Aerospace Science and Technology 80 (2018) 101–126

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

# A comprehensive high-order solver verification methodology for free fluid flows

Farshad Navah *, Siva Nadarajah

*Mechanical Engineering Department, McGill University, Montréal, Québec, H3A 0C3, Canada*

A R T I C L E I N F O

*Article history:*
Received 9 October 2017
Received in revised form 26 June 2018
Accepted 5 July 2018
Available online xxxx

*Keywords:*
Code verification
High-order accuracy
Method of manufactured solutions
RANS
Curved grids
Flux reconstruction

A B S T R A C T

The aim of this article is to present a comprehensive methodology for the verification of computational fluid dynamics (CFD) solvers with a special attention to aspects pertinent to discretizations with orders of accuracy (OOAs) higher than two. The method of manufactured solutions (MMS) is adopted and a series of manufactured solutions (MSs) is introduced that examines various components of CFD solvers for free flows (not bounded by walls), including inviscid, laminar and turbulent problems when the latter are modeled by the Reynolds-averaged Navier–Stokes (RANS) equations. The treatment of curved elements is also examined. These MSs are furthermore conceived with demonstrated suitability for the verification of OOAs up to the sixth. Each MS is as well utilized to discuss salient aspects useful to the code verification methodology such as the relative qualities of the most useful norms in measuring the discretization error, the sensitivity analysis of the verification process to forcing function terms, the relation between residual minimization and discretization error convergence in iterative solutions and finally the sensitivity of high-order discretizations to grid stretching and self-similarity. Furthermore, scripts and code are provided as accompanying material to assist the interested reader in reproducing the verification results of each manufactured solution (MS).



## 1. Introduction

Code verification is a crucial step prior to the application of a scientific simulation software to the solution of practical problems as it aims at examining the soundness of the implementation of the governing equations in the numerical framework. With the increasing interest of the research community in the design and application of high-order-of-accuracy discretization methods for CFD problems, there is an imperative need to extend the verification methodology to this class of methods. Code verification is in fact even more critical for higher-order methods since it is the only means to provide assurance that the effort invested in their design and development is justified by the delivery of the expected higher performance in terms of accuracy per computational effort. We hence present in this paper the fundamental aspects towards a comprehensive code verification methodology for CFD solvers with all orders of accuracy.

To carry out the demonstration of the methodology and without loss of generality, we choose the numerical framework composed of the compressible RANS equations closed by the original and negative versions of the Spalart-Allmaras (SA) model of turbulence [1] and discretized by the correction procedure via flux reconstruction (CPR) scheme [2].

The article is structured as this: in Section 1, the context and contributions of this work are introduced, followed by a comprehensive presentation of the theoretical background in verification and validation (V&V) in Section 2. The governing equations as well as the compact high-order numerical method are respectively exhibited in Sections 3 and 4, including a precise description of all the employed boundary conditions. The application results of the verification and the discussion of the salient aspects of the methodology appear in Section 5 and the article ends with conclusions in the last Section.

### 1.1. Contributions

A series of trigonometric manufactured solutions for the sequential verification of high-order RANS solvers is devised such that it demonstrably achieves all OOAs up to the sixth order on moderately fine isotropic grids, without being trivially reached on the coarsest ones. Attention is invested in ensuring that the MSs produce a fair balance between different terms of the governing equations. The sequence of MSs targets constitutive components of solvers in an isolated fashion and with incremental complexity

* Corresponding author.
*E-mail address:* farshad.navah@mail.mcgill.ca (F. Navah).

https://doi.org/10.1016/j.ast.2018.07.006
1270-9638/

such that systematic debugging is enabled and gathering cumulative evidence on the soundness of high-order CFD solver implementation is made possible. The MSs serve thus to examine the implementation of Euler, Navier–Stokes (NS) and RANS equations along with the original and also with the negative SA model, for free flows, i.e., flows that are not bounded by walls. The set of MSs is as well employed to explore the following concepts:

- The comparative description of different norms and a demonstration of the importance of $L_\infty$ norm in code verification;
- The need for the inclusion of a relatively high order of accuracy in code verification;
- The significance of the balancing of forcing function terms of the MMS and the sensitivity analysis of the verification process to terms with the lowest magnitude in the forcing functions;
- The verification of both the original and the negative SA models of turbulence;
- The relation of residual convergence level with regards to discretization error magnitude and insight on the necessary level of residual convergence;
- The examination of the treatment of non-affine mapping of curved elements;
- The effect of grid self-similarity and stretching on grid convergence of solutions with smooth gradients.

Accompanying IPython [3] notebook and C routine (available at [4]) facilitate the application of the described verification methodology through the reproduction of the manufactured fields and forcing functions of the presented MSs.

## 2. Theoretical background

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

### 2.1. Terminology in V&V

Fig. 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, NS or those of RANS-based turbulence models. Almost in all cases, these models are a mere, yet hopefully reliable, approximation of the inherent complexity of physics and as such they contain a *modeling error*. In this regard, *model validation* questions how well the phenomenon of interest is 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,

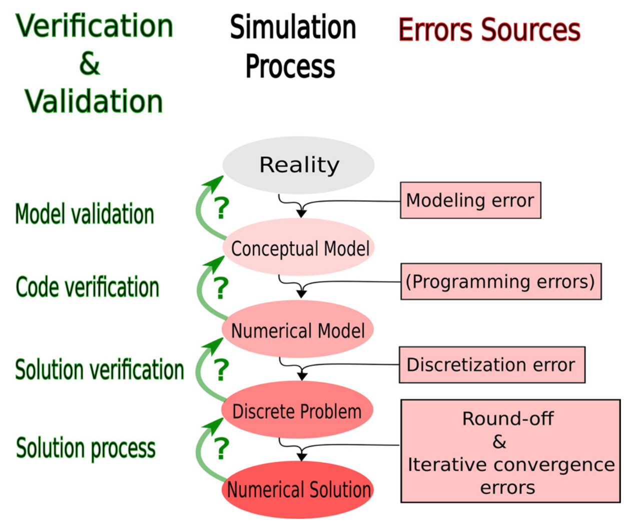


**Fig. 1.** Verification and validation in relation to sources of error in a scientific simulation.

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, the solution of a numerical model is expected to tend towards that of the conceptual model with a rate known as the *formal order of accuracy*. A numerical model 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” [5]. *Code verification* has for purpose to identify and eliminate the mistakes affecting the correspondence between the scientific software and the conceptual model via 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* (DOFs), 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 discrete problem. 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 DOFs.

#### 2.1.1. Code verification methods

The evaluation of OOAs in code verification requires the knowledge of the exact solution which could be provided by devising a