AIAA R-141-2021

Recommended Practice

Code Verification in Computational Fluid Dynamics

AIAA R-141-2021

Recommended Practice for Code Verification in Computational Fluid Dynamics

TABLE OF CONTENTS

Section **Page**

Foreword

This Recommended Practice is intended for: (1) code developers and their managers; (2) researchers, analysts, and applications engineers who use CFD codes for investigation, design, or analysis; and (3) managers who have the ultimate responsibility of deciding on the reliability of computational simulation results used in making decisions.

At the time the AIAA R-141-2021 was completed and approved, the AIAA Committee on Standards for CFD, which is the author of this Recommended Practice, had the following members:

Urmila Ghia, Chairperson
Sami Bayyuk
John Benek
Kenneth Bryden
Robert Bush
Terrence Conlisk
Sami Habchi
Ali Hadid
Robert Harris
Charles Hirsch
Mark Jurkovich
Hyung Lee
Edward Lynch
Mory Mani
Unmeel Mehta
William Oberkampf
Francisco Palacios
Dominique Pelletier
Joseph Powers
Christopher Roy
Christopher Rumsey
Tom Shih
John Schaefer
Christopher Steffen

The above-listed members all voted on this Recommended Practice. The balloters may have voted for approval, disapproval, or abstention. The above consensus body approved this document in January 2021.

The AIAA Committee on Standards for CFD intends to publish a “Standard for Code Verification in CFD” as a sequel to this “Recommended Practice for Code Verification in CFD.” When published, the Standard will supersede and replace this Recommended Practice.

Executive Summary

This Recommended Practice outlines a procedure for Computational Fluid Dynamics (CFD) code verification and identifies the minimum criteria that should be satisfied to assert that code verification has been successfully carried out.

This Recommended Practice defines code verification and explains its role and importance within the overall scheme of code and solution verification, validation, and uncertainty quantification (VVUQ) in CFD. The document emphasizes the need for code verification as a necessary first step in the VVUQ process. It also identifies who should carry out code verification and who should assume responsibility for ensuring that appropriate code verification has been carried out before relying on computational results.

This Recommended Practice provides and explains the main definitions relating to: (1) discretization of the mathematical model equations used in CFD codes; (2) discrete solutions and discrete solution methods; (3) errors in discrete solutions; (4) systematic mesh and time-step refinement; (5) formal order of accuracy; and (6) observed order of accuracy. The code verification process is then explained in terms of comparing the observed order of accuracy with the formal order of accuracy. This document also outlines how round-off errors and iterative convergence errors can be accounted for in code verification and how they can be quantified and controlled.

This Recommended Practice also explains how a code verification program should be matched to a target application, in that the code verification program should be designed to cover all the terms of the mathematical model equations that are exercised in the target application, as well as the methods, inputs, and parameters that are used for the application. The criteria to be used for assessing the outcome of the code verification process should also be chosen *a priori*. Establishing the observed order of accuracy requires determination of the error in the computational solution and this in turn requires the corresponding exact solution or a highly accurate numerical solution to be known. The following three types of reference solution may be used for code verification: (1) traditional exact analytical solutions; (2) manufactured exact solutions; and (3) benchmark computational solutions. The use of each of these types is explained and demonstrated with an example. Sufficient detail is given for each of these examples to enable the process to be repeated for different test cases with different code. This Recommended Practice also briefly discusses the main weaknesses and limitations of code verification, especially its lack of applicability to models with a built-in dependence on the mesh spacing, such as wall-modeled LES, and the difficulty of applying it to coupled or staged computations.

The overall conclusion and recommendation of this Recommended Practice is that any claim of correctness of a code or a computational solution should rest on adequate code verification, whether done by developers, software vendors, analysts, or other end users, and that without code verification, reliance on computational results in making decisions incurs unquantifiable risks.

The body of this Recommended Practice is kept intentionally brief. Additional details are provided in appendices and references to external literature are also given for additional coverage and details