Discrete adjoint design optimization

Cristopher Morales Ubal

c.m.ubal@gmail.com

January 20, 2025

Introduction

The following document summarizes the discrete adjoint design optimization reported in [9], which was written by the same author of the present document. The purpose of this document is to explain the theoretical and modelling approach used in SU2 when multicomponent flows are simulated.

1 Governing equations

The flow simulations are assumed to be three dimensional, steady and incompressible. The Computational Fluid Dynamics (CFD) solution is obtained by solving the Navier-Stokes equations (N-S) which correspond to transport equations for the flow (1), (2) and species mass fraction (3) for the conservative variables $\mathbf{U} = [\rho, \rho\vec{v}, \rho Y_i]$, and are given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{1}$$

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot (\rho \vec{v}\vec{v}) + \nabla p - \nabla \cdot \boldsymbol{\tau} = 0 \tag{2}$$

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot (\rho Y_i \vec{v}) - \nabla \cdot (\rho D_i \nabla Y_i) = 0 \quad , \forall i = 1, \ldots, N \tag{3}$$

where $\vec{v}$ is the velocity field, ρ is the density, p is the pressure, D_i is the mass diffusivity of the species i into the mixture, and Y_i is the mass fraction of species i. As the N species equations in (3) together with the continuity equation (1) are over-determined, only $N-1$ species equations are solved from (3). The last species mass fractions is obtained from:

$$Y_N = 1 - \sum_{i=1}^{N-1} Y_i \tag{4}$$

Likewise, the compressible equation for the shear stress tensor $\boldsymbol{\tau}$ is used and given as follows:

$$\boldsymbol{\tau} = \mu \left(\nabla \vec{v} + (\nabla \vec{v})^T - \frac{2}{3} (\nabla \cdot \vec{v}) \mathbf{I} \right) \tag{5}$$

The fluid properties ρ, μ and D_i and how they are computed based on the fluid-composition will be explained in the next section.

For modelling turbulence, we solve the Reynolds-Averaged Navier Stokes equations (RANS) considering the Boussinesq hypothesis[13], where turbulence can be modelled as an increase in viscosity. Therefore, the viscosity becomes an effective viscosity given by:

$$\mu_{\text{eff}} = \mu + \mu_t \tag{6}$$

where μ and μ_t and the laminar and turbulent viscosities of the gas mixture, respectively. The same approach is adopted for the mass diffusivity which is computed by:

$$D_{i,\text{eff}} = D_i + \frac{\mu_t}{\text{Sc}_\text{t}} \tag{7}$$

where $\mathrm{Sc_t}$ is the turbulent Schmidt number taken as 0.7 [10].
For computing μ_t, the Shear Stress Transport ($k-w$ SST) model of Menter [8] is used given its well-extensive validation through three-dimensional incompressible steady flows. In this model, two additional transport equations are solved for the turbulent dissipation rate w and turbulent kinetic energy k, which are given below:

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \vec{v}) = \nabla \cdot \left[(\mu + \sigma_k \mu_t) \nabla k \right] + P_k - \beta^* \rho w k \tag{8}$$

$$\frac{\partial \rho w}{\partial t} + \nabla \cdot (\rho w \vec{v}) = \nabla \cdot \left[(\mu + \sigma_w \mu_t) \nabla k \right] + \frac{\gamma}{\nu_t} P_k - \beta^* \rho w^2 + 2 (1 - F_1) \frac{\rho \sigma_{w,2}}{w} \nabla k \cdot \nabla w \tag{9}$$

where σ_k, σ_w, β^*, γ and $\sigma_{w,2}$ are closure coefficient and auxiliary relations, F_1 and F_2 are blending coefficients, and P_k is the production term. More details about these terms are given in [8].
Although the N-S equations are written in the conservative form, inside SU2 the working variables correspond to the primitive variables $\mathbf{V} = [p, \vec{v}, Y_i]$. Moreover, the discretization of the set of equations (1), (2), (3), (8) and (9) forms the system which will be referred as $R(V) = 0$ when the Adjoint Method will be explained in the section 3. Finally, the N-S equations are solved using a preconditioning approach for Low-Mach number incompressible flows, which are discretized using a finite volume method on a vertex-based median-dual grid [5].

2 Mixing model

The composition-dependent model is based on the mixing laws for incompressible ideal gases. First, the mean molecular weight (W) is determined as follows[10]:

$$W = \left[\sum_{i=1}^{N} \frac{Y_i}{W_i} \right]^{-1} \tag{10}$$

where W_i and Y_i are the molecular mass and mass fraction of species i, respectively.
Adopting the low-mach number approximation [3], the thermodynamic pressure can be decoupled from the governing equations (2) and can be set constant. Then, using (10), the density is computed as a mass fraction average given by [10]:

$$\rho = \frac{p_{\mathrm{op}}}{RT/W} \tag{11}$$

where p_{op} is the thermodynamics (or operating) pressure, set as 101325[Pa], T is the temperature and R is the universal gas constant which value is 8.314[J / (mol· K)].
Similarly, the heat capacity at constant pressure and volume are computed as a mass fraction average of the individual species heat capacities [10]. They are given as follows:

$$C_p = \sum_{i=1}^{N} C_{p,i} Y_i \tag{12}$$

$$C_v = C_p - \frac{R}{W} \tag{13}$$

where $C_{p,i}$ is the heat capacity at constant pressure of species i.
Two different methods have been implemented in SU2 for computing the viscosity of a mixture μ. The first one is given by Davidson's method [4], which is based on the transfer of momentum between molecules during collision and the ability of a fluid of being transported. These two assumptions are used in order to estimate the mixture viscosity of a gas using the momentum fraction y_i and the fluidity f. Hence, the mixture viscosity is computed as follows:

$$\mu = \frac{1}{f} \tag{14}$$

$$f = \sum_{i}^{N} \sum_{j}^{N} \frac{y_i y_j}{\sqrt{\mu_i \mu_j}} E_{ij}^{A} \tag{15}$$

$$E_{ij} = \frac{2\sqrt{W_i W_j}}{W_i + W_j} \tag{16}$$

$$y_i = \frac{X_i \sqrt{W_i}}{\sum_{j=1}^{N} X_j \sqrt{W_j}} \tag{17}$$

where $A[-]$ is an empirical constant equal to 0.375 and $E_{i,j}$ is the mean efficient for momentum transfer between two spherical bodies given by (16). Likewise, X_i is the mole fraction of species i given by:

$$X_i = \frac{W}{W_i} Y_i \tag{18}$$

The second method is given by the Wilke's method [13]. This method computes the laminar viscosity for a gas mixture as a mole fraction average given as follows:

$$\mu = \sum_{i=1}^{N} \frac{X_i \mu_i}{\sum_{j=1}^{N} X_j \phi_{ij}} \tag{19}$$

where, μ_i are the laminar viscosities of species i and ϕ_{ij} are binary coefficients computed by:

$$\phi_{ij} = \frac{\left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{1/2} \left(\frac{W_j}{W_i}\right)^{1/4}\right]^2}{\left[8\left(1 + \frac{W_i}{W_j}\right)\right]^{1/2}} \tag{20}$$

Similarly, following the Wilke's method approach, the thermal conductivity for a gas mixture is computed as follows [11]:

$$k = \sum_{i=1}^{N} \frac{X_i k_i}{\sum_{j=1}^{N} X_j \phi_{ij}} \tag{21}$$

where k_i is the thermal conductivity of species i and ϕ_{ij} is computed by (20).
Finally, the mass diffusivity coefficient, D_i, which measures the fluid transport process due to molecular motion is based on the constant Lewis number assumption. The Lewis number, Le_i, compares the heat diffusivity with respect to mass diffusivity[10]. This is defined as follows:

$$\mathrm{Le_i} = \frac{Sc_i}{Pr} \tag{22}$$

where Sc_i and Pr are the Schmidt and Prandtl number, respectively. As $Pr = \mu \cdot C_p/\lambda$ and $Sc_i = \nu/D_i$, then from (22) the diffusivity coefficient can be computed by:

$$D_i = \frac{k}{\rho C_p Le_i} \tag{23}$$

A widely used assumption is the Unity Lewis number ($Le_i = 1$), where the thermal and molecular diffusion can be assumed equal. Then, equation (23) can be simplified to:

$$D = \frac{k}{\rho C_p} \tag{24}$$

3 Adjoint method

Within SU2, a discrete adjoint (DA) solver has already been implemented using Algorithmic differentiation (AD) [6] via the C++ library Code-Differentiation Package CoDiPack [12]. This DA approach is a highly efficient sensitivity methodology that allows for the computational of sensitivities of an objective function such as surface variance, drag, and others at the computational cost that is approximately equal to performing an additional simulation of the primal problem (CFD simulation) [1, 2]. This enables the use of gradient-based optimization with a large

number of design variables. The DA solver has been constructed in a general way, allowing extensions inside SU2 as the one presented in the paper [9] without the need to manually differentiate new variables. Here, we are going to briefly explain the DA approach. First, we can write the following optimization problem:

$$\min_{\alpha} \quad J(V(\alpha), X(\alpha)) \tag{25}$$

$$\text{subject to } V(\alpha) = G\left(V(\alpha), X(\alpha)\right) \ , \tag{26}$$

$$X(\alpha) = M(\alpha) \ , \tag{27}$$

where $V(\alpha)$ and $X(\alpha)$ are the primitive variables and mesh state as functions of the design variables α, respectively. Likewise, $J(V(\alpha), X(\alpha))$ is the objective function that we aim to minimize. The constraints are the discretized Navier-Stokes equations $R(V) = 0$ written as a fixed point equation (26) and the mesh state (27), which relates the mesh to the design variables allowing its deformation. This equation corresponds to the linear elasticity equations. Additionally, It can be proven using the Banach fixed-point theorem that, written the N-S equations as fixed point operator, enables to inherit the convergence of the flow solver and use the same approximate Jacobian from the primal solver[1],[2].
Subsequently, we can compute the Lagrangian associated to the optimization problem (25), (26), (27) as follows:

$$L\left(\alpha, V, X, \Psi, \Phi\right) = J(V, X) + \Psi^T \left(G(V, X) - V\right) + \Phi^T \left(M - X\right) \ , \tag{28}$$

where Ψ and Φ are the so-called adjoint variables of the optimization problem.
It must be noted in (28) that the terms in brackets represent the optimization constraints. In this way, using the change rule, we differentiate (28) with respect to the design variables α:

$$\frac{\mathrm{d}L}{\mathrm{d}\alpha}^T = \frac{\partial J}{\partial V}\frac{\partial V}{\partial \alpha} + \frac{\partial J}{\partial X}\frac{\partial X}{\partial \alpha} + \Psi^T \left(\frac{\partial G}{\partial V}\frac{\partial V}{\partial \alpha} + \frac{\partial G}{\partial X}\frac{\partial X}{\partial \alpha} - \frac{\partial V}{\partial \alpha}\right) + \Phi^T \left(\frac{\partial M}{\partial \alpha} - \frac{\partial X}{\partial \alpha}\right) \tag{29}$$

Then, reordering terms, we have:

$$\frac{\mathrm{d}L}{\mathrm{d}\alpha}^T = \Phi^T \frac{\partial M}{\partial \alpha} + \left(\frac{\partial J}{\partial V} + \Psi^T \frac{\partial G}{\partial V} - \Psi^T\right)\frac{\partial V}{\partial \alpha} + \left(\frac{\partial J}{\partial X} + \Psi^T \frac{\partial G}{\partial X} - \Phi^T\right)\frac{\partial X}{\partial \alpha} \tag{30}$$

As we desire to avoid the expensive computations of jacobians $\frac{\partial V}{\partial \alpha}$ and $\frac{\partial X}{\partial \alpha}$ in (30), we can choose Ψ and Φ such that the terms in brackets are zero. This leads to the adjoint equations shown below:

$$\Psi^T = \frac{\partial J}{\partial V} + \Psi^T \frac{\partial G}{\partial V} \tag{31}$$

$$\Phi^T = \frac{\partial J}{\partial X} + \Psi^T \frac{\partial G}{\partial X} \tag{32}$$

Thus, by solving the fixed point equation (31) and equation (32), we can compute the sensitivities as follow:

$$\frac{\mathrm{d}J}{\mathrm{d}\alpha}^T = \frac{\mathrm{d}L}{\mathrm{d}\alpha}^T = \Phi^T \frac{\partial M}{\partial \alpha} \tag{33}$$

Finally, the shape design is done using the SciPy SLSQP optimizer [7].

References

[1] Tim Albring, Max Sagebaum **and** Nicolas R Gauger. "Efficient aerodynamic design using the discrete adjoint method in SU2". **in***17th AIAA/ISSMO multidisciplinary analysis and optimization conference*: 2016, **page** 3518. URL: https://api.semanticscholar.org/CorpusID:124364739.

[2] Tim A. Albring, Max Sagebaum **and** Nicolas R. Gauger. "Development of a Consistent Discrete Adjoint Solver in an Evolving Aerodynamic Design Framework". **in***16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*: DOI: 10.2514/6.2015-3240. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2015-3240.

[3] John D. Buckmaster. *The Mathematics of Combustion*. **byeditor**John D. Buckmaster. Society for Industrial **and** Applied Mathematics, 1985. DOI: 10.1137/1.9781611971064.

[4] T A Davidson. "A simple and accurate method for calculating viscosity of gaseous mixtures". **in**Report of investigations: (**january** 1993). URL: https://www.osti.gov/biblio/6129940.

[5] Thomas D. Economon. "Simulation and Adjoint-based Design for Variable Density Incompressible Flows with Heat Transfer". **in***2018 Multidisciplinary Analysis and Optimization Conference*: (). DOI: 10.2514/6.2018-3111. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-3111.

[6] Andreas Griewank. "A mathematical view of automatic differentiation". **in***Acta Numerica 2003*: **byeditor**ArievEditor Iserles. **volume** 12. Acta Numerica. Cambridge University Press, 2003, **pages** 321–398. DOI: 10.1017/CBO9780511550157.005.

[7] Eric Daniel Jones, Travis E. Oliphant **and** Pearu Peterson. "SciPy: Open Source Scientific Tools for Python". **in**2001: URL: https://api.semanticscholar.org/CorpusID:215874460.

[8] F. R. Menter. "Two-equation eddy-viscosity turbulence models for engineering applications". **in***AIAA Journal*: 32.8 (1994), **pages** 1598–1605. DOI: https://doi.org/10.2514/3.12149.

[9] Cristopher Morales Ubal **andothers**. "Adjoint-based design optimization of a Kenics static mixer". **in***Results in Engineering*: 21 (2024), **page** 101856. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2024.101856. URL: https://www.sciencedirect.com/science/article/pii/S2590123024001099.

[10] T. Poinsot **and** D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2005. ISBN: 9781930217102. URL: https://books.google.nl/books?id=cqFDkeVABYoC.

[11] B.E. Poling, J.M. Prausnitz **and** J.P. O'Connell. *The Properties of Gases and Liquids 5E*. McGraw Hill professional. McGraw-Hill Education, 2000. ISBN: 9780071499996. URL: https://books.google.nl/books?id=9tGclC3ZRX0C.

[12] Max Sagebaum, Tim Albring **and** Nicolas R. Gauger. *High-Performance Derivative Computations using CoDiPack*. 2017. DOI: 10.48550/ARXIV.1709.07229.

[13] C. R. Wilke. "A Viscosity Equation for Gas Mixtures". **in***The Journal of Chemical Physics*: 18.4 (1950), **pages** 517–519. DOI: 10.1063/1.1747673.