stable up to a CFL of 1000 or more. Calculations are converged to machine precision in the residual of the pressure equation, unless otherwise noted. The default expression for β from Eqn. 14 is used for all calculations without any additional tuning.

[a] https://turbmodels.larc.nasa.gov/

Downloaded by Thomas Economon on July 3, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-3111

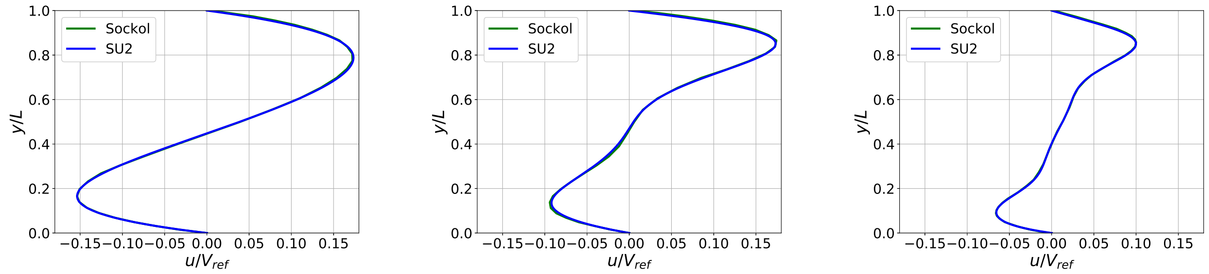
A. 2D Buoyancy-driven Cavity Verification Case

The first verification case for the preconditioning approach is a 2D buoyancy-driven cavity, also known as a differentially-heated cavity. This is a classic natural convection case for testing incompressible flow solvers with heat transfer effects. The geometry consists of a square cavity in 2D with opposing hot and cold vertical walls and insulated horizontal walls. A gravity force is added in the negative y-direction. For small temperature differences between the two walls, the Boussinesq approximation is valid. However, a large temperature difference is imposed between the two plates in order to test the full variable density capability of the solver.

For comparison purposes, the same problem set up from Sockol[7] is employed here. The Rayleigh number (Ra) is the key parameter, and it is defined by

$$Ra = \frac{\Delta T \, \rho^2 \, g \, c_p \, L^3}{\mu_{dyn} \, \kappa}, \tag{59}$$

where $\Delta T = 2(T_h - T_c)/(T_h + T_c)$, T_h is the constant temperature of the hot left wall, T_c is the constant temperature of the cold right wall, g is the acceleration due to gravity, and L is the length of a side of the square cavity. Additionally, we impose that $T_h/T_c = 4$ and that μ_{dyn} and κ are constants with $\kappa = \mu_{dyn} \, c_p / Pr_d$. The laminar Prandtl ($Pr_d$) and Froude ($Fr$) numbers are 0.7 and 1.2, respectively. A reference velocity can then be computed as $V_{ref} = \sqrt{Fr \, g \, L}$.



(a) X-velocity comparison at the centerline for $Ra = 1e3$.

(b) X-velocity comparison at the centerline for $Ra = 1e5$.

(c) X-velocity comparison at the centerline for $Ra = 1e6$.

Figure 1. Buoyancy-driven cavity verification of x-velocity profiles for varying Ra number.

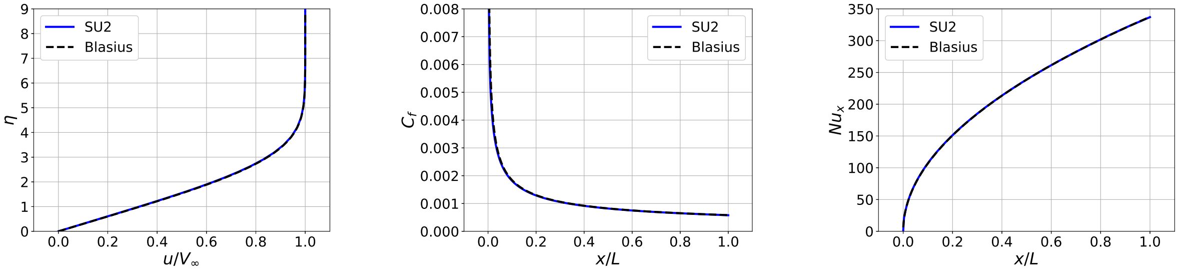
The flow is computed on a relatively fine 257 × 257 uniform grid for a 1 m × 1 m domain, resulting in a total of 66,049 vertices and 65,536 quadrilaterals. An adiabatic no-slip condition is applied to the horizontal walls, while an isothermal no-slip condition is applied to the vertical walls. Standard air governed by the ideal gas law is assumed for the fluid with $\gamma = 1.4$ and $R = 287.058$ J/(kg-K). The hot wall is 461.04 K, the cold wall is 115.26 K, and the initial temperature in the cavity is 288.15 K. The viscosity and thermal conductivity are constants with the values 1.716e-5 kg/(m-s) and 0.02462 W/(m-K), respectively. Gravity is imposed as -9.81 m/s^2 in the y-direction. The Ra number is controlled by changing the initial operating density in the cavity, which also sets the constant thermodynamic pressure in the cavity via the ideal gas law.

Flow velocity is induced within the cavity due to the heated walls combined with the buoyancy effect (variable density fluid under the influence of gravity). Calculations for $Ra = 1e3$, 1e5 and 1e6 were performed, and the SU2 results for x-velocity profiles at the cavity centerline are compared to those of Sockol in Figs. 1(a)-1(c). The agreement is excellent in all three comparisons.

B. 2D Laminar Flat Plate Verification Case

The next verification case is for 2D laminar, incompressible, external flow over a flat plate. The analytical solutions of Blasius[58] can be compared to computed values for the velocity profile and skin friction coefficients. In this case, the flat plate is also fixed at a different temperature from that of the free-stream fluid, such that a thermal boundary layer appears for comparison with Blasius correlations for the local Nusselt number Nu_x. To accomplish this, the energy equation is solved with a one-way coupling while the fluid density is held constant.

The flat plate grid is composed of quadrilaterals with 65 nodes in both the x- and y-directions. The flat plate is part of the lower boundary of the domain ($y = 0$) starting at $x = 0$ m with a total length of 0.3048 m. An isothermal, no-slip boundary condition is applied to the flat plate. Upstream of the flat plate, a symmetry condition is applied to represent the free-stream approaching the zero-thickness flat plate. The mesh is stretched in the y-direction to better resolve the boundary layer on the flat plate with approximately 30 points in the boundary layer. A uniform velocity inlet condition is imposed at the inlet boundary, and pressure outlet conditions are imposed on the upper and downstream boundaries of the domain.



(a) Laminar velocity profile comparison.

(b) C_f comparison.

(c) Local Nusselt number comparison.

Figure 2. Comparisons of computed results from SU2 and the analytic solutions of Blasius for laminar flow past a flat plate.

The free-stream velocity is 69.1687 m/s in the positive x-direction, and the free-stream temperature is 297.62 K. The outlet pressure is set to atmospheric (0 Pa). The dynamic viscosity is assumed constant at 1.83463e-5 kg/(m-s). The resulting Reynolds number is 1.3e6. The flat plate temperature is set to 197.62 K by the isothermal, no-slip boundary condition. Fig. 2(a) presents the well-known laminar boundary layer velocity profile plotted against the non-dimensional variable η (wall-normal coordinate). For the skin friction and local Nusselt number, we have

$$C_f = \frac{\tau}{0.5\rho V_\infty^2} \approx 0.664\, Re_x^{-1/2}, \tag{60}$$

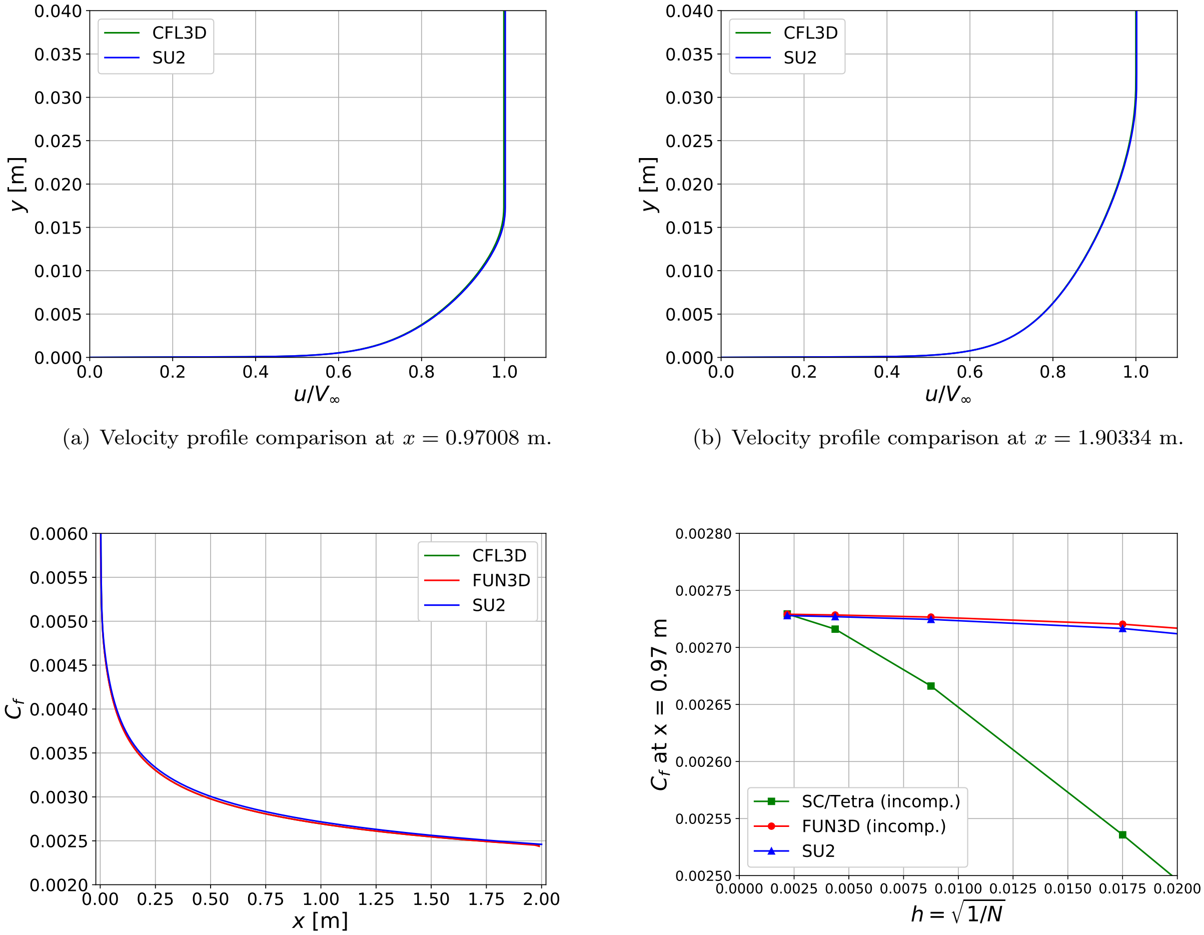
$$Nu_x = \frac{\dot{q}x}{\kappa \Delta T} \approx 0.33\, Re_x^{1/2}\, Pr_d^{1/3}, \tag{61}$$

where $Re_x = \rho V_\infty x/\mu_{dyn}$, $\dot{q}$ is the heat flux, and ΔT here is the difference in temperature between the wall and the free-stream fluid. Figs 2(b) and 2(c) show the computed values of C_f and Nu_x from SU2 (the first expressions on the right-hand side of Eqns. 60 and 61) as compared to the Blasius correlations (second expressions). All of the computed results match exceptionally well with the analytic solutions for this problem.

C. 2D Zero Pressure Gradient Flat Plate Verification Case

2D flow over a flat plate with zero pressure gradient is a useful first verification for turbulence model implementations. Here, SU2 is demonstrated for this RANS case and compared to results from NASA's compressible CFL3D and FUN3D codes. Grids, comparison data, and flow conditions are available from the NASA TMR.[59] The flow conditions for this case are $M_\infty = 0.2$, $Re = 5\text{e}6$ based on plate length, $T_\infty = 300$ K, and the plate is at a 0° angle of attack. The new preconditioning approach is applied here in constant density form (isothermal flow) for the mean flow equations with the S-A model for turbulence. To match the specified Re for this problem in the incompressible solver, the following conditions are used: $\rho_\infty = 1.32905$ kg/m^3, $V_\infty = 69.4448$ m/s in the positive x-direction, and $\mu_{dyn} = 1.84592\text{e-}5$ kg/(m-s). The dynamic viscosity is held constant.

A rectangular domain is considered and discretized with quadrilateral elements. A grid from the NASA TMR is chosen with 545 × 385 cells in the stream-wise and normal directions, respectively. An adiabatic, no-slip boundary condition is applied to the flat plate. Upstream of the flat plate, a symmetry condition is applied to represent the free-stream approaching the zero-thickness flat plate. The mesh is stretched in the y-direction to ensure $y^+ < 1$ over the plate. A uniform velocity inlet condition is imposed at the inlet boundary, and pressure outlet conditions are imposed on the upper and downstream boundaries of the domain. The outlet pressure is set to atmospheric (0 Pa).



(a) Velocity profile comparison at $x = 0.97008$ m.

(b) Velocity profile comparison at $x = 1.90334$ m.

(c) C_f comparison along the length of the plate.

(d) Grid convergence comparison for the value of C_f at $x = 0.97008$ m for different incompressible codes. h is an effective grid spacing proportional to $\sqrt{1/N}$, where N is the number of cells in the grid.

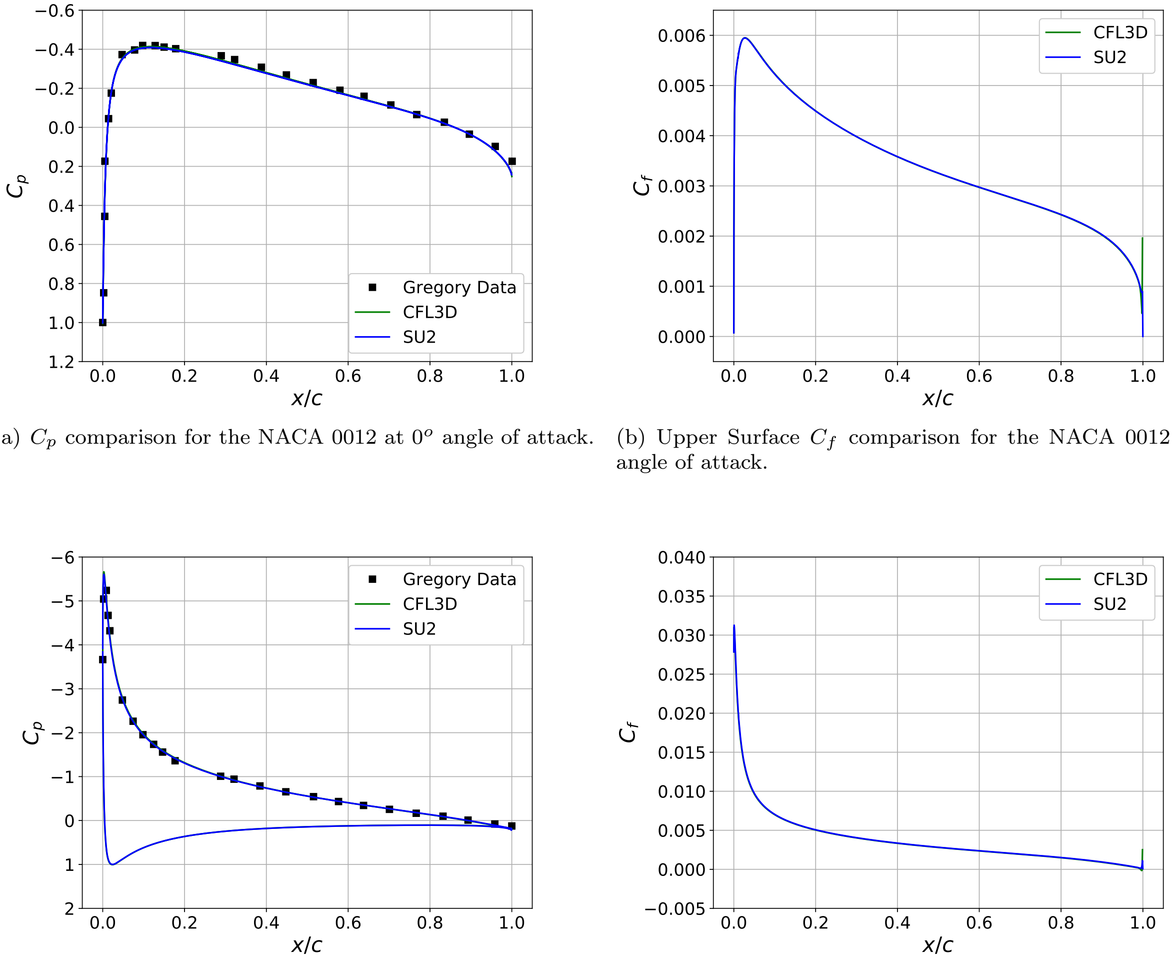
Figure 3. Verification of the 2D Zero Pressure Gradient Flat Plate from the NASA TMR.

Fig. 3(a) and 3(b) contain comparisons of results from SU2 and CFL3D of velocity profiles extracted at two different x-locations as the boundary layer develops along the plate. The profiles are in very good agreement. Fig. 3(c) presents the skin friction coefficient along the length of the plate as computed by SU2 and compared to both CFL3D and FUN3D. Here, some disagreement is visible, but this is to be expected given the mildly compressible conditions and comparisons against results from compressible codes. The NASA TMR reports that compressible and incompressible codes arrive at slightly different results for this case, and additional data for incompressible codes are also available. Consequently, Fig. 3(d) displays a grid convergence study for the value of C_f at $x = 0.97008$ m computed by SU2 using the series of NASA TMR grids and compared against data from two different incompressible solvers. All three solvers converge to a very similar value of C_f on the finest grid, and the results from SU2 and FUN3D in incompressible mode compare favorably on all grids. The C_f values for the incompressible codes are all slightly larger than their compressible counterparts, which is consistent with the findings in Fig. 3(c).

D. 2D NACA 0012 Airfoil Validation Case

This RANS validation case simulates 2D flow over a NACA 0012 airfoil under essentially incompressible free-stream conditions as specified by the NASA TMR.[60] SU2 results are compared against published experimental data from Gregory[61] and also against NASA's CFL3D compressible code. The flow conditions for this case are $M_\infty = 0.15$, $Re = 6\text{e}6$ based on chord length, $T_\infty = 300$ K, and the calculations are performed at two angles of attack (0°, 10°). Constant density, incompressible flow is applied for the mean flow equations with the S-A model for turbulence. To match the specified Re for this problem in the incompressible solver, the following conditions are used: $\rho_\infty = 2.13163$ kg/m^3, $V_\infty = 52.1572$ m/s, and $\mu_{dyn} = 1.853\text{e-}5$ kg/(m-s). The dynamic viscosity is held constant.

A series of C-grids are available from the NASA TMR, and for the results presented below, the quadrilateral grid with 897 nodes in the airfoil-normal and 257 in the airfoil-tangent directions is used. The far-field boundary is approximately 500 chord lengths away from the airfoil surface, and the mesh spacing near the airfoil is sufficient to ensure $y^+ < 1$ over the airfoil surface. An adiabatic, no-slip condition is applied to the airfoil surface.



(a) C_p comparison for the NACA 0012 at 0^o angle of attack.

(b) Upper Surface C_f comparison for the NACA 0012 at 0^o angle of attack.

(c) C_p comparison for the NACA 0012 at 10^o angle of attack.

(d) Upper Surface C_f comparison for the NACA 0012 at 10^o angle of attack.

Figure 4. NACA 0012 surface pressure and skin friction coefficients as compared to experiment and CFL3D data from the NASA TMR.

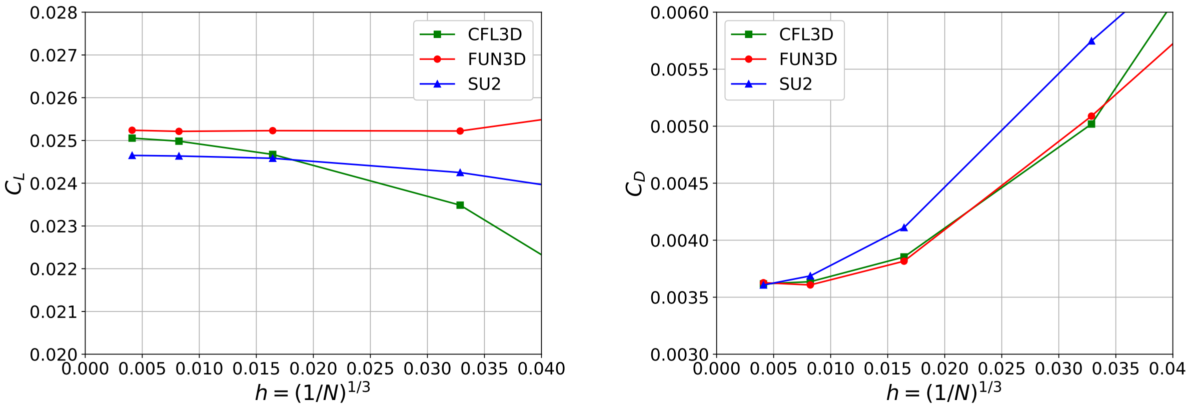
Airfoil surface pressure and skin friction coefficients are shown in Figs. 4(a)-4(d) for the two angle of attack conditions. The computed SU2 solutions are in good agreement with the published data from Gregory. In addition, the computed values for C_p and C_f for both angle conditions are nearly indistinguishable from the CFL3D results with the S-A model. The only noticeable discrepancies appear at the suction peak for the pressure and at the trailing edge of the airfoil for the skin friction. However, it is common to see some solution noise at sharp trailing edges.

E. 3D Bump-in-Channel Verification Case

Flow over a bump in a channel introduces non-zero pressure gradients on the wall, which makes it a valuable verification case for RANS solvers. This is an important verification case for the preconditioning approach, as 3D internal flows are common applications for incompressible codes. Additionally, this case considers the SST turbulence model, which has not been demonstrated yet. As with the flat plate and NACA 0012, grids and comparison data, computed using the CFL3D and FUN3D codes with the SST model, are available through the NASA TMR.[62]

The flow conditions for this case are $M_\infty = 0.2$, $Re = 3\text{e}6$ based on length "1" of the grid (assumed meters), and $T_\infty = 300$ K. Constant density, incompressible flow is applied for the mean flow equations with the SST model for turbulence. To match the specified Re for this problem in the incompressible solver, the following conditions are imposed at the inlet: $\rho = 1.0$ kg/m^3, $V = 60.0$ m/s in the positive x-direction, and $\mu_{dyn} = 2.0\text{e-}5$ kg/(m-s). The dynamic viscosity is held constant.

A series of nested, hexahedral grids are available from the NASA TMR with the following cell dimensions: $65 \times 705 \times 321$, $33 \times 353 \times 161$, $17 \times 177 \times 81$, $9 \times 89 \times 41$, $5 \times 45 \times 21$. The channel geometry features a curved bump on the lower surface where an adiabatic, no-slip condition is applied. A symmetry condition is applied upstream and downstream of the bump, as well as along the side walls and upper surface of the channel. A uniform velocity inlet condition is imposed at the inlet boundary, and a pressure outlet condition is imposed on the downstream boundary of the domain. The outlet pressure is set to atmospheric (0 Pa). The mesh spacing is stretched in the wall-normal direction to ensure $y^+ \approx 1$ over the bump surface for all grids, and the cells are also clustered near the beginning and end of the no-slip bump. The span-wise spacing is uniform.



(a) Grid convergence for the value of C_L on the bump.

(b) Grid convergence for the value of C_D on the bump.

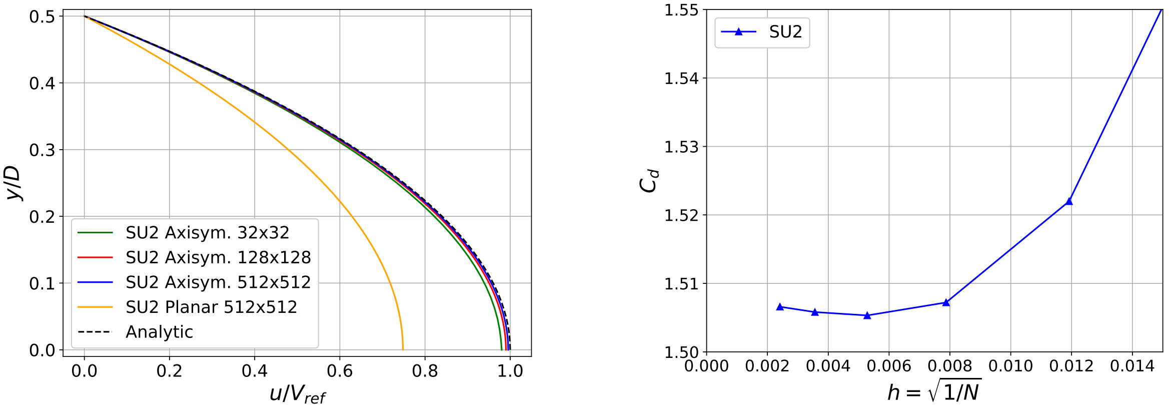
Figure 5. Grid convergence study of the 3D bump-in-channel as compared to CFL3D and FUN3D data from the NASA TMR. h is an effective grid spacing proportional to $(1/N)^{1/3}$, where N is the number of cells in the grid.

Calculations are performed on each of the 5 available grids in order to assess the grid convergence behavior of C_L and C_D as compared to other software, as seen in Figs. 5(a) and 5(b). The C_L plateaus at a lower value than the compressible codes as the grid is refined. However, similar to the 2D flat plate case, these comparisons of the incompressible SU2 results are against compressible codes in slightly compressible conditions, so small discrepancies are expected as the grids are refined, as noted by the NASA TMR. The C_D values on the finest grid match across all three codes. As these 3D cases are larger and more complex than previous tests, achieving convergence is more challenging. The Venkatakrishnan limiter was applied during the second-order reconstruction of the mean flow convective fluxes, and at least one case ($17 \times 177 \times 81$) exhibited residual "ringing" behavior that is typical of slope limiters after converging several orders of magnitude. Overall, the SU2 results compare favorably to those of CFL3D and FUN3D, and the differences are consistent with expectations.

F. Axisymmetric Laminar Pipe Flow Verification Case

This verification case is for laminar, incompressible, internal flow within a circular pipe, which has a well-known analytic solution for the velocity profile once fully-developed. The solution is axisymmetric, which offers an opportunity to verify the correctness of the axisymmetric source term implementation. The pipe has a length L of 3 m and a diameter D of 0.2 m. A constant density, incompressible fluid flows through the pipe with $Re = 100$ based upon pipe diameter. To match the specified Re for this problem in the incompressible solver, the following conditions are imposed at the inlet: $\rho = 1.0$ kg/m^3, $V = 1.0$ m/s in the positive x-direction, and $\mu_{dyn} = 2.0$e-3 kg/(m-s). The dynamic viscosity is held constant.

An axisymmetric calculation is performed on a series of 2D grids with rectangular elements discretizing the upper half of the pipe domain. No grid stretching is applied. A uniform velocity inlet is imposed at the inlet of the pipe, and a pressure outlet condition is applied to the outlet. The outlet pressure is set to atmospheric (0 Pa). The upper boundary of the rectangular domain is an adiabatic, no-slip wall, and the lower boundary of the domain is the axis of symmetry at $y = 0$.



(a) X-velocity profiles for fully developed laminar pipe flow at x = 2.81 m on a series of rectangular grids (cell numbers given).

(b) Grid convergence for the value of C_d on a cylinder in laminar flow at $Re = 40$.

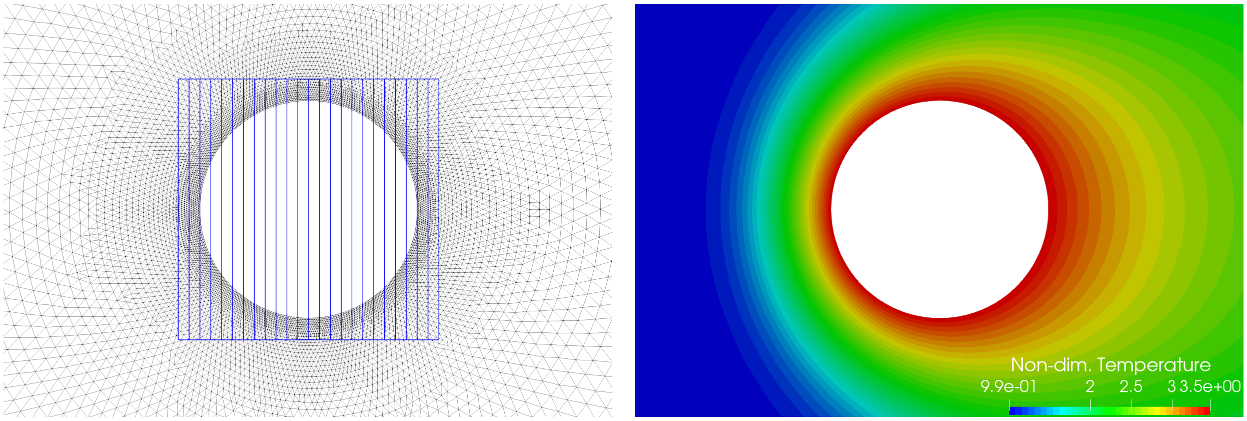
Figure 6. Verification case for axisymmetric laminar pipe flow (left) and a grid convergence study for a 2D laminar cylinder (right).

Fig. 6(a) compares the SU2 results for the x-velocity profile to the analytic solution, which is parabolic with a maximum velocity of 2 m/s at the centerline. The velocity profiles are non-dimensionalized by a reference velocity V_{ref} chosen as the maximum velocity. The pipe radius is aligned with the y-direction and non-dimensionalized by D. As the grid is refined, the velocity profile computed by SU2 closely approaches the analytic solution. For illustration purposes, the solution is also computed in 2D planar mode (without the axisymmetric source term) and plotted in Fig. 6(a). The impact of the additional source term for axisymmetry is made clear by the difference in profiles between the planar and the axisymmetric solutions.

G. 2D Cylinder Verification and Optimization Case

The final case is for laminar, incompressible flow past a 2D cylinder. A grid convergence study is first performed in isothermal flow at $Re = 40$, based on a cylinder diameter D of 1 m, to verify the value of the coefficient of drag computed by SU2 against those reported in the literature. As a second investigation, the cylinder is heated, and the geometry is optimized for improved heat transfer with a drag constraint using the adjoint technique. Therefore, another goal of this case is to demonstrate the adjoint methodology for computing sensitivities of both the drag and the heat flux on the cylinder surface with respect to shape changes.

For the grid convergence study in constant density, isothermal flow, an adiabatic, no-slip boundary condition is set on the cylinder surface. A family of curvilinear O-grids discretizes the domain with quadrilateral elements with a circular far-field boundary. The grids are generated by setting a number of equally-spaced vertices on the cylinder surface and extruding the grid normally such that the first cell height off the wall matches the distance between surface vertices. The extrusion marches radially from the surface at a growth rate of 1.1 until reaching a distance of 50 cylinder diameters. It should be noted that no deliberate effort was made to refine the wake region, rather, it was desired to maintain a family of grids with similar characteristics (although not perfectly nested). The following conditions are imposed at the far-field: $\rho_\infty = 1.0$ kg/m^3, $V_\infty = 4.0$ m/s, and $\mu_{dyn} = 0.1$ kg/(m-s). The dynamic viscosity is held constant. Fig. 6(b) contains the convergence of C_d as the grid is refined. The value of C_d on the finest three grids is 1.507, which is in good agreement with values reported in literature, such as 1.51 by Park et al.[63] and 1.509 by Sen et al.[64]



(a) Zoom view of the unstructured grid and FFD control points for shape parameterization.

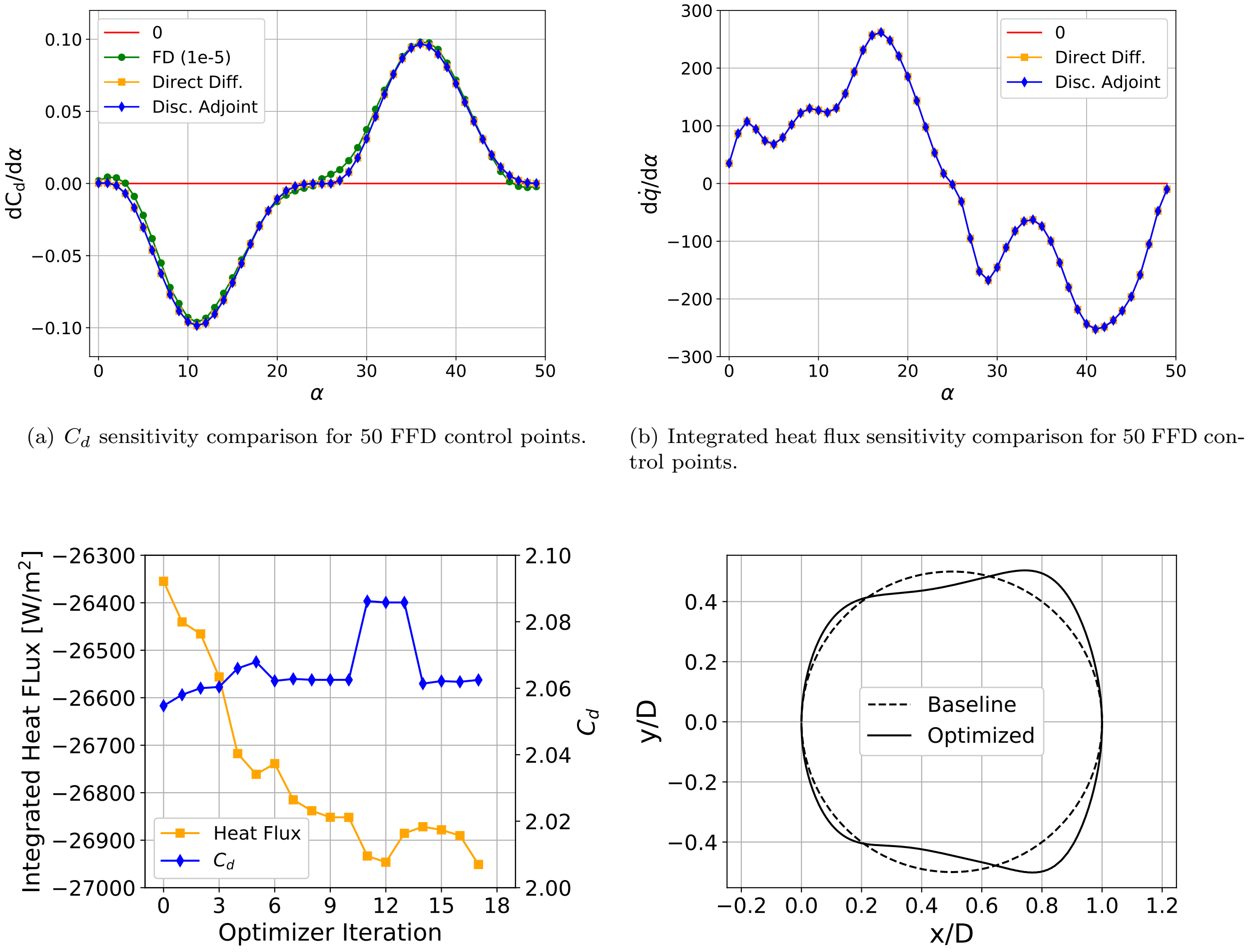
(b) Temperature contours for flow past a heated cylinder for $Re = 40$.

Figure 7. Mesh and temperature field visualization for flow past a heated cylinder.

As a second investigation, an isothermal condition is imposed on the cylinder such that its temperature is considerably larger than the incoming free-stream. The test case is meant to be a simple analogy to a pin heat sink for a cooling application. For this case, a 2D grid around the cylinder is composed of 26,192 triangular elements with tighter spacing near the cylinder and a far-field approximately 15 diameters away from the cylinder. Fig. 7(a) contains a zoom view of the triangular grid near the cylinder, along with the lattice of 50 FFD control points (25 × 2) making up the design parameterization for adjoint-based shape optimization. The 25 control points on the top and bottom of the cylinder are each free to move in the y-direction only, and their motion controls the shape of the cylinder.

A variable density, incompressible flow governed by the ideal gas law is assumed (standard air), requiring the fully-coupled solution of the energy equation for the temperature in the field. An isothermal, no-slip boundary condition is applied to the cylinder with a temperature of 1000 K. The free-stream conditions are $\rho_\infty = 0.00021$ kg/m^3, $V_\infty = 3.40297$ m/s in the positive x-direction, $T_\infty = 288.15$ K, and $\mu_{dyn_\infty} = 1.7893\text{e-}5$ kg/(m-s). Given the large disparity in temperature between the cylinder and the free-stream, this case is an opportunity to test both the variable density capability of the solver along with temperature-dependent fluid properties. Consequently, μ_{dyn} and κ vary with temperature according to Sutherland's Law and a constant Pr_d of 0.72, respectively. Fig. 7(b) shows contours of non-dimensional temperature around the heated cylinder.

Before attempting shape optimization, the gradients of the C_d and integrated heat flux with respect to the vertical motion of the 50 FFD control points are computed via the discrete adjoint method and compared to a simple finite differencing (FD) and a direct differentiation using AD. Finite differencing and direct differentiation are both direct methods for sensitivity analysis, meaning that the cost is proportional to the number of design variables α, i.e., computing the gradient for N design variables requires at least N calculations. Fig. 8(a) and 8(b) present the gradient comparisons. At convergence, the discrete adjoint should provide numerically exact sensitivities, and direct differentiation is typically used to verify their accuracy. In both comparisons for C_d and integrated heat flux, the discrete adjoint matches the result of direct differentiation to high precision, which is to be expected (reverse and forward mode, respectively, of the same AD tool). FD results are shown for C_d sensitivities given a step size of 1e-5 m for the FFD control point perturbations. The FD results compare favorably, but some discrepancies are apparent, likely due to the typical truncation error or subtractive cancellation issues with FD. FD results for integrated heat flux are not shown, as a suitable step size could not be found to give accurate sensitivities for this problem.



(a) C_d sensitivity comparison for 50 FFD control points.

(b) Integrated heat flux sensitivity comparison for 50 FFD control points.

(c) Shape optimization history for the heated cylinder case.

(d) Comparison of baseline and optimized geometries for improved heat flux with a drag constraint.

Figure 8. Gradient verification and adjoint-based shape optimization results for the heated cylinder case.

As a final demonstration, gradient-based shape optimization of the cylinder is performed. The objective is to increase the integrated heat flux from the cylinder to the fluid (improve cooling) while imposing a constraint on drag. Drag and integrated heat flux are opposing objectives for this problem: to minimize drag, the cylinder profile should be made thinner and streamlined, while to improve heat flux, the profile should be made thicker with more area for heat transfer. Figs 8(c) and 8(d) display the history for the drag and integrated heat flux values for each major optimizer iteration of the SLSQP algorithm, as well as a comparison of the initial and final geometries. The magnitude of the integrated heat flux from the cylinder is increased by 2.2 % while the C_d is maintained at 2.062, which is just slightly in violation of the imposed C_d constraint of 2.06. The results could be improved with further tweaking of the design variable number and position, optimizer parameters (including scale factors), and additional optimizer convergence. However, the results are a successful demonstration of adjoint-based optimization for variable density, incompressible flows with heat transfer with the preconditioning approach in SU2.

V. Conclusion

In this article, we have developed a preconditioning scheme for computing variable density incompressible flows with heat transfer. The preconditioning is applied in order to solve the low-Mach Navier-Stokes equations with a coupled approach in the context of an unstructured finite volume code with vertex-based schemes. In addition to providing detail about the preconditioning formulation and implementation, we have demonstrated the solver through a set of standard verification and validation (V&V) cases for laminar and turbulent flows. Design of a heated cylinder demonstrates the adjoint capability of the solver that can be used for efficient shape optimization.

The primary contribution of the article is the preconditioning formulation that has been shown to perform robustly for all cases considered. The formulation can be seen as a generalization of the artificial compressibility approach to a preconditioning that treats all equations in the low-Mach Navier-Stokes system, as an extension of earlier work by Turkel,[1] or it could also be seen as a simplification of the preconditioning approach developed by Weiss and Smith[6] who sought to unify artificial compressibility and low-speed preconditioning approaches into one method for all flow speeds. During development of the present formulation, the author found that treating all equations with the preconditioning, as opposed to the density equation alone as with artificial compressibility, is critical for the stability and robustness of the method, especially when fully coupling the energy equation for variable density problems.

The V&V of the preconditioning approach is a secondary contribution. The incompressible results from SU2 compare very favorably to analytical solutions, well-established codes, and experiment in all cases investigated. V&V is critically important in order to build a high degree of confidence in the implementation and to overcome skepticism concerning accuracy or quality of the solutions produced by open-source software. The results here are reproducible with open data, and the source code is available for all potential future users and developers. This article can be considered a reference to build upon for simulating incompressible flows in SU2. Future work is already underway for including more complex fluid models (thermodynamics, chemistry, etc.) as well as pursuing unsteady flows with dynamic grid motion.

References

[1] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comput. Phys.*, 72(2):277–298, October 1987.

[2] Y.-H. Choi and C.L. Merkle. The application of preconditioning in viscous flows. *Journal of Computational Physics*, 105(2):207 – 223, 1993.

[3] E. Turkel, V. N. Vatsa, and R. Radespiel. Preconditioning methods for low-speed flows. *AIAA Paper 1996-2460*, 1996.

[4] C. Viozat. Implicit upwind schemes for low mach number compressible flows. Technical report, Institut National De Recherche En Informatique En Automatique, 1997.

[5] E. Turkel. Preconditioning techniques in fluid dynamics. Technical report, School of Mathematical Sciences, Tel-Aviv University, 1999.

[6] J. M. Weiss and W. A. Smith. Preconditioning applied to variable and constant density flows. *AIAA Journal*, 33(11):2050–2057, 1995.

[7] Peter M. Sockol. Multigrid solution of the navier–stokes equations at low speeds with large temperature variations. *Journal of Computational Physics*, 192(2):570 – 592, 2003.

[8] Y. Colin, H. Deniau, and J.-F. Boussuge. A robust low speed preconditioning formulation for viscous flow computations. *Computers & Fluids*, 47(1):1 – 15, 2011.

[9] R. G. Rehm and H. R. Baum. Equations of motion for thermally driven, buoyant flows. *Journal of Research of the National Bureau of Standards*, 83(3):297–308, 1978.

[10] Bernhard Müller. Low-mach-number asymptotics of the navier-stokes equations. *Journal of Engineering Mathematics*, 34(1):97–109, 1998.

[11] F. Nicoud. Conservative high-order finite-difference schemes for low-mach number flows. *Journal of Computational Physics*, 158(1):71 – 97, 2000.

[12] J. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195:6011–6045, 2006.

[13] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 137:118–125, 1997.

[14] U. Riedel. A finite volume scheme on unstructured grids for stiff chemically reacting flows. *Combustion Science and Technology*, 135(1-6):99–116, 1998.

[15] E. Shapiro and D. Drikakis. Artificial compressibility, characteristics-based schemes for variable density, incompressible, multi-species flows. part i. derivation of different formulations and constant density limit. *Journal of Computational Physics*, 210(2):584 – 607, 2005.

[16] E. Shapiro and D. Drikakis. Artificial compressibility, characteristics-based schemes for variable-density, incompressible,

multispecies flows: Part ii. multigrid implementation and numerical tests. *Journal of Computational Physics*, 210(2):608 – 631, 2005.

[17]D. Drikakis and W. Rider. *High-Resolution Methods for Incompressible and Low-Speed Flows*. Springer-Verlag, 2005.

[18]F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso. Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper 2013-0287*, 2013.

[19]F. Palacios, T. D. Economon, A. C. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, A. S. Padron, B. Tracey, A. Variyar, and J. J. Alonso. Stanford University Unstructured (SU^2): Open-source analysis and design technology for turbulent flows. *AIAA Paper 2014-0243*, 2014.

[20]T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.

[21]F. Palacios, J. J. Alonso, and A. Jameson. Optimal design of free-surface interfaces using a level set methodology. *AIAA Paper 2012-3341*, 2012.

[22]F. Palacios, J. J. Alonso, and A. Jameson. Design of free-surface interfaces using rans equations. *AIAA Paper 2013-3110*, 2013.

[23]O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.

[24]A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[25]F. M. White. *Viscous Fluid Flow*. McGraw Hill Inc., New York, 1974.

[26]D.C. Wilcox. *Turbulence Modeling for CFD*. 2nd Ed., DCW Industries, Inc., La Cañada, CA, 1998.

[27]F.R. Menter. Zonal two equation $k - \omega$, turbulence models for aerodynamic flows. *AIAA Paper 93-2906*, 1993.

[28]P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 1992-0439*, 1992.

[29]A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. Number AIAA Paper 1991-1596, 1991.

[30]P.A. Thompson. *Compressible-fluid dynamics*. Control Systems Engineering Series. McGraw-Hill, 1972.

[31]K. A. Hoffman, S. T. Chiang, S. Siddiqui, and M. Papadakis. *Fundamental Equations of Fluid Mechanics*. Engineering Education Sysiem, Witchita, KS, 1996.

[32]T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. von Karman Institute. 25th Computational Fluid Dynamics Lecture Series, March 1994.

[33]A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23 of *Springer series in computational mathematics*. Springer-Verlag Berlin Heidelberg New York, 1997.

[34]Antony Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37(2):197 – 243, 2001.

[35]R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Univesity Press, 2002.

[36]P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg New York, 2000.

[37]A. Jameson. Analysis and design of numerical schemes for gas dynamics 1 artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *RIACS Technical Report 94.15, International Journal of Computational Fluid Dynamics*, 4:171–218, 1995.

[38]A. Jameson. Analysis and design of numerical schemes for gas dynamics 2 artificial diffusion and discrete shock structure. *RIACS Report No. 94.16, International Journal of Computational Fluid Dynamics*, 5:1–38, 1995.

[39]E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*. Springer-Verlag, 1999.

[40]A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Paper 1981-1259*, 1981.

[41]B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

[42]T. J. Barth and D. Jespersen. The design and application of upwind schemes on unstructured grids. *AIAA Paper 89-0366*, 1989.

[43]V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper 1993-0880*, 1993.

[44]D. J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal*, 28(2):213–221, 1990.

[45]D. J. Mavriplis and A. Jameson. Multigrid solution of the Navier-Stokes equations on triangular meshes. *AIAA Journal*, 28(8):1415–1425, 1990.

[46]J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, Oxford, 2005.

[47]J. M. Weiss, J. P. Maruszewski, and A. S. Wayne. Implicit solution of the Navier-Stokes equation on unstructured meshes. *AIAA Paper 1997-2103*, 1997.

[48]P. Eliasson. Edge, a Navier-Stokes solver for unstructured grids. Technical Report FOI-R-0298-SE, FOI Scientific Report, 2002.

[49]F. Palacios, T. D. Economon, A. D. Wendorff, and J. J. Alonso. Large-scale aircraft design using SU2. *AIAA Paper 2015-1946*, 2015.

[50]T. D. Economon, F. Palacios, and J. J. Alonso. Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53(9):2437–2453, 2015.

[51]T. Albring, M. Sagebaum, and N. R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. *AIAA Paper 2015-3240*, 2015.

[52] T. Albring, M. Sagebaum, and N.R. Gauger. Efficient aerodynamic design using the discrete adjoint method in SU2. *AIAA Paper 2016-3518*, 2016.

[53] Max Sagebaum, Tim Albring, and Nicolas R. Gauger. High-Performance Derivative Computations using CoDiPack. *arXiv preprint arXiv:1709.07229*, 2017.

[54] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[55] J. A. Samareh. A survey of shape parameterization techniques. Technical Report NASA/CP-1999-209136, NASA, 1999.

[56] R. P. Dwight. Robust mesh deformation using the linear elasticity equations. Proceedings of the Fourth International Conference on Computational Fluid Dynamics (ICCFD), pages 401–406, 2006.

[57] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[58] J.D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill series in aeronautical and aerospace engineering. McGraw-Hill Education, 2016.

[59] C. Rumsey and et al. VERIF/2DZP: 2D Zero Pressure Gradient Flat Plate Verification Case - Intro Page. World Wide Web electronic publication, 2018.

[60] C. Rumsey and et al. 2DN00: 2D NACA 0012 Airfoil Validation Case. World Wide Web electronic publication, 2018.

[61] N. Gregory and C. L. O'Reilly. Low-speed aerodynamic characteristics of naca 0012 aerofoil sections, including the effects of upper-surface roughness simulation hoar frost. *NASA R & M 3726*, 1970.

[62] C. Rumsey and et al. VERIF/3DB: 3D Bump-in-channel Verification Case - Intro Page. World Wide Web electronic publication, 2016.

[63] Jeongyoung Park, Kiyoung Kwon, and Haecheon Choi. Numerical solutions of flow past a circular cylinder at reynolds numbers up to 160. *KSME International Journal*, 12(6):1200–1205, Nov 1998.

[64] S. Sen, S. Mittal, and G. Biswas. Steady separated flow past a circular cylinder at low reynolds numbers. *Journal of Fluid Mechanics*, 620:89–119, 2009.