



Motivations and methods of verification for high-order RANS solvers and solutions

Farshad Navah* and Siva Nadarajah†

Computational Aerodynamics Group

Department of Mechanical Engineering, McGill University

Montreal, QC H3A 2S6 Canada

High-order accurate discretization methods applied to computational fluid dynamics (CFD) have recently benefited from the increasing interest of the research community. The potential of high-order methods as an efficient approach for the control and reduction of the discretization error has been established for inviscid and laminar flows via analytical and canonical test cases. For turbulent flows modelled by the Reynolds-averaged Navier-Stokes (RANS) equations however, the absence of analytical solutions hinders the verification of the high-order accuracy for a given scheme/code. Instead, to examine the soundness of the implementation, the common practice relies on validation, i.e., the comparison of the solution of benchmark cases with that from experiment and/or established solvers. Nevertheless, due to the stringent mesh requirements, the absence of global metrics and the convergence difficulties of realistic RANS cases, the validation alone is incapable of providing an exhaustive and reliable diagnostic on the correctness of the implementation and consequently on the benefit of a given solver in delivering higher orders of accuracy for this class of problems. In this work, we demonstrate the application of a full verification methodology to high-order RANS-modelled turbulence solvers and solutions. Without loss of generality, the demonstration is carried on a framework composed of a high-order discretization of the Spalart-Allmaras turbulence model by the correction procedure via flux reconstruction (CPR) scheme. First, the solver is verified via the method of manufactured solutions (MMS). Then, the solution is verified for the prediction of turbulent boundary layer on a flat plate.

I. Introduction

Most current state-of-the-art industrial CFD solvers are 2^{nd} order accurate at best. This class of solvers, called *low-order*, are prone to large discretization errors that compromise the reliability of CFD for the estimation of engineering quantities such as lift and drag coefficients. High-order accurate methods have the potential to produce lower discretization errors for the same number of degrees of freedom as their low-order counterparts and thus allow a larger flexibility in mesh generation. Furthermore, high-order methods enable the adaptation of both h , the element size, and P , the polynomial degree, that in combination, offer the full potential for optimal error control. While the efficacy of high-order methods in reducing the discretization errors is demonstrated and established for inviscid and laminar flows, the high-order accuracy of the solutions of the RANS equations, the most popular approach towards turbulence modelling, yet needs further verification.

Verification is a mathematical exercise that examines the correctness of the implementation and solution of the model equations. *Validation*, on the other hand, questions the appositeness of the model to represent the physical phenomena of interest. The *code verification* is furthermore distinguished from the *solution verification*. The former is an exercise performed only once for a given version of a code and investigates the soundness of the implementation via the *evaluation* of numerical errors, requiring the exact solution to be

*Graduate Student, Student Member AIAA

†Associate Professor, AIAA Associate Fellow

known; whereas the latter refers to the *estimation* of numerical errors that occur in a given computational solution. We further expand these concepts in the following lines.

Assuming the round-off and iterative convergence errors to be negligible, the major source of numerical error is the discretization error that is the difference between the discrete and continuous solutions of a given mathematical model (governing equations). If the discretization is uniformly refined for problems exempt of singularities, the discrete solution is expected to converge to the continuous solution with a rate dictated by the order of accuracy of the scheme. This convergence is furthermore subject to stability and consistency assumptions of the Lax equivalence theorem (strictly speaking, the Lax theorem is proven for linear problems only and "supposed" valid for others by extension). A monotonic or asymptotic convergence (constant rate) is reached as soon as the refinement is sufficient for the leading term to preponderate over the higher-order terms of the truncation error. With the analytical solution in hand, one can evaluate the exact amount of discretization error and hence compare the theoretical (expected) and the actual (observed) rates of convergence and establish whether the solver is verified (concordant rates) or not. A verified code can be trusted in producing discrete solutions that approach the continuous solution with a known rate and recover exactly the continuous solution in the limit of infinite degrees of freedom. Code verification is therefore crucial in a context of uncertainty quantification for engineering applications. Furthermore, the implementation of high-order discretization schemes is often more complex and their solution sometimes more costly compared to low-order methods. This extra effort is solely justified by the higher order of accuracy and the ensuing lower discretization error, supplied by high-order methods, for a given computational budget. In this regard, demonstrating the achievement of the expected higher accuracy for a given scheme/implementation is an imperative task especially as the higher-order methods are more sensitive to the most minute implementation bugs as will be demonstrated in this work.

Computing the rate of convergence necessitates the knowledge of the exact solution. Nevertheless, sufficiently complex analytical solutions, capable of exercising all the terms of the governing equations in the targeted flow regime, are not always available. An elegant remedy is supplied by the MMS that enables the code verification in such cases by adding adequate forcing terms to the discrete equation, resulting in a modified equation for which the solution is defined *a priori*. Thus, by extending the code verification to arbitrary sets of governing equations and flow regimes, the MMS constitutes the most rigorous methodology for scrutinizing the ability of a solver in producing the expected performance in terms of accuracy/effort ratio. However, the MMS requires code modification to accommodate the addition of the forcing functions and manufactured values of boundary conditions, a delicate task that is itself bug-prone if not conducted with care. This is a major reason behind the rather disregarded potential of the MMS in code verification amongst CFD code developers who instead opt for acquiring a certain level of confidence in the code by comparing its solutions on practical test cases with values obtained by renown CFD solvers or by experiment. This in fact represents a minimalistic level of verification for multiple reasons. For example, a bug might have a negligible impact on the order of magnitude of a given quantity while it rather prevents an improvement of its accuracy as the mesh is refined. This would be of a lesser impact on low-order codes due to their rather large amount of numerical diffusion which is itself usually an upper bound on the accuracy, hiding the impact of minor bugs on coarse and moderately fine grids. Another example is the presence of inaccuracies on the reference values themselves, either due to uncertainty in experimental measurements or to the presence of a bug in the reference CFD solvers. Furthermore, differences in the implementation of the boundary conditions between the reference solver and the solver being examined would result in slight divergences in the result that would restrict the scope of the direct comparison. It could also happen that a given bug affects some problems less and some others more, hence making its detection by this comparative methodology harder. Another less common approach is to use extrapolation-based error estimation methods to acquire an approximation of the exact solution for integrated outputs such as lift and drag coefficients and use such estimates, called *truth values*, to evaluate the error and the order of accuracy on a series of grids for simple airflow cases. However, this method is also deficient in providing a comprehensive diagnostic on the soundness of the scheme implementation for a few reasons. First, the accuracy of the truth value itself largely relies on the adequacy of the grid to reach the asymptotic range for highly non-linear problems such as turbulent boundary layers. Additionally, this method only provides a diagnostic for local values (integrated values on the wall) and yet, the effect of potential inconsistencies in any of the boundary conditions (such as inlet) on these functional outputs is not clear. Finally, achieving proper residual convergence for highly anisotropic grids is not always an easy task. We could hence conclude that the MMS remains a method of choice and the most rigorous approach for code verification, despite its intricate implementation.

Once the implementation of the scheme is verified, the CFD code could be used in solving practical engineering problems where the exact solution is unavailable. To employ a given CFD solution in the design process, one needs to establish a certain level of confidence by estimating the numerical uncertainty associated with it via solution verification.¹ For this purpose, the continuum solution could be approximated by an extrapolation-based error estimation technique, such as Richardson's, using three consecutive solutions in the asymptotic range obtained on grids with spatially and sequentially constant refinement ratios. This approximation then serves to estimate the discretization error and compute other metrics, such as the grid convergence index (GCI),² that quantify the level of uncertainty in the discrete solution.

The high-order discretization scheme adopted in this work is the CPR,³ that is an extension of the flux reconstruction (FR)⁴ scheme to simplices by a lifting collocation penalty (LCP) technique. Variations of the correction function at the core of this scheme allow to recover a number of prominent compact high-order methods such the spectral difference (SD), the spectral volume (SV) and the discontinuous Galerkin (DG). Hence, the FR/CPR method is regarded as a unifying compact high-order scheme and a representative member of this class. The governing equations are the compressible RANS, equivalent to the Favre-averaged Navier-Stokes (FANS), along with the Spalart-Allmaras (SA) closure model⁵ that is one of the most popular *complete* turbulence models for aerodynamic applications.

The remaining of the paper is organized as follows: we first expose the theoretical background in verification and validation, followed by the detailed presentation of the CPR-RANS-SA solver with regards to the scheme, the governing equations and the boundary conditions. We then dwell on the verification of the high-order implementation via the method of manufactured solutions by explaining the important aspects of this process in details, especially with regards to high-order methods. An example of solution verification for polynomial discretizations of P1 to P3 is finally provided for the benchmark problem of zero-pressure-gradient flat plate from NASA's Turbulence Modelling Resource (TMR). The model validation is omitted in this work since the SA model has already been subject of extensive validation for a wide range of applications by the aerospace community.

II. Theoretical background

In this section, first, the terminology involved in verification and validation (V&V) is completed and defined further, the MMS is introduced then and finally, a short review of the previous works with a focus on verification via the MMS in CFD is presented.

II.A. Terminology in V&V

The Figure 1 illustrates the relation between major concepts of interest under the three themes of *simulation process*, *error sources* as well as *verification and validation*. As a scientific simulation process takes place, errors from various sources slip into its different steps, contaminating incrementally the outcome of the process. Abstractly, as the reality that we aim to capture cascades through a simulation, it diminishes at each step of the process. The role of V&V is hence to ensure that the amount of original reality captured by the simulation is sufficient for the purpose that the simulation is meant to serve, by ideally providing a dependable measurement of the discrepancies. In what follows, we describe more precisely these ideas with reference to Fig. 1.

Any scientific *simulation process* starts from a *reality*, a physical phenomenon that it aims at reproducing. Based on experimental data and previous theoretical knowledge, the relation between various quantities playing a role in the physical process is described by a *conceptual model*, i.e., a series of mathematical equations such as the partial differentials equations (PDEs) of Euler, Navier-Stokes or those of RANS turbulence models. Almost in all cases, these models are a mere, yet hopefully reliable, approximation of the complex nature of the material reality and as such they contain a *modelling error*. In this regard, *model validation* questions how well the physics are approached by the conceptual model. For complex problems such as those encountered in CFD, this is however only considerable once the simulation process has ended and a numerical solution is available. In order to solve the conceptual model, a numerical algorithm, a scheme, is often needed. The application of this numerical recipe to the conceptual model yields a *numerical model* such as the discretization of the RANS-SA system of PDEs by the CPR scheme. As the size of the discrete problem is increased for well-posed and smooth solutions, a numerical model is expected to tend towards the conceptual model with a rate known as the *formal order of accuracy*. A numerical model

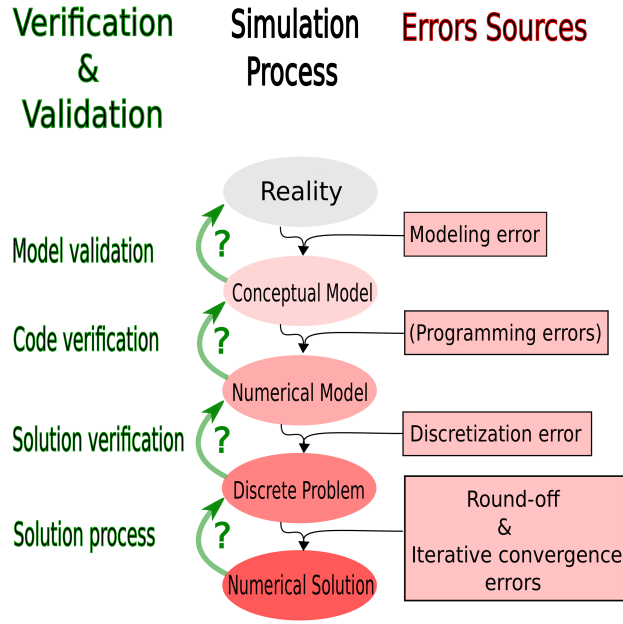


Figure 1: Verification and validation in relation to sources of error in a CFD simulation

with suitable properties such as stability and efficiency is translated to a computer code. Considering the complexity of the numerical model, *programming errors* often occur at this step. Indeed, according to an exhaustive analysis of the quality of scientific computing codes: "There were about 8 serious static faults per 1000 lines of executable lines in C, and about 12 serious faults per 1000 lines in Fortran".⁶ *Code verification* has for purpose to identify and eliminate the mistakes affecting the correspondence between the scientific software and the conceptual model, i.e. the formal order of accuracy. Similarly to model validation, code verification relies on the numerical solution of specific problems. To solve a given problem, the spatial and temporal domains are discretized by a set of points, called *degrees of freedom*, to which the discrete solution is associated. The *discretization error* is the difference between the continuous and discrete solutions and the *solution verification* is the estimation of this error for a given solution. The *solution process* refers to the minimization of discrete residual equations by iterative methods, mandatory for tackling non-linear systems, and by algorithms handling linear algebraic systems. The truncation of real values for representation on computer architectures, by double precision types for example, introduces a *round-off error* that affects the numerical solution by propagating through the discrete equations. On the other hand, a lack of sufficient minimization of the discrete residual equations results in an *iterative convergence error* that imposes a gap between the achieved numerical solution and the actual solution of the numerical model. Both the round-off and iterative convergence errors need to be controlled during the solution process to ensure that these sources of error are minimized such that the discretization error is isolated as the major source of numerical error. This condition enables code and solution verifications to be carried out since they operate only on the discretization error and its rate of reduction for increasing degrees of freedom.

II.B. Verification via the MMS

Consider a linear differential operator, \mathcal{A} , producing a homogeneous output, uniquely when applied to Q^{ex} , a time-invariant and smooth scalar field on a bounded spatial domain, viz.,

$$\mathcal{A}(Q^{\text{ex}}) = 0. \quad (1)$$

For a tessellation of the domain augmented by a polynomial expansion P resulting in a representative resolution size, h , the equivalent discrete equation reads

$$\mathcal{A}^h(Q^h) = \epsilon \approx 0. \quad (2)$$

where \mathcal{A}^h and Q^h are respectively the discrete operator and solution and ϵ is a finite precision representation of zero such as the *machine precision*. The truncation error is defined as

$$\Theta_{\mathcal{A}}^h \equiv \mathcal{A}^h(Q^{\text{ex}}) - \mathcal{A}(Q^{\text{ex}}) = \mathcal{A}^h(Q^{\text{ex}}). \quad (3)$$

According to the Lax equivalence theorem, if \mathcal{A}^h is stable, i.e., its variations with regards to finite variations of Q^h are bounded, and if \mathcal{A}^h is furthermore consistent, i.e.,

$$\lim_{h \rightarrow 0} \Theta_{\mathcal{A}}^h = \mathcal{A}(Q^{\text{ex}}) = 0,$$

sufficient conditions are fulfilled for the discrete solution to be convergent, i.e.,

$$\lim_{h \rightarrow 0} Q^h = Q^{\text{ex}},$$

and hence the discretization error, $\mathcal{E}_Q \equiv (Q^h - Q^{\text{ex}})$, tends to zero with $\|\mathcal{E}\| \sim \mathcal{O}(h^o)$ where $\|\cdot\|$ is a measurement in an appropriate norm and o is the formal rate of convergence (order of accuracy).

Now, let's apply the operator \mathcal{A} to an arbitrary smooth scalar field, $Q^{\text{MS}} \neq Q^{\text{ex}}$. Obviously,

$$\mathcal{A}(Q^{\text{MS}}) \neq 0.$$

But, Eq. (1) can be modified by the addition of a suitable *forcing function*, $\mathcal{F}(Q^{\text{MS}})$, such that

$$\mathcal{R}(Q^{\text{MS}}) \equiv \mathcal{A}(Q^{\text{MS}}) - \mathcal{F}(Q^{\text{MS}}) = 0. \quad (4)$$

where \mathcal{R} designates the modified operator. An immediate choice of forcing function is $\mathcal{F}(Q^{\text{MS}}) = \mathcal{A}(Q^{\text{MS}})$. The modified counterparts of the discrete equation (2) and the truncation error (3) respectively are

$$\mathcal{R}^h(Q^h) = \mathcal{A}^h(Q^h) - \mathcal{F}(Q^{\text{MS}}) = \epsilon \approx 0, \quad (5)$$

and

$$\Theta_{\mathcal{R}}^h \equiv \mathcal{R}^h(Q^{\text{MS}}) - \mathcal{R}(Q^{\text{MS}}) = \mathcal{R}^h(Q^{\text{MS}}), \quad (6)$$

via Eq. (4).

The stability and consistency of \mathcal{R}^h follow from those of \mathcal{A}^h and hence the convergence to the manufactured solution is established as

$$\lim_{h \rightarrow 0} Q^h = Q^{\text{MS}},$$

and $\|\mathcal{E}_Q\| = \|Q^h - Q^{\text{MS}}\| \sim \mathcal{O}(h^o)$ is expected. The reader is referred to the literature for further information on the value of the formal order of accuracy in finite difference and finite volume frameworks. As for continuous finite element and discontinuous compact variational methods, we refer to the comprehensive analyses in.^{7,8} A generic result for a DG-like method applied to linear (and by extension non-linear) advection-diffusion problems is that the expected order in L_1 , L_2 and L_∞ norms is $o = P + 1$, i.e., $\mathcal{E} \sim \mathcal{O}(h^{P+1})$, whereas it rather is $o = P$ in H_1 norm. The definition of each norm is provided in Appendix B.

In this work, the forcing functions are created by Python scripts via symbolic module Sympy and the MMS is implemented by applying a small number of point-wise adjustments to the solver. The focus is here on the verification of the steady-state RANS equations since a semi-discrete scheme is considered. In other words, the integration in time is segregated from the spatial discretization and could hence be verified in an independent fashion.

II.C. Review of the literature

The V&V in CFD originates from pioneering and comprehensive contributions of Roache,^{2,9} Oberkampf et al.,⁶ Salari and Knupp¹⁰ as well as Roy et al.^{1,11} The applications of the MMS via trigonometric manufactured solutions are reported for Euler and Navier-Stokes solvers,^{10–16} for RANS solvers,^{13,17–19} for fluid-structure interaction²⁰ and more recently for multiphase flows.²¹ Physically realistic manufactured solutions for the verification of RANS-based turbulence models, that mimic a turbulent boundary layer in the vicinity of no-slip wall, are introduced by Eça et al.^{22–26} as well as by Oliver and colleagues.¹⁹ Finally, a library for verification via MMS is presented in.²⁷ The mentioned applications involve low-order finite volume and finite element codes. Applications of the MMS to the verification of high-order spatial discretizations are found in^{28–31} for discontinuous Galerkin/FR/CPR methods and in³² for mixed finite differences methods.

III. Compact high-order numerical method

III.A. Governing equations

The governing equations are the steady-state, compressible RANS equations assuming a Newtonian and calorically perfect gas along with the conservative form of the revised SA model of turbulence,⁵ that has proved effective for the simulation of aerodynamic flows. The revised SA model supplements the original model by conditional variations of some of its terms in order to provide enhanced numerical stability despite the occurrence of negative values of the model's working variable, especially in presence of coarse spatial discretizations of the boundary layer in high-order solutions.

We cast the governing PDEs into the following compact form for general unsteady advection-diffusion models:

$$\partial_t(Q_k) + \partial_j(F_{kj}^{inv}) - \partial_j(F_{kj}^{vis}) = S_k, \quad (7)$$

where Q_k is a state variable and the solution of the k^{th} PDE with $k \in [1..N_{eq}]$ where $N_{eq} = N_d + 3$ is the number of equations based on N_d space dimensions; F_{kj}^{inv} and F_{kj}^{vis} respectively express the k^{th} inviscid (advective) and viscous (diffusive) fluxes for $j \in [1..N_d]$, S_k refers to the source term for equation k and we use the Einstein's summation convention for repeated indices. The expanded form of the conservation laws is retrieved by substituting the following expressions and values in Eq. (7):

- Conservation of mass ($k = 1$)

$$Q_k = \rho, \quad F_{kj}^{inv} = \rho u_j, \quad F_{kj}^{vis} = 0, \quad S_k = 0; \quad (8)$$

- Conservation of momentum ($k = 2..N_d + 1$ and $i = k - 1$)

$$Q_k = \rho u_i, \quad F_{kj}^{inv} = \rho u_j u_i + p \delta_{ij}, \quad F_{kj}^{vis} = \tau_{ij}, \quad S_k = 0; \quad (9)$$

- Conservation of energy ($k = N_d + 2$)

$$Q_k = \rho E, \quad F_{kj}^{inv} = \rho u_j H, \quad F_{kj}^{vis} = u_i \tau_{ij} + \omega_j, \quad S_k = 0; \quad (10)$$

- Transport of the turbulent working variable ($k = N_d + 3$)

$$Q_k = \rho \tilde{\nu}, \quad F_{kj}^{inv} = \rho u_j \tilde{\nu}, \quad F_{kj}^{vis} = \frac{1}{\sigma}(\mu + \rho \tilde{\nu} f_n) \partial_j \tilde{\nu}, \quad S_k = \rho \mathcal{P} - \rho \mathcal{D} + \rho \mathcal{T} \\ + \frac{c_{b2}}{\sigma} \rho \partial_j \tilde{\nu} \partial_j \tilde{\nu} - \frac{1}{\sigma}(\nu + \tilde{\nu} f_n) \partial_j(\rho \partial_j \tilde{\nu}). \quad (11)$$

The quantities appearing in equations (8) through (11) are defined as follows: ρ is the density, $\mathbf{u} \equiv \mathbf{e}_i u_i$ is the velocity vector, E is the total energy per mass defined as $E = e + \frac{1}{2}(u_i u_i)$ where e is the internal energy that for a calorically perfect gas is defined as $e = \frac{R}{\gamma-1}T$ and T is the temperature. The total enthalpy is defined as $H = E + \frac{p}{\rho}$ where R is the gas constant and p is the pressure related to the energy via the ideal gas law as

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2}(u_i u_i) \right), \quad (12)$$

where γ is the specific heat ratio ($\gamma = 1.4$ for air).

In Eq. (9), τ_{ij} are the components of the viscous stress tensor that for compressible Newtonian fluids read

$$\tau_{ij} = 2\mu_{\text{eff}} S_{ij}, \quad \text{with } S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i) - \frac{1}{3}\partial_k u_k \delta_{ij},$$

where μ_{eff} is the effective viscosity, defined as the sum of the dynamic viscosity, μ , and the eddy viscosity, μ_t , viz., $\mu_{\text{eff}} = \mu + \mu_t$. Note that the dynamic viscosity is assumed to be spatially constant throughout this work.

In Eq. (10), $\omega_j = -\lambda_{\text{eff}} \partial_j T$ is the j^{th} component of the heat flux vector where λ_{eff} is the effective thermal conductivity defined as $\lambda_{\text{eff}} = \lambda + \lambda_t$, with $\lambda = \frac{\gamma R}{(\gamma-1)} \frac{\mu}{\text{Pr}}$, the molecular conductivity, and $\lambda_t = \frac{\gamma R}{(\gamma-1)} \frac{\mu_t}{\text{Pr}_t}$, the

eddy conductivity. The laminar and turbulent Prandtl numbers are respectively set to $Pr = 0.72$ (air) and $Pr_t = 0.9$.

Unless specified, the variations of the dynamic viscosity with regards to temperature are represented by the Sutherland's law,

$$\mu = \frac{C_1 T^{3/2}}{T + T_s},$$

where C_1 is a coefficient: $C_1 = 1.461 \times 10^{-6}$ ($kg/(ms\sqrt{K})$), and T_s is the Sutherland's temperature: $T_s = 110.3$ (K). In the SA model, μ_t , the turbulent (eddy) viscosity, is expressed by

$$\mu_t = \rho \nu_t = \begin{cases} \rho \tilde{\nu} f_{v1} & \tilde{\nu} \geq 0, \\ 0 & \tilde{\nu} < 0, \end{cases} \quad (13a)$$

$$(13b)$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \tilde{\nu}/\nu, \quad c_{v1} = 7.1,$$

and $\tilde{\nu}$ is the working variable of the SA model that represents a turbulent kinematic viscosity.

In Eq. (11), the production term, \mathcal{P} , is defined as

$$\mathcal{P} = \begin{cases} c_{b1}(1 - f_{t2})\tilde{s}\tilde{\nu} & \tilde{\nu} \geq 0, \\ c_{b1}(1 - c_{t3})s\tilde{\nu} & \tilde{\nu} < 0, \end{cases} \quad (14a)$$

$$(14b)$$

where $c_{b1} = 0.1355$, $f_{t2} = c_{t3} \exp(-c_{t4} \chi^2)$ with $c_{t3} = 1.2$ and $c_{t4} = 0.5$ is the laminar suppression term, $s = |\varepsilon_{ijk} \partial_j u_k|$ is the vorticity magnitude with ε_{ijk} standing for the Levi-Civita symbol of permutation, and \tilde{s} is the modified vorticity defined as

$$\tilde{s} = \begin{cases} s + \bar{s} & \bar{s} \geq -c_{v2}s, \\ s + \frac{s(c_{v2}^2 s + c_{v3}\bar{s})}{(c_{v3} - 2c_{v2})s - \bar{s}} & \bar{s} < -c_{v2}s, \end{cases} \quad (15a)$$

$$(15b)$$

where

$$\bar{s} = \frac{\tilde{\nu} f_{v2}}{\kappa^2 d_w^2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad c_{v2} = 0.7, \quad c_{v3} = 0.9, \quad \kappa = 0.41,$$

and d_w being the distance to the closest wall. In Eq. (11), the destruction term, \mathcal{D} , is defined as

$$\mathcal{D} = \begin{cases} \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \frac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} \geq 0, \\ -c_{w1} \frac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} < 0, \end{cases} \quad (16a)$$

$$(16b)$$

where

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{b2} = 0.622, \quad \sigma = 2/3, \quad f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}.$$

We refer the reader to⁵ for the full definition of the trip term, \mathcal{T} , in Eq. (11) that serves to mimic the effect of a forced transition. A value of $\mathcal{T} = 0$ is considered throughout this work. The remaining closure functions and constants of the SA model are

$$f_n = \begin{cases} 1 & \tilde{\nu} \geq 0, \\ \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3} & \tilde{\nu} < 0, \end{cases} \quad (17a)$$

$$(17b)$$

$$g = r + c_{w2}(r^6 - r), \quad r = \min \left(\frac{\tilde{\nu}}{\tilde{s} \kappa^2 d_w^2}, r_{lim} \right),$$

$$r_{lim} = 10, \quad c_{n1} = 16, \quad c_{w2} = 0.3, \quad c_{w3} = 2.$$

The original SA model is represented by equations 11, 13a, 14a, 15a, 16a and 17a whereas the modified SA model corresponds to equations 11, 13b, 14b, 15b, 16b and 17b.

III.B. CPR Scheme

To discuss the formulation of the CPR scheme, let's consider a scalar hyperbolic conservation law,

$$\partial_t Q + \partial_j F_j = 0, \quad (18)$$

where Q is the state variable and $\mathbf{F} \equiv \mathbf{e}_q F_q(Q) \in \mathbb{R}^{N_d}$ is a generic flux vector that could be or not function of state derivatives. Multiplying by an arbitrary test function, ϕ , integrating and applying the divergence theorem, the variational formulation of Eq. (18) in Green's form is obtained

$$\int_{\Omega} \partial_t(Q) \phi d\Omega - \int_{\Omega} F_j \partial_j(\phi) d\Omega + \int_{\Gamma} F_j n_j \phi d\Gamma = 0,$$

where Ω is a bounded spatial domain in \mathbb{R}^{N_d} , Γ is its frontier and $\mathbf{n} \equiv \mathbf{e}_q n_q$ is the local unit outward normal. Considering a tessellation, Ξ , of Ω , into a set of continuous and non-overlapping elements, $e_i \in \Xi$, the variational formulation reads

$$\sum_{e_i} \left(\int_{\Omega_{e_i}} \partial_t(Q) \phi d\Omega - \int_{\Omega_{e_i}} F_j \partial_j(\phi) d\Omega + \int_{\Gamma_{e_i}} F_j n_j \phi d\Gamma \right) = 0.$$

Taking into account the inner, Q^- , and the outer, Q^+ , states defined with regards to the direction of \mathbf{n} at a given element boundary, the inter-element coupling of discontinuous solutions can be achieved via a common Riemann flux, $\hat{F}^n(Q^-, Q^+, \mathbf{n}^-, \mathbf{n}^+)$, such that

$$F_j n_j \equiv F^n \approx \hat{F}^n(Q^-, Q^+, \mathbf{n}), \quad (19)$$

since for a well constructed mesh, $\mathbf{n} \equiv \mathbf{n}^- = -\mathbf{n}^+$ where $-$ and $+$ exponents respectively refer to the unit outward normal from the element e_i and from its neighbor at the same point on the interface.

Integrating by parts and applying the divergence theorem once again yields the variational formulation in the divergence form,

$$\sum_{e_i} \left(\int_{\Omega_{e_i}} \partial_t(Q) \phi d\Omega + \int_{\Omega_{e_i}} \partial_j(F_j) \phi d\Omega + \int_{\Gamma_{e_i}} (\hat{F}^n - F^n) \phi d\Gamma \right) = 0.$$

The boundary term can be projected into the element. This is achieved by the following *lifting operator* that provides a correction field, δ , as output

$$\int_{\Omega_{e_i}} \delta \phi d\Omega = \int_{\Gamma_{e_i}} (\hat{F}^n - F^n) \phi d\Gamma. \quad (20)$$

We are interested in a correction that belongs to the space of polynomials of degree N_P or less, viz., $\delta \in \mathbb{P}^{N_P}$ since the solution and the correction on element e_i are discretized by a set of interpolation polynomials φ_l with the property that $\varphi_l \in \mathbb{P}^{N_P}$ on element e_i and identically 0 on the others, such that

$$Q_{e_i} \equiv Q_l \varphi_l, \quad (21)$$

and

$$\delta_{e_i} \equiv \delta_l \varphi_l, \quad (22)$$

where Q_l and δ_l are respectively the state and correction values at the solution point $l \in [1..N_{\text{node}}]$ where N_{node} is the number of solution points per element ($N_{\text{node}} = (N_P + 1)^{N_d}$ for a tensor-product element). Throughout this paper, the notation P is used equivalently to N_P . The short notation, P3, P4, etc. will also signify P=3, P=4, etc. Remark that integrals in Eq. (20) could be performed once on a reference element and stored for use on physical elements, assuming that a valid geometrical mapping exists.

In the case where the flux is a non-linear function of the state, the flux divergence will not necessarily belong to the space \mathbb{P}^{N_P} . We therefore consider the following projection of the flux divergence onto the space \mathbb{P}^{N_P} :

$$\int_{\Omega_{e_i}} \prod_{e_i} (\partial_j F_j) \phi d\Omega = \int_{\Omega_{e_i}} \partial_j F_j \phi d\Omega. \quad (23)$$

where $\prod_{e_i}(\partial_j F_j)$ is the projected flux divergence. The resulting discrete formulation is

$$\sum_{e_i} \int_{\Omega_{e_i}} \left(\partial_t(Q_{e_i}) + \prod_{e_i}(\partial_j F_j) + \delta_{e_i} \right) \phi d\Omega = 0,$$

where all terms are of degree N_P or less. Choosing a proper test space that guarantees solution uniqueness yields

$$\sum_{e_i} \left(\partial_t(Q_{e_i}) + \prod_{e_i}(\partial_j F_j) + \delta_{e_i} \right) = 0, \quad (24)$$

that is the variational formulation in the divergence form of Eq. (18) that could be cast into a purely differential scheme, thus avoiding costly explicit quadratures. In fact, the direct projection of the divergence term via Eq. (23) still requires the use of quadratures. Two more efficient alternatives could be considered:³ the Lagrange polynomial (LP) and the chain rule (CR) approaches.

The LP method consists of interpolating the flux before the application of the divergence operator, as

$$(F_j)_{e_i} \equiv F_{jl} \varphi_l,$$

where F_{jl} is the j^{th} spatial component of the flux evaluated at the solution point l . The projection of the divergence can then be computed as

$$\prod_{e_i}(\partial_j F_j) = \partial_j(F_j)_{e_i} = \partial_j(\varphi_l) F_{jl}.$$

The CR approach employs the flux Jacobian, A , and the spatial derivatives of the state variable, both evaluated at the solution point l of e_i ,

$$A_{jl} \equiv \left(\frac{\partial F_j}{\partial Q} \right)_l,$$

and

$$(\partial_j Q_{e_i})_l = \partial_j(\varphi_l Q_l)_l = (\partial_j \varphi_l)_l Q_l,$$

for a projection onto the space \mathbb{P}^{N_P} that reads

$$\prod_{e_i}(\partial_j F_j) = A_{jl} (\partial_j Q_{e_i})_l.$$

Numerical experiments have shown³ that while the LP is fully conservative, the CR is more accurate at the expense of a slight loss in strict conservation. A fix to retrieve the full conservation of the CR has been proposed in.³³ In this work, the CR and LP methods are used for the evaluation of the divergence of inviscid and viscous fluxes respectively. Furthermore, we choose the Lagrange polynomials as both test functions (ϕ in Eq. (20)) and interpolation functions (φ in Eqs. (21) and (22)), thus retrieving a pre-integrated version of the discontinuous Galerkin method. Finally, we utilize the Gauss-Legendre-Lobatto (GLL) set, also known as Gauss-Lobatto, as solution nodes on quadrangular elements.

III.C. Numerical fluxes

For finite number of degrees of freedom, the scheme described in the precedent section yields discontinuous solutions at element interfaces. The common interface flux of Eq. (19) couples these solutions by taking the following form for the considered governing equations:

$$\hat{F}^n = \mathcal{H}^{inv}(Q^-, Q^+, \mathbf{n}) + \mathcal{H}^{vis}(Q^-, Q^+, \mathbf{n}, \mathbf{e}_q(\partial_q Q)^-, \mathbf{e}_q(\partial_q Q)^+), \quad (25)$$

where \mathcal{H}^{inv} and \mathcal{H}^{vis} are proper numerical fluxes for respectively the inviscid and viscous components of the equations and $(\cdot)^-$ and $(\cdot)^+$ refer to the traces of a given quantity from the inner and outer sides of the interface respectively. The choices of numerical fluxes are further described in what follows.

III.C.1. Inviscid flux

We opt for the Roe's approximate Riemann solver³⁴ as \mathcal{H}^{inv} in Eq. (25) that can be expressed as:

$$\mathcal{H}^{inv} = \frac{1}{2} (F_j^{inv}(Q^-)(n_j) + F_j^{inv}(Q^+)(-n_j) + D), \quad (26)$$

where D is the dissipation added to the central flux, stabilizing it through upwinding.

As we solve the RANS and the SA equations in a fully coupled fashion, the Roe numerical flux is re-derived following Appendix D of³⁵ to account for this coupling. The resulting upwinding contrasts with that of decoupled approaches in which, only the normal velocity at the surface is considered in the dissipation term of the SA common flux. In the coupled upwinding, the dissipation term accounts as well for sound speed and pressure. The coupling is reported³⁵ to enhance the smoothness and robustness of the RANS-SA solutions.

III.C.2. Viscous flux

An intuitive approach for deriving a viscous numerical flux would be to consider a central flux based on the element-wise approximation of $\partial_q Q$, computed as $\partial_q(\varphi_l)Q_l$, such as

$$\mathcal{H}^{vis} = \frac{1}{2} (F_j^{vis}(Q^-, \mathbf{e}_q(\partial_q Q)^-) + F_j^{vis}(Q^+, \mathbf{e}_q(\partial_q Q)^+)) n_j. \quad (27)$$

However, this choice suffers from instabilities for under-resolved elliptic problems due to the singularity of the resulting discrete viscous operator.³⁶ Different approaches have been devised to alleviate this problem that mostly rely on the basic idea of introducing a correction on the solution derivatives that takes into account the data from neighboring elements and penalizes the derivatives for solution jumps at the interface. We consider a representative method of this class that is the second flux of Bassi and Rebay,³⁷ referred to as BR2, which employs an auxiliary vector function, $\boldsymbol{\theta} \equiv \mathbf{e}_q \theta_q$, such that

$$\theta_q = \partial_q Q. \quad (28)$$

Applying the approach described in section III.B to Eq. (28) yields its variational formulation in the divergence form,

$$\int_{\Omega_{e_i}} \theta_q \phi d\Omega - \int_{\Omega_{e_i}} \partial_q(Q) \phi d\Omega = \int_{\Gamma_{e_i}} (\hat{Q} - Q) \phi n_q d\Gamma, \quad (29)$$

where $\hat{Q} = \frac{1}{2}(Q^- + Q^+)$ is the common interface value of the state variable.

By considering a lifting operation such that

$$\int_{\Omega_{e_i}} R_q \phi d\Omega = \int_{\Gamma_{e_i}} (\hat{Q} - Q) \phi n_q d\Gamma, \quad (30)$$

where R_q is a correction field accounting for the data from the neighboring element, the term $\overline{\partial_q Q} \equiv \theta_q = \partial_q Q + R_q$ can be interpreted as a corrected divergence that serves in the computation of the viscous flux, $\mathbf{F}^{vis} \equiv \mathbf{e}_j F_j^{vis}(Q, \mathbf{e}_q \overline{\partial_q Q})$ for the discontinuous interface flux value in Eq. (20) and for flux divergence value in Eq. (24). Note that $\lim_{h \rightarrow 0} R_q = 0$ is expected and Eq. (28) is recovered in the continuum limit. As for the viscous numerical flux in Eq. (25), such a correction for the divergence results in an extended stencil since it couples the elemental solution to the data from neighbors of its immediate neighbors. Nevertheless, considering a segmentation of Γ_{e_i} into N_f non-overlapping and continuous faces, the viscous numerical flux on face $(\Gamma_{e_i})_f$, with $f \in [1..N_f]$, can be evaluated via a partially corrected derivative, $\overline{\partial_q Q} \equiv \partial_q Q + r_q$, as

$$\mathcal{H}^{vis} = \frac{1}{2} (F_j^{vis}(Q^-, \mathbf{e}_q(\overline{\partial_q Q})^-) + F_j^{vis}(Q^+, \mathbf{e}_q(\overline{\partial_q Q})^+)) n_j, \quad (31)$$

where the partial correction, r_q , is the result of the following lifting operation:

$$\int_{\Omega_{e_i}} r_q \phi d\Omega = \int_{(\Gamma_{e_i})_f} (\hat{Q} - Q) \phi n_q d\Gamma. \quad (32)$$

III.D. Boundary conditions

The soundness of a CFD implementation can not be established without the verification of its treatment of boundary conditions (BC). Therefore, in this section, we precisely define the expression of the boundary conditions that will be examined in this work.

III.D.1. Riemann BC

The inviscid boundary conditions for free flows are imposed in the sense of *weak-Riemann* of³⁸ via the numerical flux of Eq. (26) that takes into account the incoming and outgoing characteristics arising from the inner, Q_k^- and the outer, $Q_k^+ = Q_k^{\text{BC}}$, states where Q_k^{BC} is the k^{th} boundary state defined at ghost nodes.

In the case of a manufactured solution we set

$$Q_k^{\text{BC}} = Q_k^{\text{MS}}|_{\Gamma_{\text{BC}}}, \quad (33)$$

where $Q_k^{\text{MS}}|_{\Gamma_{\text{BC}}}$ designates the value of the manufactured solution evaluated at the point of the boundary where the BC is imposed.

It is important to note that for verification purposes, the treatment of the boundary conditions should be the same in the manufactured case as in the general flow problem. Hence, for the general flow problem, one would simply provide the desired boundary state at the ghost node instead of $Q_k^{\text{MS}}|_{\Gamma_{\text{BC}}}$. For a farfield BC for example, the free-stream values, Q_k^∞ , are imposed by setting $Q_k^{\text{BC}} = Q_k^\infty$.

III.D.2. Viscous BC

The boundary condition for the viscous terms is enforced through the numerical flux of Eq. (31), by setting the desired values of the state variables and their derivatives at the ghost nodes via $Q_k^+ = Q_k^{\text{BC}}$ and $(\partial_q Q_k)^+ = (\partial_q Q_k)^{\text{BC}}$, and by taking the common state in Eqs. (30) and (32) to be $\hat{Q} = \frac{1}{2}(Q^- + Q^{\text{BC}})$.

In the case of manufactured solutions, the boundary value of the state is set via Eq. (33) and its derivatives read

$$\partial_q Q_k^{\text{BC}} = (\partial_q Q_k)^{\text{MS}}|_{\Gamma_{\text{BC}}}. \quad (34)$$

In the case of a farfield boundary, the boundary values are set to $Q_k^{\text{BC}} = Q_k^\infty$ and $\partial_q Q_k^{\text{BC}} = (\partial_q Q_k)^-$, i.e., the partially corrected derivatives from the inner side of the boundary are imposed at the ghost nodes.

III.D.3. Wall BC

We consider the adiabatic wall condition, the treatment of which is split into inviscid and viscous steps as

- Inviscid (slip condition): the boundary states of all dependent variables, except those of momentum equations, are set to their inner counterparts, i.e., $Q_k^{\text{BC}} = Q_k^-$ for $k \notin [2..N_d+1]$, whereas the boundary velocity vector is defined by cancelling the impacting component, $(u_i^- n_i^-) \mathbf{e}_q n_q^-$, of the inner velocity, $\mathbf{e}_q u_q^-$, while preserving its tangential component, $\mathbf{e}_q u_q^- - (u_i^- n_i^-) \mathbf{e}_q n_q^-$. We thus obtain

$$Q_k^{\text{BC}} = (\rho u_q)^{\text{BC}} = (\rho u_q)^- - 2((\rho u_i)^- n_i^-) n_q^-, \quad (35)$$

for $k \in [2..N_d+1]$ and $q = k-1$.

The boundary state, Q_k^{BC} , is provided to the inviscid numerical flux of Eq. (26) that is modified by prescribing a null dissipation on the wall, i.e., $D = 0$.

- Viscous (no-slip condition): we first set $Q_k^{\text{BC}} = Q_k^-$ (hence $\hat{Q}_k = Q_k^-$) for $k \notin [2..N_d+1]$ and $\hat{Q}_k = 0$ ($Q_k^{\text{BC}} = -Q^-$) for $k \in [2..N_d+1]$ in the computation of full and partial derivative corrections in Eqs. (30) and (32). As for the viscous numerical flux of Eq. (31), we consider $\mathcal{H}_k^{\text{vis}} = F_{kj}^{\text{vis}}(Q_k^-, \mathbf{e}_q(\partial_q Q_k)^-) n_j$ for all k at the adiabatic wall boundary except for $k = N_d+2$ (conservation of energy) that instead receives $\mathcal{H}_{N_d+2}^{\text{vis}} = 0$.

III.E. Solution process

Starting from an initialization by exact values (in manufactured cases) or by freestream values (in the flat plate case), an implicit Euler scheme, i.e., the relaxed Newton's method with pseudo-time integration, is employed to smoothly converge the non-linear residuals to the vicinity of the final solution by gradually decreasing the relaxation (increasing the pseudo-time step) down to the final stage where the full Newton's method yields a quadratic convergence to the discrete solution. The linearization of the residual equation with regards to the solution (the Jacobian matrix) is required by the Newton's method to determine the direction of descent. In order to ensure optimal residual minimization (to machine precision), an analytical Jacobian is implemented for the fully coupled system of RANS-SA including the original and modified portions of the SA equation. This linearization is then verified to provide ≈ 15 significant digits when compared to a linearization by *complex step* via *operator overloading*.³⁹ The operator overloading automatically ensures that the Jacobian matrix is consistent with the residual equation such that any modification in the latter is inherently accounted for. The effort involved in the achievement of an exact Jacobian is justified since it enables the full operation of the Newton's method in the minimization of residuals. In fact, as iterative and round-off errors scale with residuals,¹ their minimization permits to safely direct the focus towards the discretization error. The linear system is solved by a generalized minimal residual (GMRES) method along with a global block-Jacobi and local ILU(0) preconditioners.

IV. Verification methodology and results

IV.A. Code verification

A trigonometric manufactured solution for each primitive variable is defined by the superposition of three sinusoidal functions in respectively x , y and both coordinates simultaneously. The latter cross terms are necessary for the verification of the viscous terms of the governing equations which feature mixed second order partial derivatives whereas the terms in single coordinates are included to maintain the spatial dependency in the case where the cross terms are turned off for debugging purposes by setting their coefficients to zero. The generic form of the MS is

$$\begin{aligned}\rho^{\text{MS}} &\equiv \rho_0 + \rho_x \sin(a_{\rho_x} \pi x/L) + \rho_y \cos(a_{\rho_y} \pi y/L) + \rho_{xy} \cos(a_{\rho_{xy}} \pi x/L) \cos(a_{\rho_{xy}} \pi y/L), \\ u^{\text{MS}} &\equiv u_0 + u_x \sin(a_{u_x} \pi x/L) + u_y \cos(a_{u_y} \pi y/L) + u_{xy} \cos(a_{u_{xy}} \pi x/L) \cos(a_{u_{xy}} \pi y/L), \\ v^{\text{MS}} &\equiv v_0 + v_x \cos(a_{v_x} \pi x/L) + v_y \sin(a_{v_y} \pi y/L) + v_{xy} \cos(a_{v_{xy}} \pi x/L) \cos(a_{v_{xy}} \pi y/L), \\ p^{\text{MS}} &\equiv p_0 + p_x \cos(a_{p_x} \pi x/L) + p_y \sin(a_{p_y} \pi y/L) + p_{xy} \cos(a_{p_{xy}} \pi x/L) \cos(a_{p_{xy}} \pi y/L), \\ \tilde{v}^{\text{MS}} &\equiv \tilde{v}_0 + \tilde{v}_x \cos(a_{\tilde{v}_x} \pi x/L) + \tilde{v}_y \cos(a_{\tilde{v}_y} \pi y/L) + \tilde{v}_{xy} \cos(a_{\tilde{v}_{xy}} \pi x/L) \cos(a_{\tilde{v}_{xy}} \pi y/L),\end{aligned}\tag{36}$$

where u and v respectively refer to the first and second velocity components, $L = 1.0$ is a reference length and the manufactured total energy, E^{MS} , is obtained via Eq. (12) with $\gamma = 1.4$.

We choose to define the MS in terms of the primitive variables since this procures a few advantages: firstly, since the Mach number features the primitive variables directly, tuning the primitive variables of the MS allows for an easier control over the flow regime (subsonic vs. supersonic, compressible vs. incompressible). Secondly, the derivatives appearing in the forcing functions are those of the primitive variables and the MS definition in terms of the latter spares futile chain-rule operations. Finally, the expression of MS in primitive variables is forthwith compatible with the non-conservative formulation of incompressible solvers.

The constants appearing in Eqs. (36) serve to tune the magnitudes and frequencies of the trigonometric functions of the manufactured primitive variables such that the manufactured solution verifies all the terms of the PDE in the desired flow regime and furthermore features a numerically desirable behavior. For example, a suitable manufactured solution would yield an asymptotic (monotonic) range that is attainable on reasonably fine grids and that is nevertheless not trivially obtained on the coarsest grids. Finally, let's recall that the manufactured solution is not bound by strict fidelity to physics since the verification is rather a mathematical exercise; it is however desirable to produce quantities that remain within the range of physically valid variations such that density and absolute temperature are defined positive for example.

The spatial domain for this set of manufactured solutions is $\Omega = [0, 1]^2$ and unless specified, all the grids are formed by isotropic quadrangular elements and refined by halving each element in each direction.

We target the solver verification of ensemble-averaged Navier-Stokes equations closed with a complete turbulence model, that are exemplified here by a RANS-SA system. The manufactured solution MS-1 serves

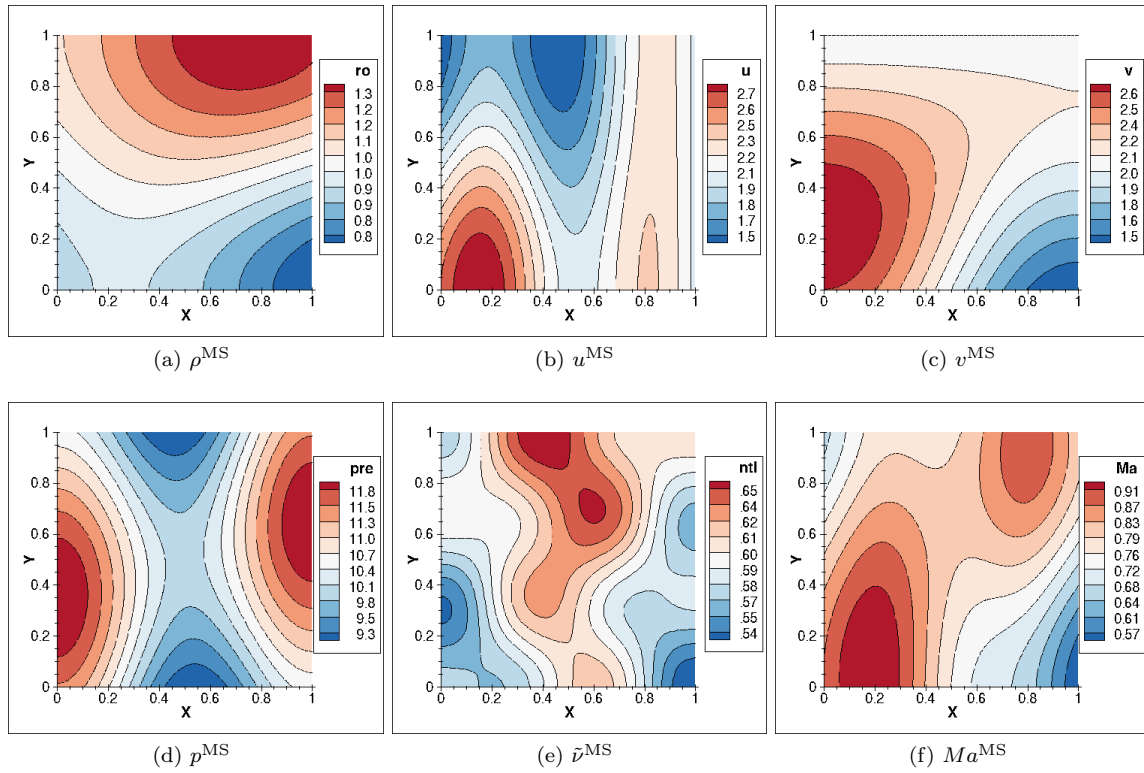


Figure 2: MS-1 fields

(\cdot)	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a(\cdot)_x$	$a(\cdot)_y$	$a(\cdot)_{xy}$
ρ	1.0	0.1	-0.2	0.1	1.0	1.0	1.0
u	2.0	0.3	0.3	0.3	3.0	1.0	1.0
v	2.0	0.3	0.3	0.3	1.0	1.0	1.0
p	10.0	1.0	1.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	0.6	-0.03	-0.02	0.02	2.0	1.0	3.0

Table 1: Parameters of the MS-1

this purpose and is determined by parameters presented in Table 1 along with Eqs. (36) and a value of dynamic viscosity of $\mu = 1 \times 10^{-3}$ chosen such that the eddy viscosity be preponderant in comparison. The MS-1 is devised such that it verifies the original portion of the SA model (Eqs. 11, 13a, 14a, 15a, 16a and 17a) for positive $\tilde{\nu}$ values in a first step. The distance to the wall is set to $d_w = y + 1$, for $y \in [0, 1]$, to avoid large values in SA source terms with dependencies on $\frac{1}{d_w^2}$, that could otherwise saturate the entire forcing function. The balancing of different SA terms is discussed at length further in this section. The MS-1 fields are presented in Fig. 2. The errors and orders of accuracy are respectively presented in Figs. 22 and 23 for L norms and in Figs. 24 and 25 for H_1 semi-norm of corrected and uncorrected gradients. For the same spatial distribution of the discretization error, each norm in Fig. 22 outputs a different value. A general trend can be identified, with the L_∞ yielding the highest value, followed by the L_2 and L_1 norms that are closer together. This is expected since the L_∞ reports the maximal value of the error throughout the domain. The L_1 norm measures the magnitude of the error, averaged over the domain, whereas the L_2 norm can be regarded as the magnitude of the error, weighed by itself first, and averaged then. Hence the L_2 norm is more sensitive to the error deviations from the average value. Consequently, the disparity between the L_1 and L_2 norms reflects the irregularity of the spatial distribution of the error.

The importance of the L_∞ norm in code verification has previously been discussed in the literature (see²⁶ and¹³) as to be a valuable metric for the detection of localized inconsistencies such as coding errors in the

implementation of the boundary conditions. But the discussion has mostly stemmed from theoretical and practical appreciations of the qualities of this norm. Here, we demonstrate the importance of the L_∞ norm by rather providing a concrete example: a bug affecting the implementation of the boundary condition at a single point is introduced by changing the boundary value of the ρu variable of MS-1 at $(x, y) = (0, 0)$ such that $(\rho u)^{\text{BC}} = 1.000001 \times (\rho^{\text{MS}} u^{\text{MS}})|_{\text{BC}}$, while keeping all other boundary conditions at all other points intact in Eq. (33). Fig. 3 shows the evolution of the resulting discretization errors and orders of accuracy versus mesh refinement for the polynomial degrees P1 to P5. For P1 and P2, none of the error norms has yet been affected by the punctual boundary error on the finest mesh. Indeed, the discretization error of the P1 is rather large and drops slowly with mesh refinement ($\mathcal{E} \simeq \mathcal{O}(h^2)$) when compared to higher orders. Hence, for the boundary inconsistency to be manifested in P1 and P2 polynomial discretizations, one needs to drive the mesh refinement process up to very fine grids for the level of discretization error to be lower than that caused locally by the boundary bug that is around $\mathcal{E}_{\rho u}(0, 0) = |\rho u(0, 0) - \rho u_{\text{ex}}(0, 0)| \approx 2 \times 10^{-7}$, see Fig. 3 (a). For P3 and P4 however, the abnormality is detected on the finest mesh but only by the L_∞ norm. As the degree of the polynomial discretization is increased, the punctual source of stagnating error dominates the norms for lower mesh sizes since the discretization error diminishes more rapidly on coarser grids for higher P. As such, the L_∞ norm of P5 is affected by the spurious boundary condition as off the third grid with a mesh size of $h \approx 2 \times 10^{-2}$ which is the coarsest grid to manifest the bug amongst all the polynomial degrees considered. But still, it takes two more levels of mesh refinement in P5 for the L_1 and L_2 norms to pick the error up as well. The effect of the mesh size on the discretization error is better illustrated in Fig. 4 that compares for P4, the higher levels of discretization error, distributed regularly on a 8×8 grid, to the lower levels of discretization error on a 32×32 grid where the manufactured bug at $(x, y) = (0, 0)$ is exhibited as the global maximum error on the domain. These observations support the idea that the L_∞ norm is a crucial metric to ensure the early detection of localized sources of error on coarse grids. Finally, it is interesting to note in Fig. 4 (b), that the error is being convected in the stream-wise direction from the punctual source.

One other essential conclusion can be drawn from Fig. 3 pertaining to the verification of high-order codes: it is desirable to include a fairly high polynomial order in the verification process to ensure the early detection of bugs causing errors with small magnitudes. Indeed, since the higher polynomials reduce the discretization error at a higher rate, the presence of the slightest bug can be revealed on coarser grids than if lower orders were used, thus providing a greater confidence in the absence of minor bugs in the code.

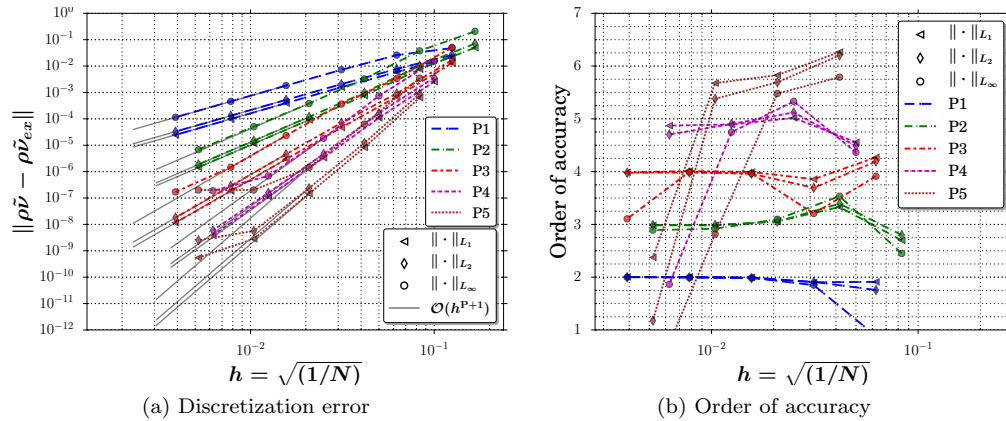


Figure 3: Evolution of the discretization errors and the orders of accuracy in L_1 , L_2 and L_∞ norms versus mesh refinement for $\rho\tilde{u}$, polynomial degrees P1–P5 and MS-1 with the spurious boundary condition of $(\rho u)^{\text{BC}} = 1.000001 \times (\rho^{\text{MS}} u^{\text{MS}})|_{\text{BC}}$ at $(x, y) = (0, 0)$

In realistic flow problems, the Reynolds number (Re) provides a measure of the relative magnitude of advective over the diffusive forces, that is also related to the ratio of the inviscid over the viscous terms of the NS equations. The magnitude of the viscous terms depends on both the value of the viscosity and the strain rate which involves the velocity gradients. The latter are intrinsically adjusted by flow features such as the boundary layer thickness, the freestream velocity, the characteristic length, etc. that are themselves correlated with Re (and consequently with μ). However, since a manufactured solution is not necessarily realistic, it might be absent of these features and the velocity gradients could take arbitrary

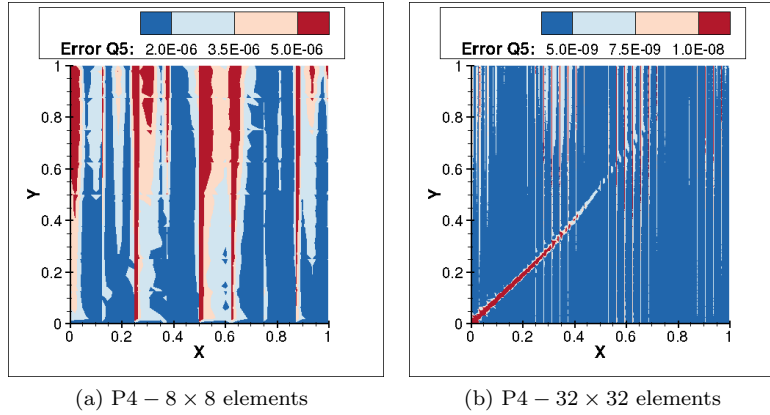


Figure 4: Effect of mesh refinement on the distribution of the $\rho\tilde{v}$ discretization error for the polynomial degree P4 and MS-1 with the spurious boundary condition of $(\rho u)^{BC} = 1.000001 \times (\rho^{MS} u^{MS})|_{BC}$ at $(x, y) = (0, 0)$

values independently of the value of viscosity. Hence, to generalize the concept of Reynolds number to manufactured cases, we directly look into $S^{rel}(\mathbf{x}) = S^{inv}(\mathbf{x})/S^{vis}(\mathbf{x})$, the ratio of the inviscid over the viscous terms of the manufactured forcing functions in the domain as in.¹¹ Figure 5 compares the S^{rel} distributions for the x -momentum equation of MS-1 and three different non-dimensional values of dynamic viscosity, i.e., $\mu = 1 \times 10^{-4}$, $\mu = 1 \times 10^{-1}$ and $\mu = 1 \times 10^{+2}$ (considering $\rho\tilde{v} = 0$ temporarily such that $\mu_{eff} = \mu$). Note that the MS fields (and hence the velocity gradients) are fixed while μ is changed resulting in different forcing functions. To ensure the verification of the inviscid and viscous terms both by the MS, the $S^{rel}(\mathbf{x})$ distribution should simultaneously include regions of viscous preponderance ($|S^{rel}(\mathbf{x})| \ll 1$) and inviscid prevalence ($1 \ll |S^{rel}(\mathbf{x})|$) along with non-negligible regions of equivalence ($|S^{rel}(\mathbf{x})| \approx 1$). This is an important criterion, since when one mode outweighs the other by much, the verification process becomes biased and the detection of minor bugs by the MMS in the outweighed mode is jeopardized, thus incurring the risk of a false *pass* verdict. Figure 5 suggests that for MS-1, a value of $\mu = 1 \times 10^{-1}$ guarantees a balanced leverage since for this value, non-negligible regions of inviscid and viscous dominance exist in different regions of the domain simultaneously.

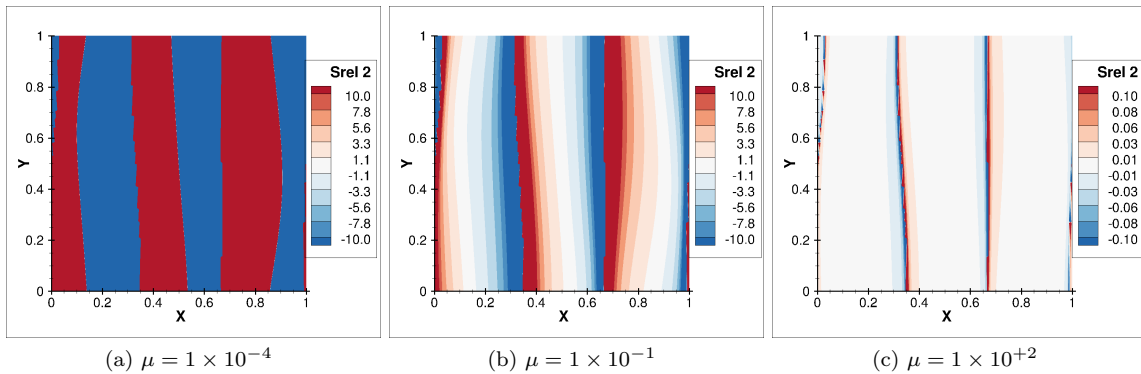


Figure 5: Relative magnitude of the inviscid over viscous terms of the forcing function of the x -momentum equation for MS-1 with different μ values

In the laminar case, achieving a balance between the advective and diffusive terms is a relatively easy task since the value of μ directly scales their ratio. For turbulent flows however, the presence of source terms within the RANS model, involving a range of quantities, makes this balancing a more challenging task. For the assessment of the sought balance, we propose considering a relative absolute magnitude, $S_{term}^{rel} = |S_{term}| / S_{SA}^{sum}$, as metric of the sensitivity of the verification process to each SA term where $|S_{term}|$

is the absolute magnitude of the SA term and

$$S_{SA}^{sum} \equiv \sum_{term} |S_{term}| = \underbrace{\left| \partial_j (\rho u_j \tilde{\nu}) \right|}_{\text{Advection}} + \underbrace{\left| -\partial_j \left(\frac{1}{\sigma} (\mu + \rho \tilde{\nu}) \partial_j \tilde{\nu} \right) \right|}_{\text{Diffusion}} + \underbrace{\left| \rho c_{b1} (1 - f_{t2}) \tilde{s} \tilde{\nu} \right|}_{\text{Production}} + \underbrace{\left| -\rho \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \frac{\tilde{\nu}^2}{d_w^2} \right|}_{\text{Destruction}} + \underbrace{\left| \frac{c_{b2}}{\sigma} \rho \partial_j \tilde{\nu} \partial_j \tilde{\nu} \right|}_{\text{Distribution}} + \underbrace{\left| \frac{1}{\sigma} (\nu + \tilde{\nu} f_n) \partial_j (\rho \partial_j \tilde{\nu}) \right|}_{\text{Conservation}}. \quad (37)$$

The 6 terms of the SA forcing function appearing in Eq. (37) are those of the original portion of the SA model, Eqs. 11, 13a, 14a, 15a, 16a and 17a. For each of these terms, the spatial distribution of the S_{term}^{rel} is provided in Fig. 6 showing that the advection, diffusion and destruction terms exhibit spatial maxima of $\max(S_{term}^{rel}) \approx 50\%$ on the domain whereas the sensitivity to production, distribution, and conservation terms only accounts for $\max(S_{term}^{rel}) \approx 5\%$. To establish whether such a low sensitivity is acceptable and how it relates to actual sensitivity to a given bug, we introduce an error in the distribution term of the SA model, $(1 + d\alpha) \frac{c_{b2}}{\sigma} \rho \partial_j \tilde{\nu} \partial_j \tilde{\nu}$, such that the original term ($d\alpha = 0$) is modified sequentially by increasing $d\alpha$ from $d\alpha = 1 \times 10^{-15}$ by an order of magnitude ($\times 10$) in each step and the orders of accuracy are recomputed until the bug is detected. Consequently, a significant degradation in the orders of the P5 discretization was detected for a value of $d\alpha = 1 \times 10^{-7}$ (see Fig. 7) demonstrating that considerably small inconsistencies in the implementation of the distribution term can be detected and hence $\max(S_{term}^{rel}) \approx 5\%$ is a sufficient criterion to ensure that all terms are correctly verified. From Fig. 7, it can be again emphasized that including a reasonably high (such as P5) polynomial order in the verification process is crucial for the detection of minor inconsistencies in the implementation.

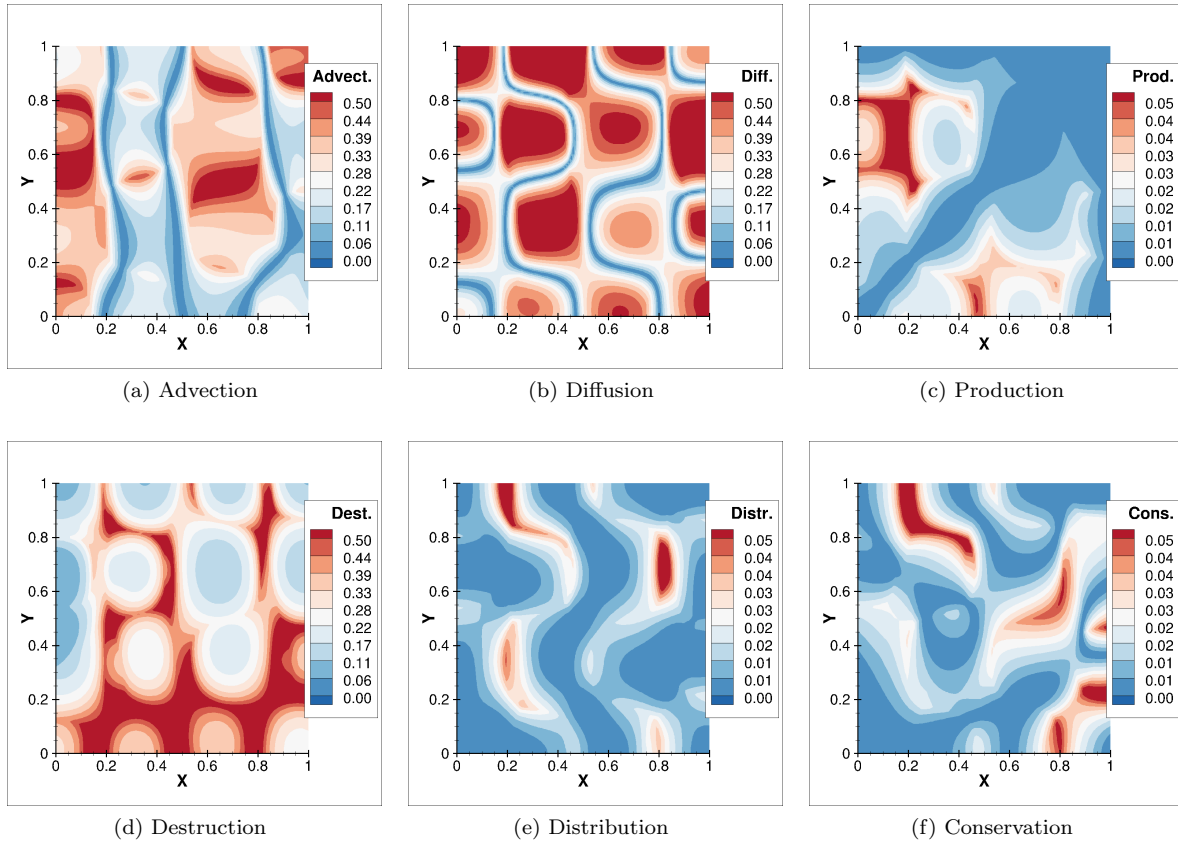


Figure 6: Sensitivity of the verification process to SA forcing function terms of MS-1, measured by S_{term}^{rel}

Based on the sensitivity analysis presented and the results obtained for orders of accuracy in Figs. 23 (L norms) and 25 (H_1 semi-norm), it is possible to conclude that the considered implementation of the RANS

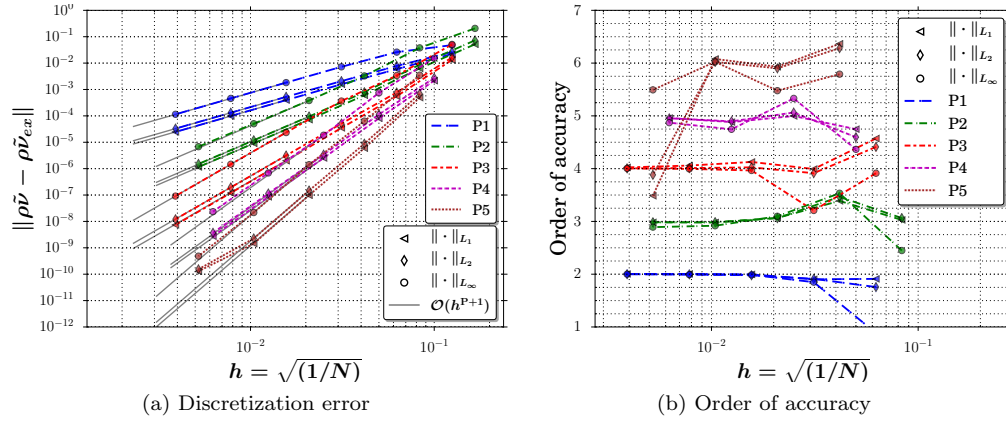


Figure 7: Evolution of the discretization errors and the orders of accuracy in L_1 , L_2 and L_∞ norms versus mesh refinement for $\rho\tilde{v}$, polynomial degrees P1–P5 and MS-1 with the spurious SA distribution term, $(1 + d\alpha)^{\frac{c_{b2}}{\sigma}} \rho \partial_j \tilde{v} \partial_j \tilde{v}$

equations along with the original portion of the SA model is verified via MS-1 and hence the focus can be moved on the modified portion of the SA model.

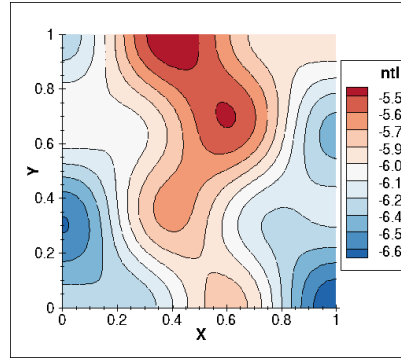


Figure 8: Manufactured solution MS-2 for \tilde{v}

We propose the MS-2 for the verification of the modified portion of the SA model, recalling that the revised model is intended to tackle the instabilities generated when negative values of the SA working variable occur in the numerical solution. The manufactured fields of the MS-2 are those of the MS-1 except for \tilde{v} . Table 2 presents the parameters that define the MS-2. The dynamic viscosity is set to $\mu = 1 \times 10^{-1}$. For negative \tilde{v} values, the RANS equations are decoupled from the SA equation by eliminating the effect of the latter on the former via $\mu_t = 0$. However, the SA equation remains unilaterally coupled to the RANS equations. Therefore, the MS-2 only verifies the implementation of the SA model and regarding the state variables of all other equations, it should result in the same exact error and orders as observed for the MS-1 in Figs. 22, 23 for L norms and in Figs. 24 and 25 for H_1 semi-norm. As for the SA equation, its manufactured field as well as errors and orders in L and H_1 norms are respectively presented in Figs. 8, 26 and 27.

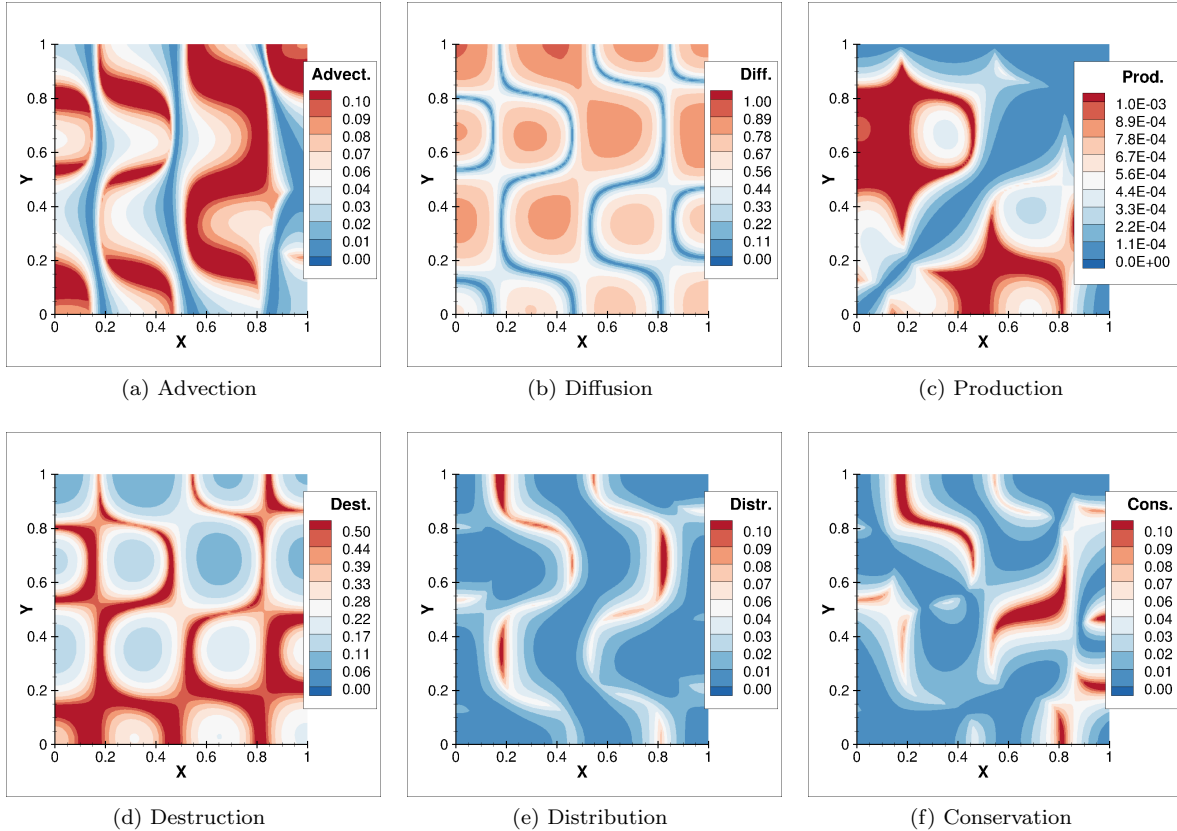
A similar budget analysis of the SA forcing function to the one for MS-1 is conducted here by rather

(\cdot)	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a(\cdot)_x$	$a(\cdot)_y$	$a(\cdot)_{xy}$
ρ	1.0	0.1	-0.2	0.1	1.0	1.0	1.0
u	2.0	0.3	0.3	0.3	3.0	1.0	1.0
v	2.0	0.3	0.3	0.3	1.0	1.0	1.0
p	10.0	1.0	1.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	-6.0	-0.3	-0.2	0.2	2.0	1.0	3.0

Table 2: Parameters of the MS-2

considering the modified portion of the SA model (Eqs. 11, 13b, 14b, 15b, 16b and 17b) such that:

$$\begin{aligned}
 S_{SA}^{\text{sum}} \equiv \sum_{\text{term}} |S_{\text{term}}| = & \underbrace{\left| \partial_j (\rho u_j \tilde{\nu}) \right|}_{\text{Advection}} + \underbrace{\left| -\partial_j \left(\frac{1}{\sigma} (\mu + \rho \tilde{\nu} \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}) \partial_j \tilde{\nu} \right) \right|}_{\text{Diffusion}} + \underbrace{\left| \rho c_{b1} (1 - c_{t3}) s \tilde{\nu} \right|}_{\text{Production}} + \\
 & \underbrace{\left| c_{w1} \frac{\tilde{\nu}^2}{d_w^2} \right|}_{\text{Destruction}} + \underbrace{\left| \frac{c_{b2}}{\sigma} \rho \partial_j \tilde{\nu} \partial_j \tilde{\nu} \right|}_{\text{Distribution}} + \underbrace{\left| \frac{1}{\sigma} (\nu + \tilde{\nu} \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}) \partial_j (\rho \partial_j \tilde{\nu}) \right|}_{\text{Conservation}}.
 \end{aligned} \tag{38}$$

Figure 9: Sensitivity of the verification process to SA forcing function terms of MS-2, measured by $S_{\text{term}}^{\text{rel}}$

This analysis resulted in the spatial sensitivity distributions of Fig. 9 showing that in descending order of magnitude, the SA forcing function depends on the diffusion, destruction, advection, conservation, distribution and production terms. The sensitivities to the latter term are very low, $\max(S_{\text{term}}^{\text{rel}}) \approx 0.1\%$, and are particularly under the criterion of $\max(S_{\text{term}}^{\text{rel}}) \approx 5\%$, established previously. It is hence prudent to conduct an actual sensitivity analysis by introducing an inconsistency in the implementation of the production term of the SA model as $(1 + d\alpha) (\rho c_{b1} (1 - c_{t3}) s \tilde{\nu})$ and computing the orders of accuracy for an initially small

and increasingly larger values of $d\alpha$ until a difference in the orders is observed. This exercise resulted in the detection of the bug as soon as $d\alpha = 1 \times 10^{-6}$. Therefore, the production term is verified by the MS-2 even for minor bugs and MS-2 can be considered for the verification of the modified SA equation.

The only term of the governing equations that remains unverified throughout the manufactured cases presented so far is the modified vorticity magnitude, \tilde{s} of Eq. 15b. In fact, this is the only term of the modified SA that is not activated by a condition on the sign of $\tilde{\nu}$ values but rather by a criterion on the vorticity magnitude, s , that is often satisfied in the vicinity of solid walls. The verification of this term is therefore better suited for wall-bounded manufactured cases.

IV.B. Solution verification

Since the code is successfully verified, it is reasonable to employ it to solve a simple but realistic engineering problem, that is, a 2D zero-pressure-gradient flat plate with $Re = 5 \times 10^6$, $Ma = 0.2$ and $\chi_\infty = 0.3$. This is in fact a benchmark case from NASA's TMR.⁴⁰ The domain, illustrated in Fig. 10, is discretized by a fine

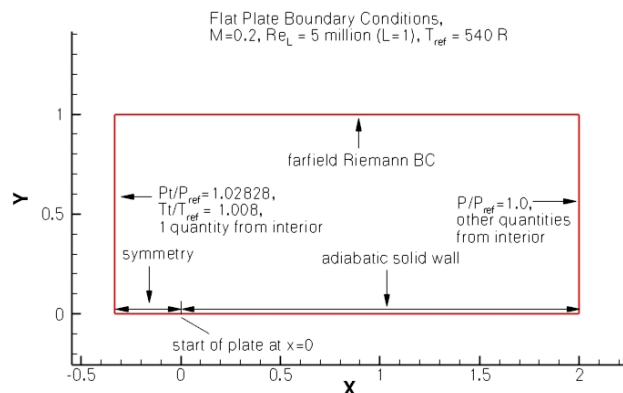


Figure 10: Domain and boundary conditions description for the flat plate case^a

grid of 544×384 quadrangle elements provided by TMR. We label this grid as L4. For the grid convergence study, the L4 grid is coarsened by removing every other grid line, resulting in a total of 5 levels of consistently coarsened grids: L4 to L0. Furthermore, on each of these grids, three levels of polynomial discretization, P1, P2 and P3, are applied to yield a total of 15 simulations. Figure 11 shows examples of residual convergence of the continuity and the SA equations. In each case, the initial residuals are reduced by at least 10 orders of magnitude.

Figure 12 compares the contour plots of μ_t/μ_∞ , the relative eddy viscosity, between the P3-L1 solution and the CFL3D reference solution from NASA's Turbulence Modeling Resource website that is obtained on the L4 grid of 209,825 quadrangle elements; the grids are displayed beneath the contour plots. The contours are comparable between the two solutions although the L1 grid is approximately 318 times coarser than the L4 grid used to obtain the CFL3D solution. One can however observe the oscillations at the edge of the boundary layer, for the P3 solution, that are due to the under-resolution of the sharp drop of $\rho\tilde{\nu}$ in this region from boundary layer to free-stream values. On even coarser discretizations, this under-resolution of the large $\rho\tilde{\nu}$ gradients at the edge of the boundary layer results in the occurrence of negative values of the SA working variable as shown in Fig. 13(a) for the P2 polynomial discretization. As the mesh is refined however, this unphysical undershoot is reduced since a better discretization of the boundary layer edge restrains the spurious numerical oscillations of $\tilde{\nu}$ values caused by the Gibbs phenomenon in this region, as can be seen in 13(b).

The relative velocity and the relative eddy viscosity profiles at $x = 0.097$ are plotted in Figs. 15–17 for respectively P1–P3 polynomial discretization on all grid refinement levels compared to the reference CFL3D solution on the L4 grid. A few comments can be made at this point: first, as the mesh is refined for a given P, the profiles converge to a final value; second, this convergence occurs faster for higher P on the same grid basis; third, the final profiles produced by the current solver, especially that of the relative velocity, show a small discrepancy with regards to the CFL3D profiles; finally, the Gibbs oscillations of the relative eddy viscosity profiles at the edge of the boundary layer, i.e., $0.015 \leq y \leq 0.025$, are particularly pronounced for

^aFigure adopted from the NASA's Turbulence Modeling Resource website

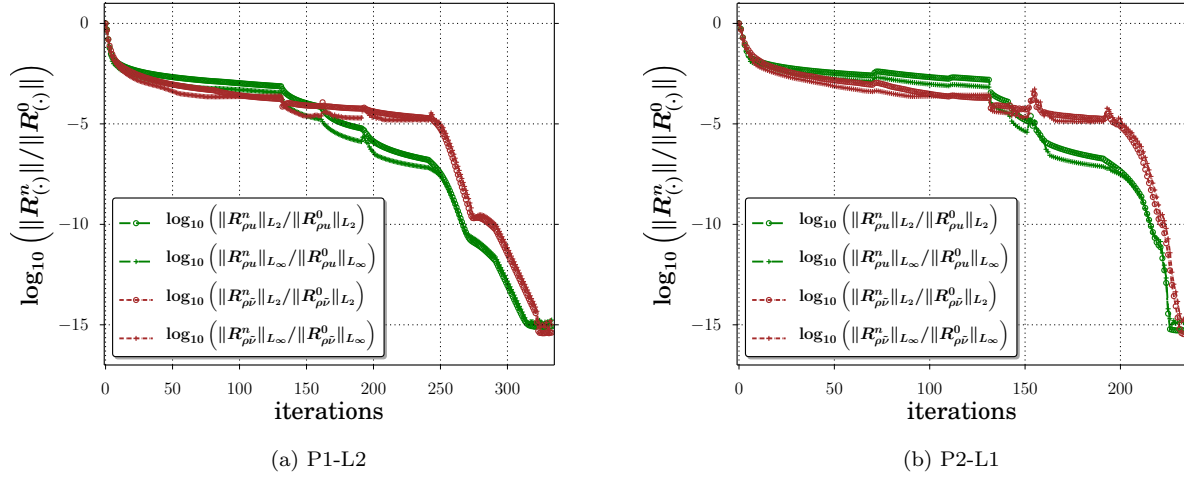


Figure 11: Convergence of the x momentum and SA relative residuals in L_2 and L_∞ norms in the flat plate case

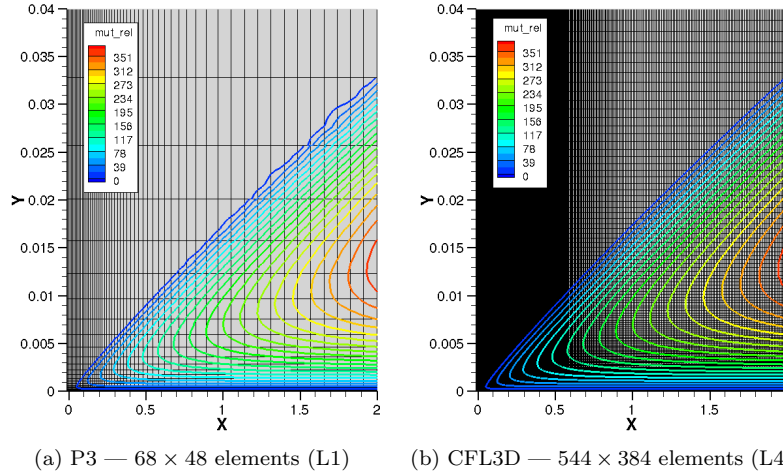


Figure 12: Comparative grid and relative eddy viscosity distributions in the boundary layer for high- and low-order discretizations of the flat plate case

coarser discretizations and reduce as the number of degrees of freedom increases by grid and/or polynomial refinements.

The distributions of

$$C_f \equiv \frac{\tau_{\text{wall}}}{\frac{1}{2}\rho_\infty U_\infty^2},$$

the friction coefficient along the plate, are illustrated in Figs. 18 (a)– 20 (a) for respectively P1–P3 discretizations on all grid levels compared to the reference CFL3D solution. As the grid density is increased, the solution converges to a final value along the plate and faster so for higher P, although a small gap of ≈ 0.2 drag counts remains between the L4 and the reference solutions at the trailing edge of the plate. This gap could be explained by the fact that the boundary values at the outlet of the domain are set to $Q_k^{\text{BC}} = Q_k^-$ via a Riemann BC and for all k except for $k = N_d + 2$, i.e., the energy equation where $Q_{N_d+2}^{\text{BC}} = \frac{p_{\text{out}}}{\gamma - 1} + \frac{1}{2}\rho^-(u_i^- u_i^-)$ along with $p_{\text{out}} = p_\infty$. The CFL3D code similarly imposes a "specified pressure ratio" condition at the domain outlet that might however be adjusted to match a Ma target. This potential difference could account to a very small discrepancy in the mass flow rate that, for the relatively small domain of this case, would impact the velocity profiles in the boundary layer. The discontinuous friction coefficient solutions undergo oscillations at the leading edge as shown in the same Figs. 18 (a)– 20 (a). The magnitude of the oscillations

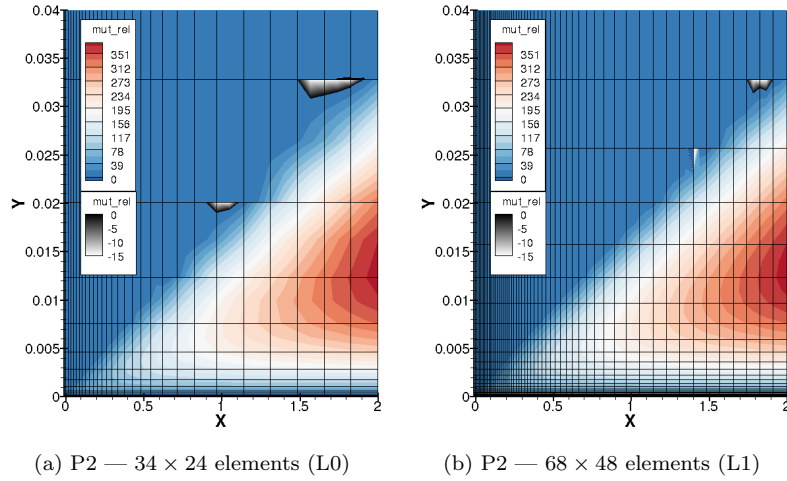


Figure 13: Occurrence of negative eddy viscosity values at the edge of the boundary layer and its reduction by mesh refinement for the flat plate case

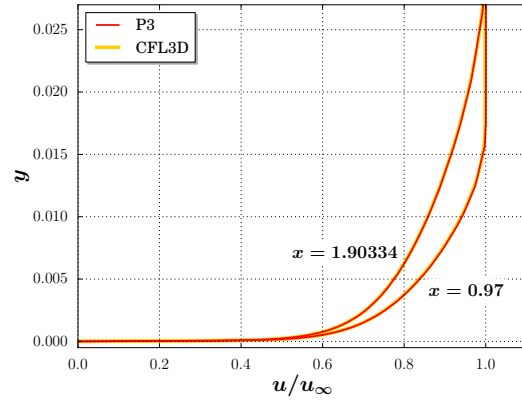


Figure 14: Dimensionless velocity profiles of the flat plate case

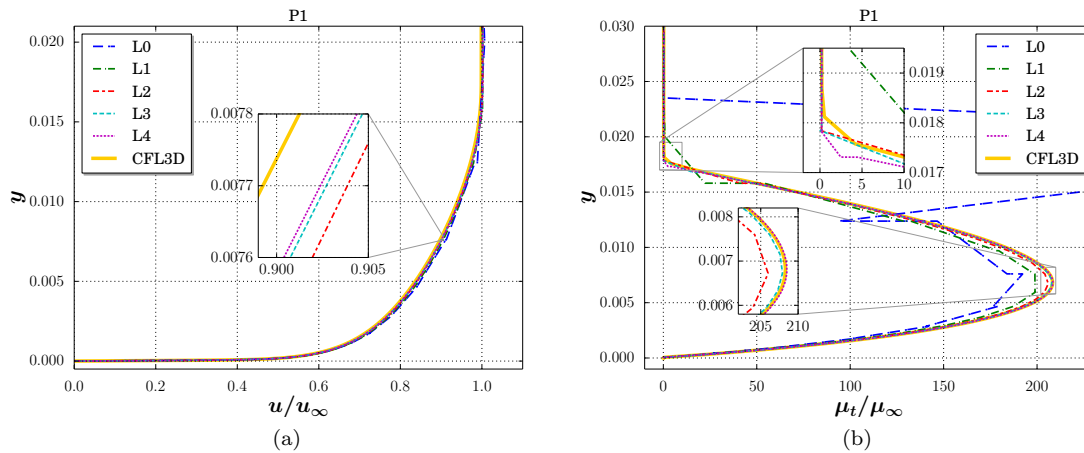


Figure 15: Relative velocity (a) and relative eddy viscosity (b) profiles at $x = 0.97$ for the reference solution of CFL3D and a CPR-P1 polynomial discretization of the flat plate case

increases as the mesh or P are refined since the scheme tends to approach the singular value at the leading

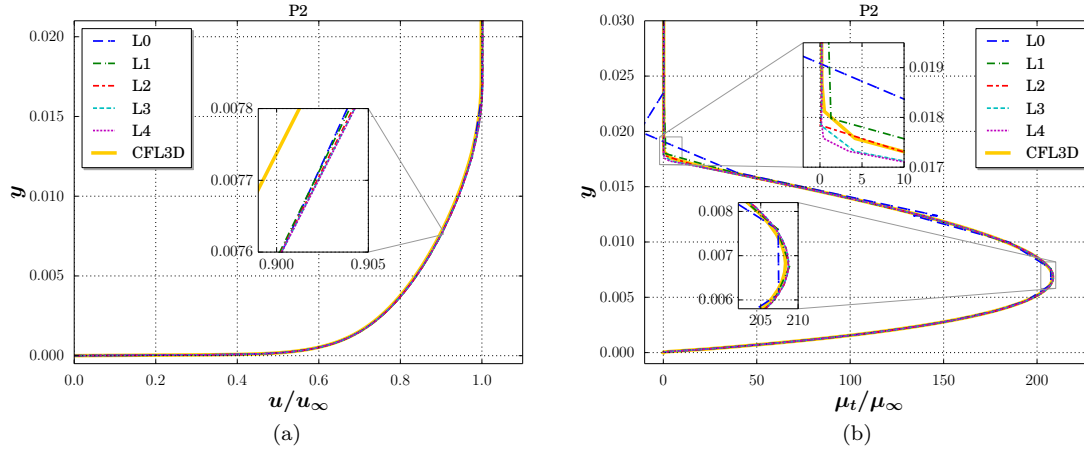


Figure 16: Relative velocity (a) and relative eddy viscosity (b) profiles at $x = 0.97$ for the reference solution of CFL3D and a CPR-P2 polynomial discretization of the flat plate case

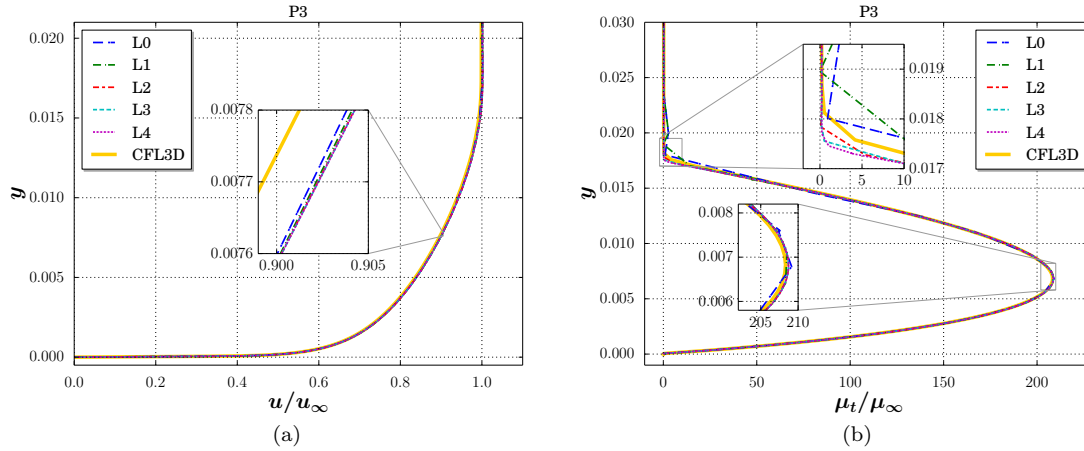


Figure 17: Relative velocity (a) and relative eddy viscosity (b) profiles at $x = 0.97$ for the reference solution of CFL3D and a CPR-P3 polynomial discretization of the flat plate case

edge more closely while accommodating as well the regular values along the plate, resulting in the propagation of sharp gradients emanating from the leading edge and contaminating a few elements downstream. However, the agreement of the CPR solutions to the CFL3D profile is improved by mesh refinement even close to the leading edge.

For respectively P1–P3, Figures 18 (b)– 20 (b) illustrate the grid convergence of profiles of

$$u^+ \equiv \frac{u}{\sqrt{\tau_{\text{wall}}/\rho_{\text{wall}}}} \quad \text{versus} \quad y^+ \equiv \frac{y\sqrt{\tau_{\text{wall}}/\rho_{\text{wall}}}}{\nu_{\text{wall}}}.$$

The y^+ values in these figures are those of the first degree of freedom off the wall corresponding the high-order solution point. Comparison to theoretical profiles in the same figures shows that under refinement, solutions approach the law of the wall, derived for the SA model by Allmaras et al.⁵ This provides an additional level of confidence in the validity of the grid-converged solution.

The grid convergence of the drag coefficient,

$$C_d \equiv \frac{\int_{\text{wall}} \mu \frac{\partial u}{\partial y} dx}{\int_{\text{wall}} dx},$$

for P1–P3 polynomial discretizations is compared to that of the CFL3D and FUN3D in Fig. 21 (a) showing that the CPR solutions tend to converge to a unique value for all polynomial degrees that is however slightly

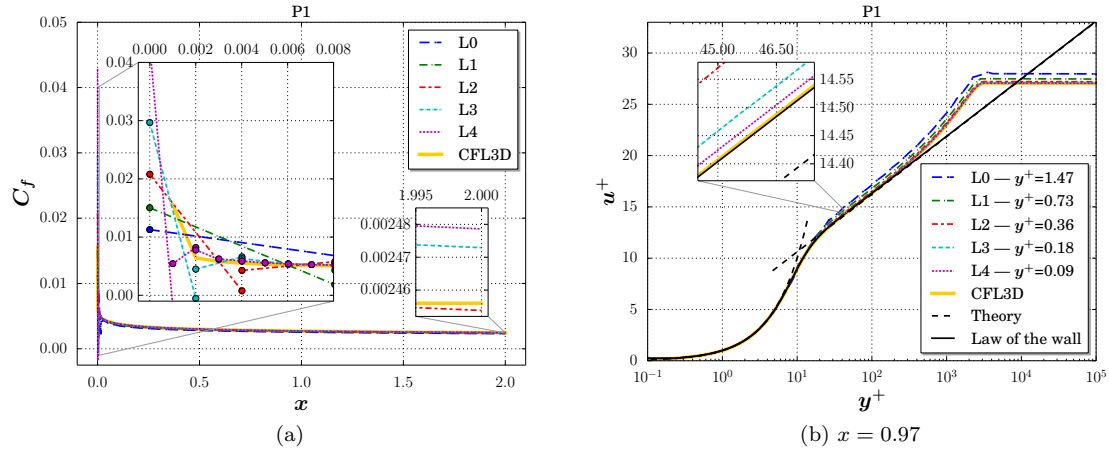


Figure 18: Friction coefficient along the plate (a) and normalized velocity profile at $x = 0.97$ (b) for the reference solution of CFL3D and a CPR-P1 polynomial discretization of the flat plate case; y^+ values in the legend refer to the normalized height of the first solution node off the wall

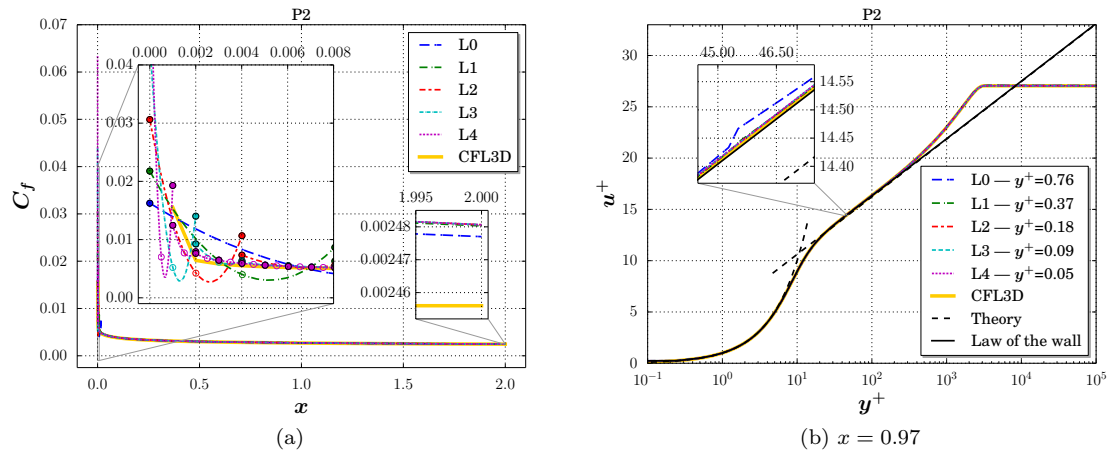


Figure 19: Friction coefficient along the plate (a) and normalized velocity profile at $x = 0.97$ (b) for the reference solution of CFL3D and a CPR-P2 polynomial discretization of the flat plate case; y^+ values in the legend refer to the normalized height of the first solution node off the wall

(≈ 0.2 drag count) different from the final value of the CFL3D/FUN3D. As mentioned previously, this could be attributed to differences in the outlet boundary condition between this code and the reference solvers. The same Fig. 21 depicts the evolution of the estimated C_d errors versus mesh refinement based on a Richardson extrapolation of the three finest P3 solutions. Two important observations can be made from this figure: firstly, the estimated orders of accuracy for the P1 and P3 seem to tend towards $\mathcal{O}(h^{P+1})$ whereas the P2 seems to rather approach $\mathcal{O}(h^P)$ on the last two grids; secondly, as expected, increasing the order of accuracy of the scheme results in a lower level of error for equal number of degrees of freedom (same h). The advantage of increasing the order is especially appreciable for a shift from P1 to P2 that produces a larger drop in the error than the P2 to P3 polynomial refinement, reflecting the fact that the low-order P1 discretization is particularly diffusive.

A detailed analysis of the apparent orders of accuracy for the drag coefficient is conducted here based on continuum solution estimates obtained by a Richardson extrapolation.¹ The latter error estimation method has the following underlying assumptions:⁶

1. The solution is smooth and exempt of geometrical and field discontinuities (no sharp corners, no shocks, etc.);
2. The formal order of accuracy of the discretization is known and the code has been verified to successfully

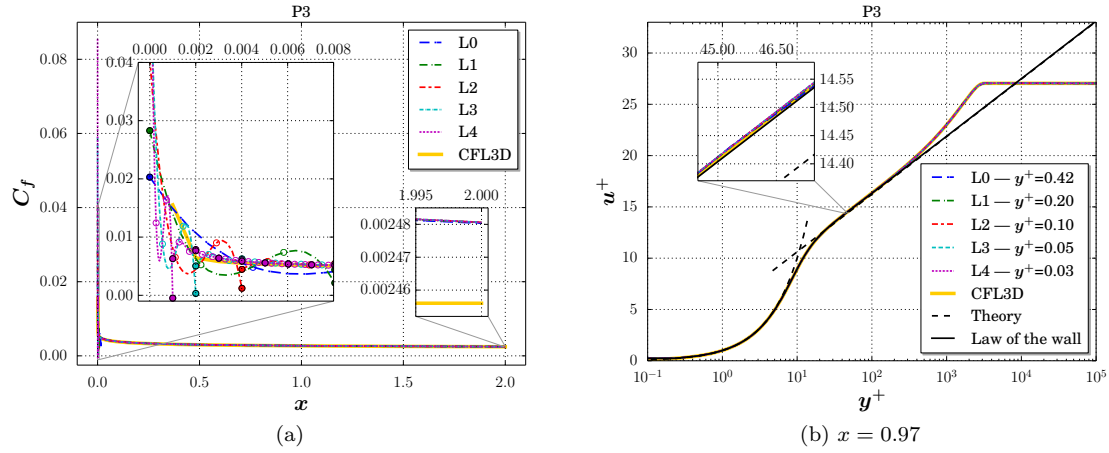


Figure 20: Friction coefficient along the plate (a) and normalized velocity profile at $x = 0.97$ (b) for the reference solution of CFL3D and a CPR-P3 polynomial discretization of the flat plate case ; y^+ values in the legend refer to the normalized height of the first solution node off the wall

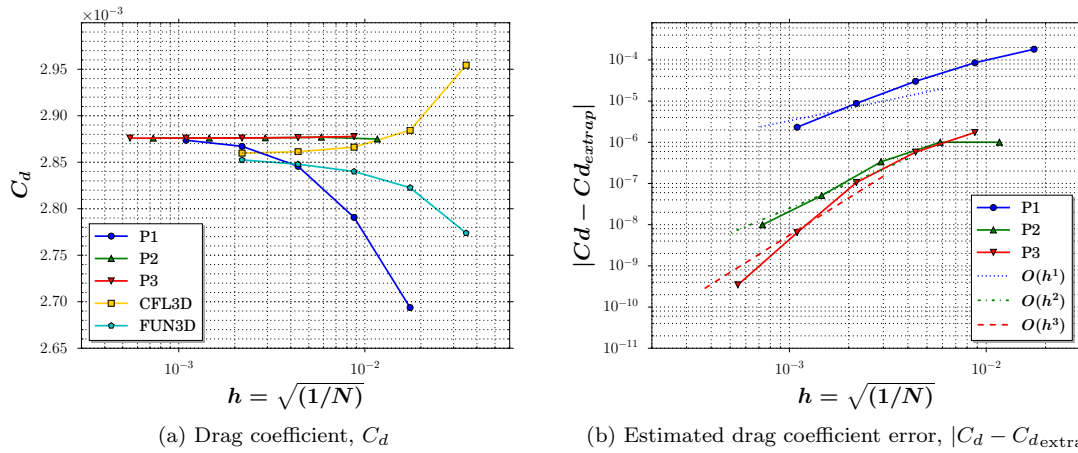


Figure 21: Grid convergence of the drag coefficient (a) and its estimated error (b) for P1 to P3 polynomial discretizations of the flat plate case

recover it;

3. The mesh size metric, h , is chosen such that it is a faithful representation of the typical element size and the grid set is generated such that h is consistently reducing from a coarser grid to the next finer one, resulting in a grid refinement ratio that is constant throughout the domain. In other words, the grid set is *self-similar*;
4. The finest solutions (for each polynomial degree) are in the asymptotic range, i.e. the rate of reduction of the logarithm of the error with regards to the logarithm of the mesh size is approaching a constant value as the mesh is refined and that value is relatively close to the expected order of accuracy.

The first requirement is obviously not satisfied here, in a strict sense, due to the presence of a geometrical discontinuity at the leading-edge of the plate that in turn induces a discontinuity in the discrete local residual equation. However, since the quantity of interest, C_d , is an integral output defined on a region of continuous solution (except at a single point), we assume that an appropriate tessellation of the domain would localize the impact of the singularity and hence the first requirement could be considered as fulfilled. The second assumption is already satisfied in the code verification step as the theoretical order of $O(h^{k+1})$ is recovered via the method of manufactured solutions for a P_k polynomial discretization. Note however that we expect an order of accuracy of $O(h^k)$ for C_d solutions in agreement with the orders of accuracy reported for high-order

DG solutions of a flat-plate case^{41–43} and high-order CPR solutions of an airfoil case.⁴⁴ The self-similarity of the grid set in the third point can be assumed satisfied since the set is generated by sequentially removing every other line from a fine grid. Regarding the mesh size metric, it is evaluated by $h = \sqrt[N_{\text{dof}}]{N_{\text{dof}}}$ where $N_{\text{dof}} \equiv \sum_{e_i} (N_{\text{node}})_{e_i}$ is the total number of degrees of freedom per equation that for a uniform Pk expansion throughout a tensor-product tessellation of N_e elements reduces to $N_{\text{dof}} = N_e N_{\text{node}} = N_e (k+1)^{N_d}$. We refer the reader to^{45,46} for further details on the effect of the mesh size metric on the observed orders of accuracy for self-similar and self-dissimilar grid sets. The fourth condition will be discussed at length in the following, and shown that although the convergence is not strictly monotonic, it is however sustained and sufficiently close to the theoretical order of accuracy for this assumption to be considered valid.

The Richardson extrapolation is based on the postulate that the discrete solution, $Q^{\text{Pk|Li}}$, of a grid refinement level Li (such as L0, L4, etc.) and a polynomial order Pk (such as P1, P3, etc.), can be represented by a power series expansion in the corresponding characteristic element size, $h^{\text{Pk|Li}}$, as

$$Q^{\text{Pk|Li}} = \theta_0 + \theta_1 h^{\text{Pk|Li}} + \theta_2 (h^{\text{Pk|Li}})^2 + \theta_3 (h^{\text{Pk|Li}})^3 + \dots + \theta_k (h^{\text{Pk|Li}})^k, \quad (39)$$

where θ_n are expansion coefficients. Let us assume that the exact solution can be approximated by

$$Q^{\text{ex}} \simeq Q^{\text{Pk|Li}} + \theta_{k+1} (h^{\text{Pk|Li}})^{k+1} + \theta_{k+2} (h^{\text{Pk|Li}})^{k+2} + \dots \quad (40)$$

Hence, the discretization error is expressed as

$$\mathcal{E}_Q^{\text{Pk|Li}} \equiv Q^{\text{ex}} - Q^{\text{Pk|Li}} \simeq \theta_{k+1} (h^{\text{Pk|Li}})^{k+1} + \mathcal{O}((h^{\text{Pk|Li}})^{k+2}), \quad (41)$$

and under the assumption of asymptotic convergence rate, i.e., when $h^{\text{Pk|Li}}$ is sufficiently small for the $\mathcal{O}((h^{\text{Pk|Li}})^{k+2})$ terms to be negligible, it is further reduced to

$$\mathcal{E}_Q^{\text{Pk|Li}} \simeq \theta_o (h^{\text{Pk|Li}})^o, \quad (42)$$

where o denotes the observed order of accuracy. Note that this effective rate of convergence of discrete to exact solutions of the quantity Q could be different from the theoretical order of the scheme ($k+1$) and/or from the recorded order of the solver in a code verification exercise. As mentioned previously, the discrepancies could arise from grid inadequacy, insufficient residual convergence, presence of singularities, etc. Let's recall that the purpose of the solution verification should hence not be to establish confidence in the soundness of the CFD solver by comparing the observed versus the theoretical orders of accuracy, but to rather provide assurance in the appositeness of the solution for use in a given engineering context.

A first estimate of the observed order of accuracy is computed via Eq. (42) and by considering $\mathcal{E}_Q^{\text{Pk|Li}}$ and $\mathcal{E}_Q^{\text{Pk|Li-1}}$. This constitutes two equations in the two unknowns θ_o and o that can easily be solved to yield $o_{\text{est1}}^{\text{Pk|Li}}$, a first estimation of the order of accuracy at $Pk|Li$, expressed as

$$o_{\text{est1}}^{\text{Pk|Li}} \equiv \frac{\log(\mathcal{R})}{\log(r)}, \quad (43)$$

with

$$\mathcal{R} \equiv \frac{C_d^{\text{Pk|Li-1}} - C_d^{\text{Pk|Li-2}}}{C_d^{\text{Pk|Li}} - C_d^{\text{Pk|Li-1}}}, \quad (44)$$

where $C_d^{\text{Pk|Li}}$ is the solution of drag coefficient for the polynomial order Pk and the grid level Li , and r is the constant mesh refinement ratio defined as $r = h^{\text{Pk|Li-1}}/h^{\text{Pk|Li}} = h^{\text{Pk|Li-2}}/h^{\text{Pk|Li-1}}$. This method hence needs three consecutive solutions in the grid refinement sequence. Note that the quotient \mathcal{R} in (43) is an indicator of the grid convergence with the following interpretation:

- $1 < \mathcal{R}$: sustained convergence,
- $0 < \mathcal{R} < 1$: sustained divergence,
- $-1 < \mathcal{R} < 0$: oscillatory divergence,

- $\mathcal{R} < -1$: oscillatory convergence.

Note as well that the $\log(\mathcal{R})$ in the nominator of Eq. (43) makes the application of this method only possible in the case of a sustained convergence (unless considering $\log(|\mathcal{R}|)$ in order to extend the method to the oscillatory convergence case by requiring $1 < |\mathcal{R}|$). Note as well that the order thus estimated is implicitly related to an estimation of the continuum solution for the polynomial order Pk . The $o_{\text{est1}}^{Pk|Li}$ values are presented in Table 3 for the flat plate case.

Another option would be to estimate the continuum solution based on drag solutions of a different (and preferably higher) polynomial order first and to use that exact solution estimate to then calculate the order of accuracy for the target polynomial order. We consider the highest polynomial order available (P3) and the finest grids (L4, L3 and L2) for the approximation of the continuum solution by

$$C_{d\text{extrap}}^{Pl|Lj} \equiv C_d^{Pl|Lj} + \frac{C_d^{Pl|Lj} - C_d^{Pl|Lj-1}}{r^{o_{\text{est1}}^{Pl|Lj}} - 1}, \quad (45)$$

for a polynomial degree Pl (P3 here) and a grid level Lj (L4 here) as well as a corresponding estimate of the order of accuracy, $o_{\text{est1}}^{Pl|Lj}$ via (43). For the flat plate case we obtain $C_{d\text{extrap}}^{P3|L4} = 0.0028759515$. This estimate is then used to compute the drag coefficient error, \mathcal{E}_{C_d} and consequently the order of accuracy for any Pk , respectively as

$$\mathcal{E}_{C_d}^{Pk|Li} \equiv |C_{d\text{extrap}}^{Pl|Lj} - C_d^{Pk|Li}|, \quad (46)$$

and

$$o_{\text{est2}}^{Pk|Li} \equiv \frac{\log\left(\frac{\mathcal{E}_{C_d}^{Pk|Li-1}}{\mathcal{E}_{C_d}^{Pk|Li}}\right)}{\log(r)}. \quad (47)$$

The results of $o_{\text{est2}}^{Pk|Li}$ are presented in Table 3. Note that $o_{\text{est1}}^{P3|L4} = o_{\text{est2}}^{P3|L4}$ as expected.

Once the continuum value is estimated using three Pl solutions, $o_{\text{est2}}^{Pk|Li}$ requires only two consecutive solutions in Pk to evaluate its observed order. Hence, in cases where only the last two Pk solutions are converging (oscillatory convergence) or where the order is not yet fixed (before the asymptotic range) in Pk but it is in Pl , we suggest to use $o_{\text{est2}}^{Pk|Li}$ instead of $o_{\text{est1}}^{Pk|Li}$, since by requiring three Pk solutions, the latter is more sensitive to deviations from monotonic grid convergence rate. It is however a good indicator of whether the asymptotic range is reached or not.

The error levels are compared in Table 3 to final residuals in L_2 norm, $\|R\|_{L2}$, of continuity and x momentum equations, showing that the residual convergence level is at least comparable and often lower by many orders of magnitude with regards to the estimated error in the drag coefficient.

With estimations of observed orders of accuracy being available, it is now possible to answer to the ultimate question of the solution verification process: "how reliable is the numerical solution?" To this end, we adopt the error indicator introduced by Roache,² that is the grid convergence index (GCI) computed as

$$\text{GCI}^{Pk|Li} \equiv \frac{\text{FS}}{r^{o_{\text{est2}}^{Pk|Li}} - 1} \left| \frac{C_d^{Pk|Li} - C_d^{Pk|Li-1}}{C_d^{Pk|Li}} \right|, \quad (48)$$

where FS is a safety factor adopted such that the GCI provides a level of 95% confidence that the exact solution is within the interval $Q^{Pk|Li} (1 - \text{GCI}) \leq Q^{\text{ex}} \leq Q^{Pk|Li} (1 + \text{GCI})$. Although the choice of the FS value to guarantee the 95% interval is an open research topic,¹ recommended values are FS = 1.25 for well-behaved cases, i.e., those featuring a sustained convergence and observed orders of accuracy that are close to the expected values on three consecutive grids and a value of FS = 3 for other conditions. As mentioned previously, the $o_{\text{est1}}^{Pk|Li}$ estimates are based on three consecutive Pk solutions; they therefore have a larger sensitivity to variations of the convergence rate compared to $o_{\text{est2}}^{Pk|Li}$ values based on two Pk solutions. We hence adopt the latter here and consequently consider FS = 3. The reader is referred to²⁵ for a detailed discussion of the extension of the GCI to irregular cases.

The $\text{GCI}^{Pk|Li}$ is equivalent to an error band on the $Q^{Pk|Li}$ solution based on the estimation of the discretization error via Richardson extrapolation¹ with the difference that instead of utilizing the exact solution estimate, Q_{extrap} , to normalize the approximation of the discretization error, the $Q^{Pk|Li}$ solution is employed in normalizing the GCI. This difference becomes obvious by comparing the $\text{GCI}^{Pk|Li}$ expression,

C_d results								
Pk	Li	$h^{Pk Li}$	$C_d^{Pk Li}$	$o_{est1}^{Pk Li}$	$o_{est2}^{Pk Li}$	$\mathcal{E}_{C_d}^{Pk Li}$	$\ R_{pu}\ _{L2}$	$\ R_p\ _{L2}$
1	L0	1.7504×10^{-2}	2.6937079×10^{-3}	—	—	$2. \times 10^{-4}$	$2. \times 10^{-13}$	$1. \times 10^{-11}$
1	L1	8.7518×10^{-3}	2.7906937×10^{-3}	—	1.10	$9. \times 10^{-5}$	$3. \times 10^{-13}$	$3. \times 10^{-11}$
1	L2	4.3759×10^{-3}	2.8456102×10^{-3}	0.82	1.49	$3. \times 10^{-5}$	$3. \times 10^{-13}$	$6. \times 10^{-11}$
1	L3	2.1879×10^{-3}	2.8671411×10^{-3}	1.35	1.78	$9. \times 10^{-6}$	$3. \times 10^{-13}$	$1. \times 10^{-10}$
1	L4	1.0940×10^{-3}	2.8736286×10^{-3}	1.73	1.92	$2. \times 10^{-6}$	$5. \times 10^{-13}$	$3. \times 10^{-10}$
2	L0	1.1669×10^{-2}	2.8749480×10^{-3}	—	—	$1. \times 10^{-6}$	$4. \times 10^{-13}$	$2. \times 10^{-11}$
2	L1	5.8345×10^{-3}	2.8769560×10^{-3}	—	0.00	$1. \times 10^{-6}$	$4. \times 10^{-13}$	$4. \times 10^{-11}$
2	L2	2.9173×10^{-3}	2.8762868×10^{-3}	1.59	1.58	$3. \times 10^{-7}$	$5. \times 10^{-13}$	$9. \times 10^{-11}$
2	L3	1.4586×10^{-3}	2.8760024×10^{-3}	1.23	2.72	$5. \times 10^{-8}$	$7. \times 10^{-13}$	$2. \times 10^{-10}$
2	L4	7.2931×10^{-4}	2.8759415×10^{-3}	2.22	2.36	$1. \times 10^{-8}$	$2. \times 10^{-12}$	$4. \times 10^{-10}$
3	L0	8.7518×10^{-3}	2.8776866×10^{-3}	—	—	$2. \times 10^{-6}$	$7. \times 10^{-13}$	$3. \times 10^{-11}$
3	L1	4.3759×10^{-3}	2.8765288×10^{-3}	—	1.59	$6. \times 10^{-7}$	$6. \times 10^{-13}$	$6. \times 10^{-11}$
3	L2	2.1879×10^{-3}	2.8760569×10^{-3}	1.30	2.45	$1. \times 10^{-7}$	$1. \times 10^{-12}$	$1. \times 10^{-10}$
3	L3	1.0940×10^{-3}	2.8759451×10^{-3}	2.08	4.04	$6. \times 10^{-9}$	$1. \times 10^{-12}$	$2. \times 10^{-10}$
3	L4	5.4698×10^{-4}	2.8759511×10^{-3}	4.21	4.21	$3. \times 10^{-10}$	$3. \times 10^{-12}$	$5. \times 10^{-10}$

Table 3: Drag coefficient, estimated orders of accuracy, estimated errors and residual convergence of the flat plate case

Error indicators					
Pk	Li	$GCI^{Pk Li} \%$	$\mathcal{E}_r^{Pk Li} \%$	$C_d^{Pk Li} (1 - \mathcal{E}_r^{Pk Li})$	$C_d^{Pk Li} (1 + \mathcal{E}_r^{Pk Li})$
1	L0	—	—	—	—
1	L1	9.2×10^0	8.9×10^0	2.5×10^{-3}	3.0×10^{-3}
1	L2	3.2×10^0	3.2×10^0	2.7×10^{-3}	2.9×10^{-3}
1	L3	9.2×10^{-1}	9.2×10^{-1}	2.84×10^{-3}	2.89×10^{-3}
1	L4	2.4×10^{-1}	2.4×10^{-1}	2.87×10^{-3}	2.88×10^{-3}
2	L0	—	—	—	—
2	L1	—	—	—	—
2	L2	3.5×10^{-2}	3.5×10^{-2}	2.875×10^{-3}	2.877×10^{-3}
2	L3	5.3×10^{-3}	5.3×10^{-3}	2.8758×10^{-3}	2.8761×10^{-3}
2	L4	1.5×10^{-3}	1.5×10^{-3}	2.87589×10^{-3}	2.87598×10^{-3}
3	L0	—	—	—	—
3	L1	6.0×10^{-2}	6.0×10^{-2}	2.874×10^{-3}	2.878×10^{-3}
3	L2	1.1×10^{-2}	1.1×10^{-2}	2.8757×10^{-3}	2.8763×10^{-3}
3	L3	7.5×10^{-4}	7.5×10^{-4}	2.87592×10^{-3}	2.87596×10^{-3}
3	L4	3.6×10^{-5}	3.6×10^{-5}	2.875950×10^{-3}	2.875952×10^{-3}

Table 4: Grid convergence indices, relative errors and error bands on C_d for the flat plate case

Eq. (48), to the relative discretization error, $\mathcal{E}_r^{\text{Pk|Li}}$, obtained by dividing Eq. (46) by Q_{extrap} and multiplying it by FS to yield

$$\mathcal{E}_r^{\text{Pk|Li}} \equiv \frac{\text{FS}}{r^{\mathcal{O}_{\text{est2}}^{\text{Pk|Li}} - 1}} \left| \frac{C_d^{\text{Pk|Li}} - C_d^{\text{Pk|Li}-1}}{C_{d\text{extrap}}} \right|, \quad (49)$$

where $C_{d\text{extrap}}$ is calculated from Eq. (45).

For the flat plate case, the $\text{GCI}^{\text{Pk|Li}}$ and $\mathcal{E}_r^{\text{Pk|Li}}$ values in Table 4 are very close except for the coarsest discretizations. The values are often very small and decrease significantly as the polynomial order is increased. For example, the P1-L4 and the P3-L3 discretizations, have both $h = 1.0940 \times 10^{-3}$ (see Table 3) and yet $\mathcal{E}_r^{\text{P3|L3}}$ is by four orders of magnitude lower than $\mathcal{E}_r^{\text{P1|L4}}$. The lower and upper bounds of confidence intervals are also presented in Table 4, demonstrating that higher orders of accuracy achieve narrower confidence intervals on the same grids. Note nevertheless that these bounds are strictly related to the numerical uncertainty and hence do not assume anything with regards to the physical values. In other words, the modeling errors are not considered in the solution verification process which rather investigates the influence of iterative, round-off and discretization errors on the solution.

In the light of the results obtained through solution verification, it can be concluded that although the observed orders of accuracy for C_d differ from the expected orders, $\mathcal{O}(h^k)$, and although the observed convergence is not perfectly monotonic, very satisfying confidence intervals are achieved on the finest solutions thanks to the fact that the grid set is capable of producing a sustained convergence with a rate close to the expected value.

V. Conclusions

We started by presenting the theoretical background in verification and validation for scientific computing with a focus on high-order CFD code verification via the method of manufactured solutions. Examples of application of the methodology are provided for the code verification of a high-order CPR-RANS-SA solver and the solution verification of a benchmark test case. Salient aspects of each verification exercise are emphasized and detailed, leading to the following conclusions:

- Code verification is a crucial and yet often neglected step prior to the application of a given code. Its objective is to verify that the CFD solver is capable of producing the expected orders of accuracy on benign problems. The code verification is especially imperative for high-order discretizations since, by demonstrating the ability of the code to deliver the expected higher accuracy per degree of freedom, it allows to confirm that the effort required by the development of the high-order solver and the achievement of a high-order solution is justified. Furthermore, proper code verification is crucial in scientific software quality assurance since it allows to continuously maintain the code's performance in terms of accuracy as it undergoes further developments.
- Attention needs to be paid to the balancing of inviscid versus viscous terms of the RANS equations and to that of different terms of the SA equation since otherwise, some terms could go unexamined through the verification process. We presented manufactured solutions that demonstrably produce a sufficient balancing for the RANS equations as well as for both the original and modified SA models.
- Whenever there is insufficient evidence that a given term is verified, we suggest to conduct a sensitivity analysis by introducing a minor alteration to that term and assessing its impact on the orders of accuracy. This will reinforce the significance of a *pass* verdict at the conclusion of the code verification phase. In this regard, let's note that code verification is rather a cumulative process of gathering evidence that the code functions as expected and can be trusted.
- The importance of the L_∞ norm in the early detection of inconsistencies that feature a local impact on the accuracy is demonstrated via a concrete example. Hence, the inclusion of this error metric in code verification is desirable.
- We recommend to include a relatively high-order discretization in the code verification process. It is indeed shown here that higher orders allow to detect minor bugs on coarser grids thanks to their higher accuracy and ensuing sensitivity to inconsistencies. However, if the order is too high, the discretization

error metrics might drop to machine precision level and get contaminated by the round-off errors before the asymptotic range is reached, thus preventing conclusions on the verification.

- The application of a code for solving practical problems should be strictly conditional to its prior successful verification for the equations and ideally all the boundary conditions involved in the problem.
- The methodology for solution verification in a high-order context is described in detail, and it is recommended to conduct a solution verification by computing GCIs prior to using the solution in an engineering context, especially for lower orders due to their larger confidence intervals (larger error bars). As for higher orders, a sustained convergence with a convergence rate close to the expected value seems sufficient to achieve very narrow confidence intervals. Nevertheless, this heavily depends on the adequacy of the grid set for the problem at hand.
- It is shown that the expected orders of accuracy can not always be achieved through solution verification even for a relatively simple turbulent flow such as that of the flat plate problem along with an "expert level" set of grids. Hence, solution verification should not be expected to replace proper code verification.
- Code comparison is also a rather defective approach to high-order code verification since as shown, discrepancies could exist in the truth values obtained by different codes due to differences in the boundary conditions for example. For a low-order solver, one could perhaps be satisfied by an order of magnitude comparison in truth values. For high-order solvers however, the verification of the order of accuracy is crucial as mentioned previously and this could not be properly conducted if the truth value is wrongly assumed portable from one code to another.

Acknowledgements

The authors gratefully acknowledge the generous support from the Fonds de recherche du Québec – Nature et technologies (FRQNT), the Natural Sciences and Engineering Research Council (NSERC), and the Department of Mechanical Engineering of McGill University. The first author would also like to thank Mr. Chih-Hao Chen for his insightful input on preconditioning for GMRES.

A. MMS results

A.A. MS-1

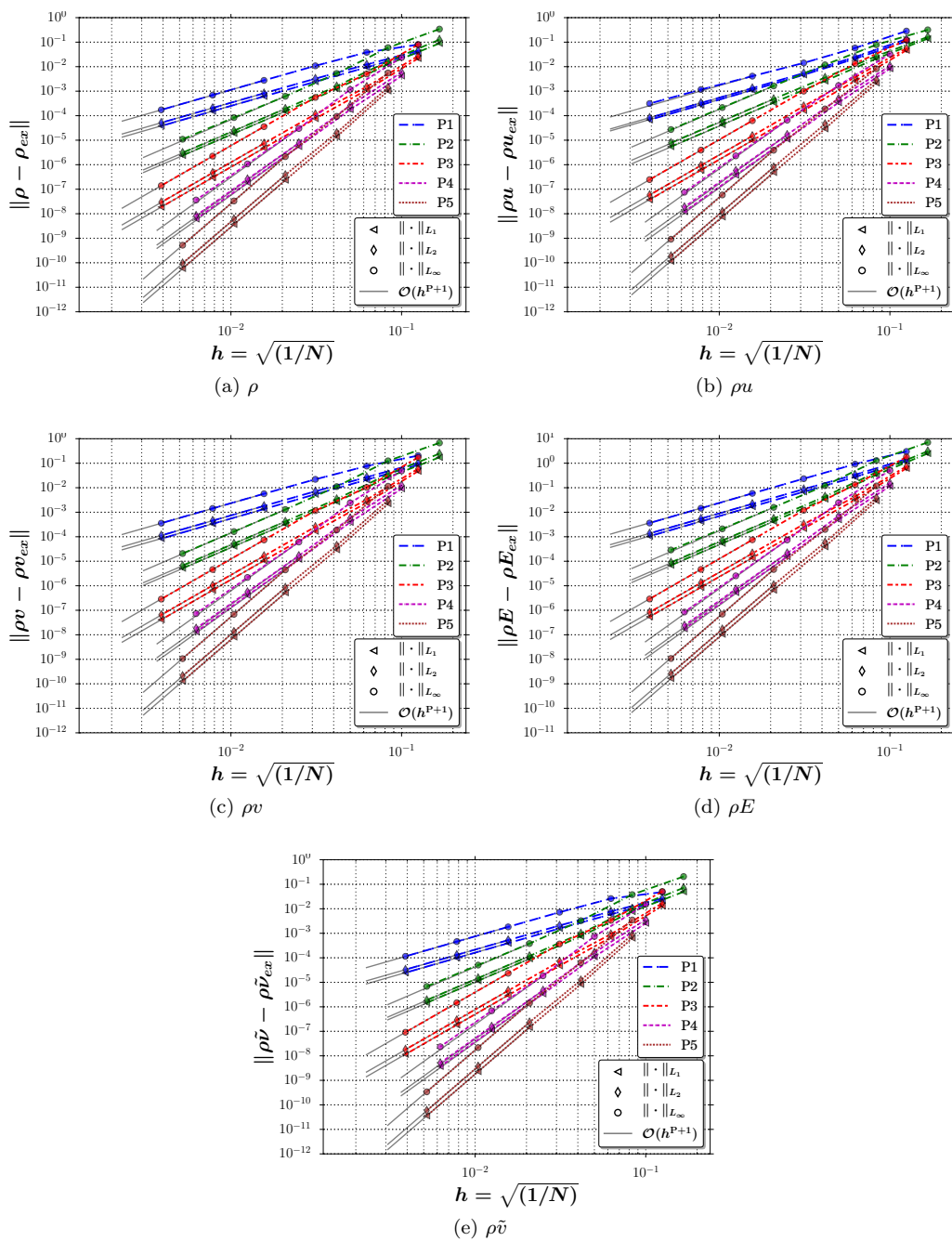


Figure 22: Evolution of the discretization error in L_1 , L_2 and L_∞ norms versus mesh refinement for MS-1 and polynomial degrees P1–P5

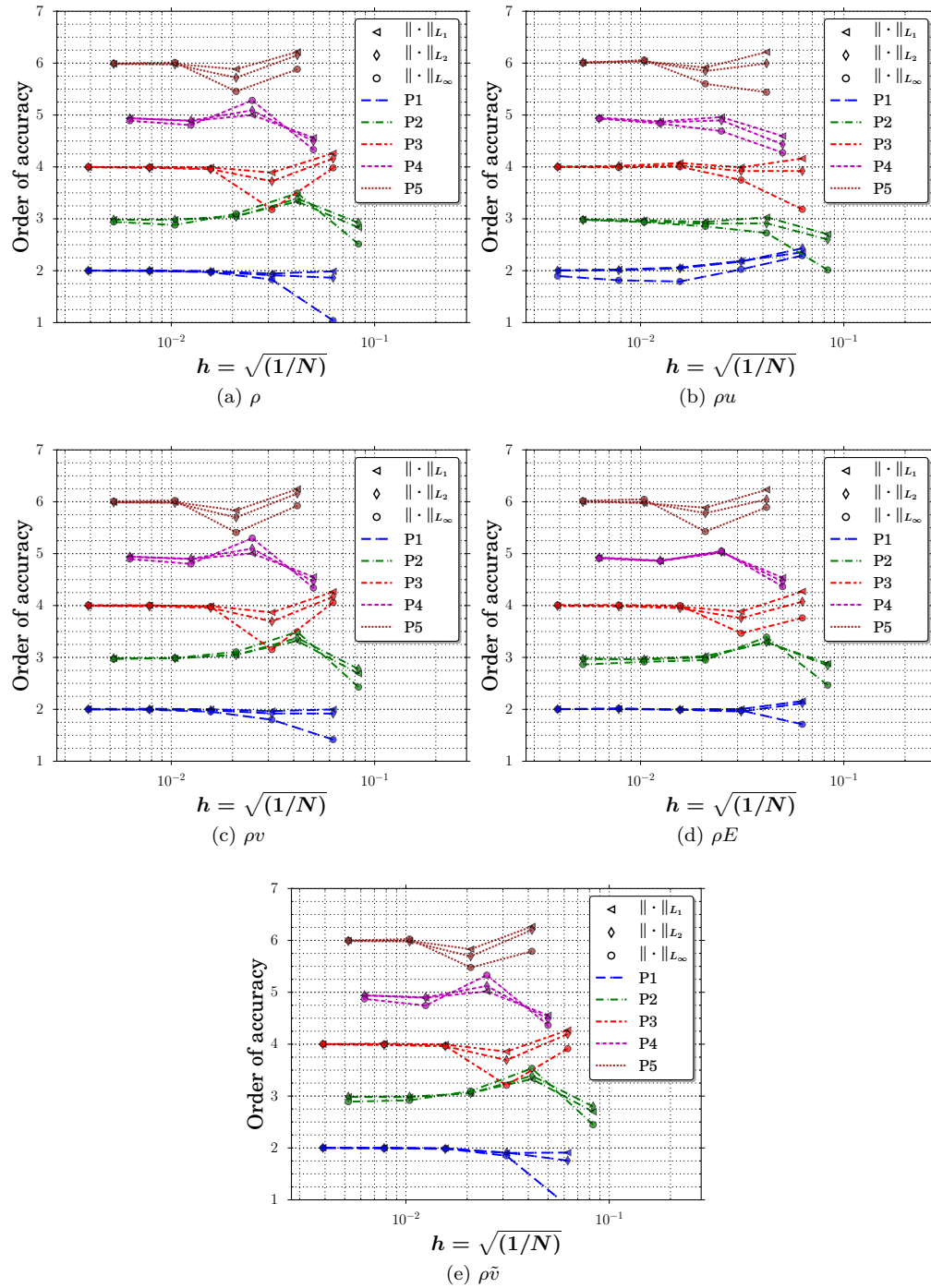


Figure 23: Evolution of the order of accuracy in L_1 , L_2 and L_∞ norms versus mesh refinement for MS-1 and polynomial degrees P1-P5

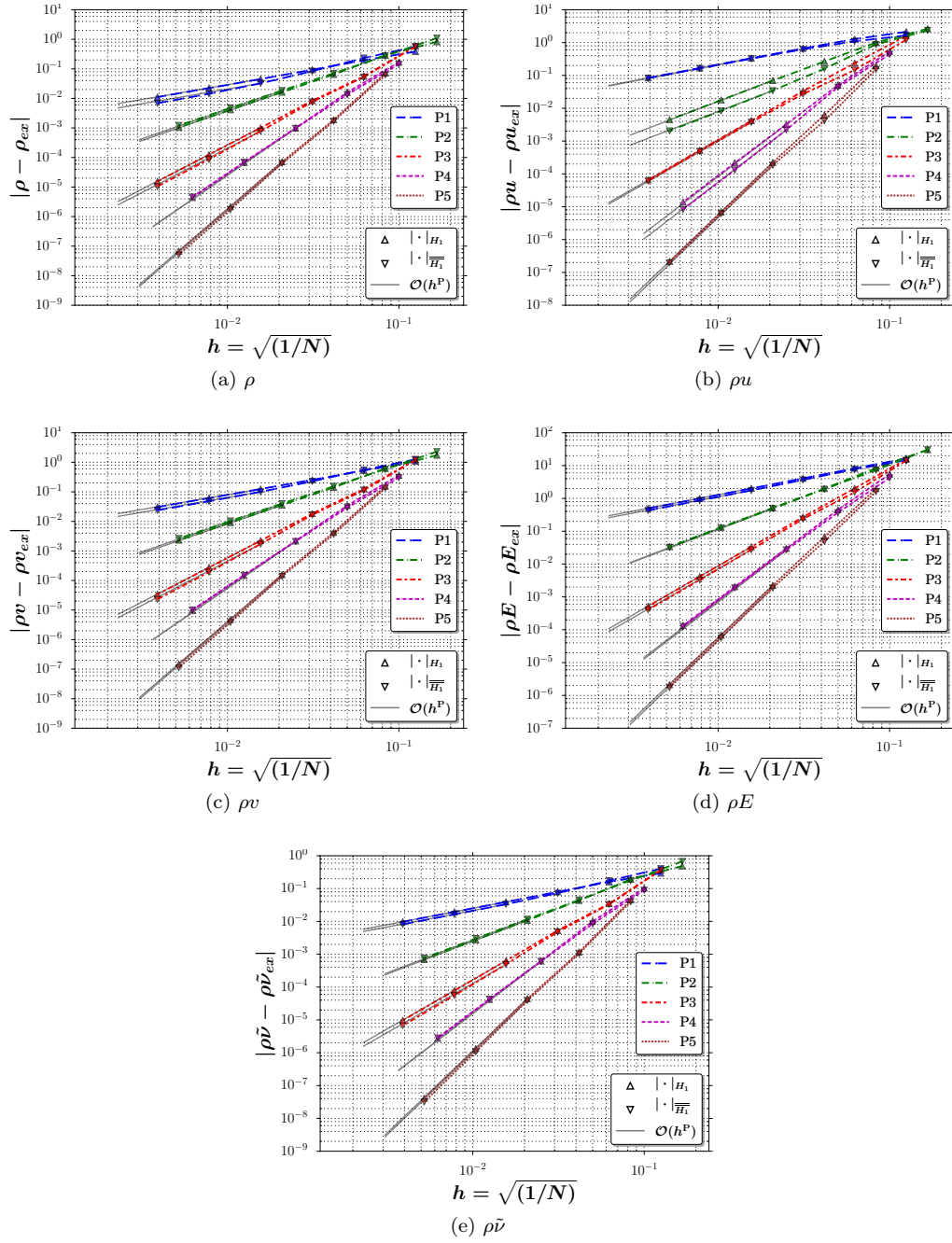


Figure 24: Evolution of the discretization error in H_1 semi-norm (for uncorrected and fully corrected derivatives) versus mesh refinement for MS-1 and polynomial degrees P1-P5

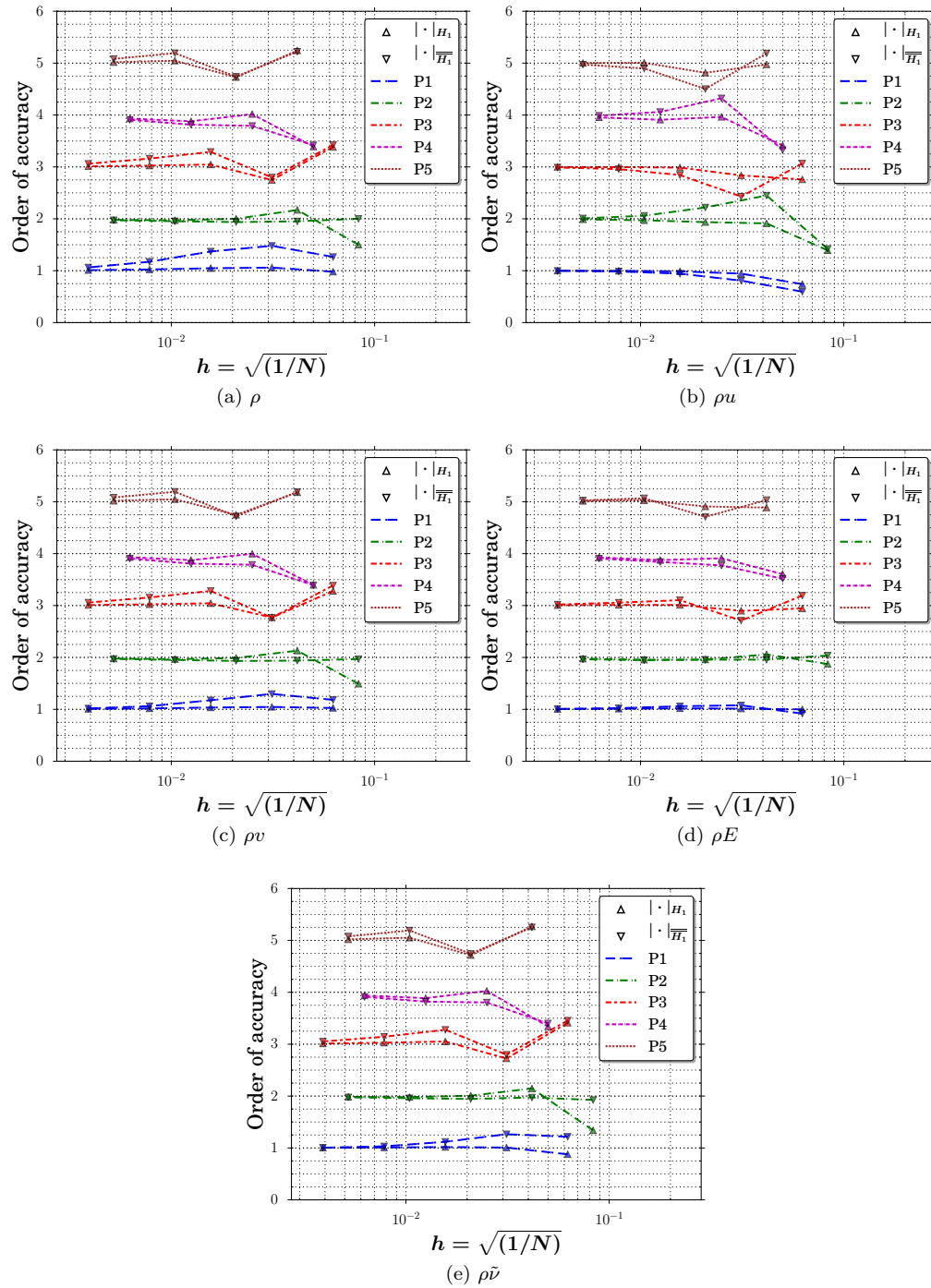


Figure 25: Evolution of the order of accuracy in H_1 semi-norm (for uncorrected and fully corrected derivatives) versus mesh refinement for MS-1 and polynomial degrees P1-P5

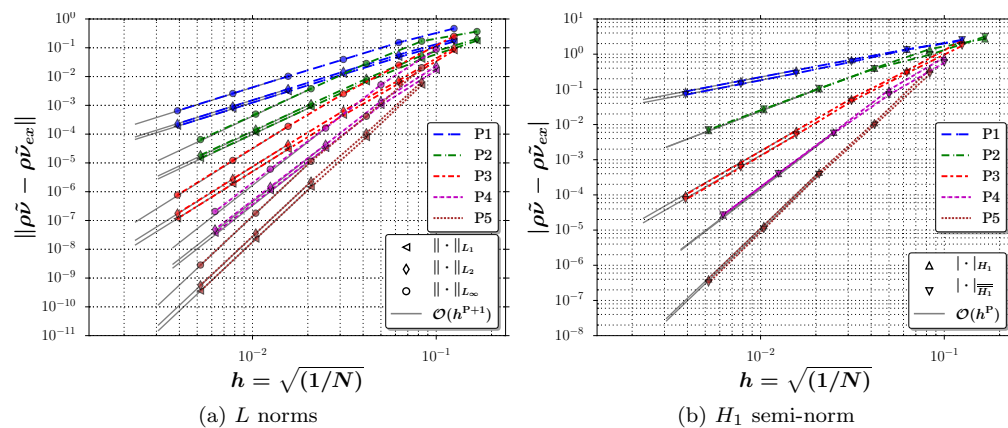


Figure 26: Evolution of the discretization error of $\rho\tilde{v}$ versus mesh refinement for MS-2 and polynomial degrees P1–P5

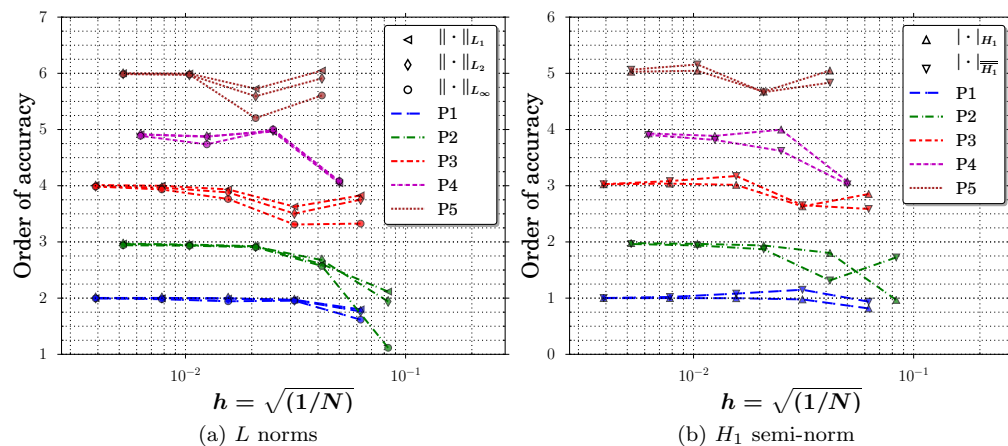


Figure 27: Evolution of the order of accuracy of $\rho\tilde{v}$ versus mesh refinement for MS-2 and polynomial degrees P1–P5

B. Error norms

Here are the definitions of the norms employed throughout this work to measure the discretization error. The integrals are computed by GLL quadratures.

- L_∞ norm:

$$\|\mathcal{E}_Q\|_\infty \equiv \max_{i: \text{dof}_i \in [\text{dof}_1 \dots \text{dof}_{N_{\text{dof}}}] } |Q_i - Q_i^{\text{ex}}| \quad (50)$$

- L_1 norm:

$$\|\mathcal{E}_Q\|_{L_1} \equiv \frac{\int_\Omega |Q - Q^{\text{ex}}| d\Omega}{\int_\Omega d\Omega} \quad (51)$$

- L_2 norm:

$$\|\mathcal{E}_Q\|_{L_2} \equiv \left(\frac{\int_\Omega (Q - Q^{\text{ex}})^2 d\Omega}{\int_\Omega d\Omega} \right)^{\frac{1}{2}} \quad (52)$$

- H_1 norm:

$$\|\mathcal{E}_Q\|_{H_1} \equiv \left(\frac{\int_\Omega (Q - Q^{\text{ex}})^2 d\Omega + \int_\Omega \sum_{q=1}^{N_d} (\partial_q Q - (\partial_q Q)^{\text{ex}})^2 d\Omega}{\int_\Omega d\Omega} \right)^{\frac{1}{2}} \quad (53)$$

- H_1 semi-norm of uncorrected gradients:

$$|\mathcal{E}_Q|_{H_1} \equiv \left(\frac{\int_\Omega \sum_{q=1}^{N_d} (\partial_q Q - (\partial_q Q)^{\text{ex}})^2 d\Omega}{\int_\Omega d\Omega} \right)^{\frac{1}{2}} \quad (54)$$

- H_1 semi-norm of fully corrected gradients:

$$|\mathcal{E}_Q|_{\overline{H_1}} \equiv \left(\frac{\int_\Omega \sum_{q=1}^{N_d} (\overline{\partial_q Q} - (\partial_q Q)^{\text{ex}})^2 d\Omega}{\int_\Omega d\Omega} \right)^{\frac{1}{2}} \quad (55)$$

References

- ¹Roy, C. J., "Review of code and solution verification procedures for computational simulation," *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.
- ²Roache, P. J., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, No. 3, sep 1994, pp. 405–413.
- ³Wang, Z. J. and Gao, H., "A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids," *Journal of Computational Physics*, Vol. 228, No. 21, 2009, pp. 8161–8186.
- ⁴Huynh, H. T., "A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion," *47th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, AIAA Paper 2009-403, Orlando, FL, jan 2009.
- ⁵Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," *7th International Conference on Computational Fluid Dynamics*, 2012.
- ⁶Oberkampf, W. L. and Trucano, T. G., *Verification and validation in computational fluid dynamics*, Vol. 38, 2002.
- ⁷Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.
- ⁸Ralf Hartmann, "Numerical Analysis of Higher Order Discontinuous Galerkin Finite Element Methods," *VKI LS 2008-08: CFD - ADIGMA course on very high order discretization methods*, Oct. 13-17, 2008, 2008.
- ⁹Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa publishers, Albuquerque, NM, 1998.
- ¹⁰Salari, K. and Knupp, P., "Code Verification by the Method of Manufactured Solutions," Tech. rep., Sandia National Laboratories, 2000.
- ¹¹Roy, C. J., Smith, T. M., and Ober, C. C., "Verification of a Compressible CFD Code Using the Method of Manufactured Solutions," *32nd AIAA Fluid Dynamics Conference and Exhibit*, 2002, pp. 1–14.
- ¹²Roy, C. J., Nelson, C. C., Smith, T. M., and Ober, C. C., "Verification of Euler/Navier-Stokes codes using the method of manufactured solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, No. 6, 2004, pp. 599–620.
- ¹³Bond, R. B., Ober, C. C., Knupp, P. M., and Bova, S. W., "Manufactured Solution for Computational Fluid Dynamics Boundary Condition Verification," *AIAA Journal*, Vol. 45, No. 9, 2007, pp. 2224–2236.
- ¹⁴Shunn, L. and Ham, F., "Method of Manufactured Solutions Applied to variable density flow solvers," *Center for Turbulence Research, Annual Research Briefs*, 2007, pp. 155–167.
- ¹⁵Vedovoto, J. M., da Silveira Neto, A., Mura, A., and Figueira da Silva, L. F., "Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme," *Computers and Fluids*, Vol. 51, No. 1, 2011, pp. 85–99.
- ¹⁶Ulerich, R., Estacio-Hiroms, K. C., Malaya, N., and Moser, R. D., "A Transient Manufactured Solution for the Compressible NavierStokes Equations With a Power Lawviscosity," *10th World Congress on Computational Mechanics*, jun 2012, pp. 1–16.
- ¹⁷Roy, C. J., Tendean, E., Veluri, S. P., Rifki, R., Luke, E. a., and Hebert, S., "Verification of RANS Turbulence Models in Loci-CHEM using the Method of Manufactured Solutions," *AIAA Computational Fluid Dynamics Conference*, Vol. 18, No. June, 2007, pp. 16.
- ¹⁸Veluri, S. P., Roy, C. J., and Luke, E. A., "Comprehensive code verification techniques for finite volume CFD codes," *Computers and Fluids*, Vol. 70, 2012, pp. 59–72.
- ¹⁹Oliver, T., Estacio-Hiroms, K., Malaya, N., and Carey, G., "Manufactured Solutions for the Favre-Averaged Navier-Stokes Equations with Eddy-Viscosity Turbulence Models," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, No. January, American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2012, pp. 1–19.
- ²⁰Tremblay, D., Etienne, S., and Pelletier, D., "Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems," *36th AIAA Fluid Dynamics Conference and Exhibit*, No. June, American Institute of Aeronautics and Astronautics, Reston, Virginia, jun 2006, pp. 1–11.
- ²¹Choudhary, A., Roy, C. J., Dietiker, J. F., Shahnam, M., Garg, R., and Musser, J., "Code verification for multiphase flows using the method of manufactured solutions," *International Journal of Multiphase Flow*, Vol. 80, 2016, pp. 150–163.
- ²²Eça, L., Hoekstra, M., Hay, A., and Pelletier, D., "A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow," *International Journal of Computational Fluid Dynamics*, Vol. 21, No. 3-4, mar 2007, pp. 175–188.
- ²³Eça, L., Hoekstra, M., Hay, a., and Pelletier, D., "On the construction of manufactured solutions for one and two-equation eddy-viscosity models," *International Journal for Numerical Methods in Fluids*, Vol. 54, No. November 2006, 2007, pp. 119–154.
- ²⁴Eça, L., Hoekstra, M., and Vaz, G., "Manufactured solutions for steady-flow Reynolds-averaged NavierStokes solvers," *International Journal of Computational Fluid Dynamics*, Vol. 26, No. February 2015, 2012, pp. 313–332.
- ²⁵Eça, L. and Hoekstra, M., "A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies," *Journal of Computational Physics*, Vol. 262, 2014, pp. 104–130.
- ²⁶Eça, L., Klaij, C., Vaz, G., Hoekstra, M., and Pereira, F., "On code verification of RANS solvers," *Journal of Computational Physics*, Vol. 310, 2016, pp. 418–439.
- ²⁷Malaya, N., Estacio-Hiroms, K. C., Stogner, R. H., Schulz, K. W., Bauman, P. T., and Carey, G. F., "MASA: A library for verification using manufactured and analytical solutions," *Engineering with Computers*, Vol. 29, 2013, pp. 487–496.
- ²⁸Fidkowski, K. J., "High-Order Output-Based Adaptive Methods for Steady and Unsteady Aerodynamics," *In H. Deconinck and R. Abgrall, editors, 37th Advanced CFD Lectures series*, Von Karman Institute for Fluid Dynamics, 2013, pp. 1–108.

- ²⁹Manuel, R. L., Bull, J., Crabill, J., Romero, J., Sheshadri, A., Ii, J. E. W., Williams, D., Palacios, F., and Jameson, A., "Verification and Validation of HiFiLES : a High-Order LES unstructured solver on multi-GPU platforms," *AIAA Aviation*, , No. June, 2014, pp. 1–27.
- ³⁰Galbraith, M. C., Allmaras, S., and Darmofal, D. L., "A Verification Driven Process for Rapid Development of CFD Software," 53rd *AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2015.
- ³¹Navah, F. and Nadarajah, S., "On the verification of high-order CFD solvers," *European Congress on Computational Methods in Applied Sciences and Engineering*, Hersonissos, Crete, Greece, 2016.
- ³²Silva, H. G., Souza, L. F., and Medeiros, M. A. F., "Verification of a mixed high-order accurate DNS code for laminar turbulent transition by the method of manufactured solutions," *International Journal for Numerical Methods in Fluids*, Vol. 64, No. 3, sep 2010, pp. 336–354.
- ³³Gao, H. and Wang, Z. J., "A conservative correction procedure via reconstruction formulation with the Chain-Rule divergence evaluation," *Journal of Computational Physics*, Vol. 232, No. 1, 2013, pp. pp. 7–13.
- ³⁴Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. pp. 357–372.
- ³⁵Burgess, N. K., *An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows*, Ph.D. thesis, University of Wyoming, 2011.
- ³⁶Hesthaven, J. S. and Warburton, T., *Nodal Discontinuous Galerkin Methods*, Vol. 54 of *Texts in Applied Mathematics*, Springer, 2008.
- ³⁷Bassi, F. and Rebay, S., "A high order discontinuous Galerkin method for compressible turbulent flows," *Discontinuous Galerkin Methods*, Springer, 2000, pp. 77–88.
- ³⁸Mengaldo, G., De Grazia, D., Peiro, J., Farrington, A., Witherden, F., and Sherwin, S. J., "A Guide to the Implementation of Boundary Conditions in Compact High-Order Methods for Compressible Aerodynamics," 7th *AIAA Theoretical Fluid Mechanics Conference AIAA*, Atlanta, GA, 2014.
- ³⁹Martins, J., Kroo, I., and Alonso, J., "An automated method for sensitivity analysis using complex variables," 38th *Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2000.
- ⁴⁰"Langley research center, turbulence modeling resource, <https://turbmodels.larc.nasa.gov/>," 2016.
- ⁴¹Nguyen, N., Persson, P.-O., and Peraire, J., "RANS Solutions Using High Order Discontinuous Galerkin Methods," 45th *AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2007.
- ⁴²Bassi, F., Ghidoni, A., Perbellini, A., Rebay, S., Crivellini, A., Franchina, N., and Savini, M., "A high-order Discontinuous Galerkin solver for the incompressible RANS and $k-\omega$ turbulence model equations," *Computers and Fluids*, Vol. 98, 2014, pp. 54–68.
- ⁴³Crivellini, A., D'Alessandro, V., and Bassi, F., "A SpalartAllmaras turbulence model implementation in a discontinuous Galerkin solver for incompressible flows," *Journal of Computational Physics*, Vol. 241, may 2013, pp. 388–415.
- ⁴⁴Shi, L. and Wang, Z. J., "Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method," *Journal of Computational Physics*, Vol. 295, 2015, pp. 261–284.
- ⁴⁵Eça, L., Hoekstra, M., Beja Pedro, J. F., and Falcão de Campos, J. A. C., "On the characterization of grid density in grid refinement studies for discretization error estimation," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 1, may 2013, pp. 119–134.
- ⁴⁶Salas, M. D., "Some observations on grid convergence," *Computers and Fluids*, Vol. 35, No. 7, aug 2006, pp. 688–692.