Journal of Computational Physics 294 (2015) 312–328

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Verification of fluid-dynamic codes in the presence of shocks and other discontinuities

C. Nathan Woods [*,1], Ryan P. Starkey [2]

Department of Aerospace Engineering Sciences, University of Colorado Boulder, 429 UCB, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 13 August 2014
Received in revised form 16 January 2015
Accepted 30 March 2015
Available online 3 April 2015

Keywords:
Verification
Shock-capturing
Manufactured solutions
Integrative manufactured solutions
Weak manufactured solutions
Discontinuity propagation

ABSTRACT

The verification that computer codes correctly solve their model equations is critical to the continued success of numerical simulation. The method of manufactured solutions (MMS) is the best method currently available for this kind of verification for differential equations. However, it cannot be used directly with discontinuous solutions, as is required for the verification of high-speed aerodynamic codes with shocks. An integrative approach can extend the applicability of MMS to both discontinuous solutions such as shocks or material interfaces, as well as integral equations. We present an implementation of integrative MMS based on intelligent subdivision of integration domains that is both highly accurate and fast, and results in a rigorous, one-step verification procedure for shock-capturing codes. Numerical integration is found to be accurate to machine precision when tested on exact solutions of the linear heat equation and the Euler equations, even in the presence of discontinuous flow features. Intelligent subdivision of integration domains also improves computational performance by approximately 60× compared to the same algorithm without intelligent subdivisions. We demonstrate the use of MMS in the verification of the `BACL-Streamer` inviscid gas dynamics code. Integral MMS is found to compute convergence rates that are equivalent to those computed using differential MMS, and comparable to those computed using discontinuous, exact solutions, suggesting integral MMS is a valid method for verification of both integral and shock-capturing codes.

1. Introduction

The development of computational science has revolutionized scientific research and engineering design, and is a crucial tool in advancing our understanding of complex phenomena. As the demands placed on computational models and the computer codes used to solve them become more stringent, it is becoming increasingly important that researchers understand their limitations precisely and reliably. The National Research Council [1] identified code verification as a crucial component in the quantification of this understanding, and specifically identified the method of manufactured solutions (MMS) as a key tool in this process.

The limitations of numerical simulation arise naturally out of the physical and numerical approximations that are used, and so it is important to both understand and quantify the errors introduced by these approximations. Roache [2] classified simulation errors as:

* Corresponding author.
E-mail addresses: woodscn@colorado.edu (C. Nathan Woods), rstarkey@colorado.edu (R.P. Starkey).

[1] Ph.D. Candidate.
[2] Assistant Professor.

http://dx.doi.org/10.1016/j.jcp.2015.03.055
0021-9991/

1. Errors that are a result of modeling approximations, such as fluid incompressibility, etc.
2. Errors that are ordered by some measure of the problem discretization.
3. Errors that are the result of some other non-physical approximation, e.g. far-field boundary conditions.
4. Errors in programming (mistakes, or bugs).
5. Errors that result from the representation of numbers on a computer.

The study and quantification of modeling errors for a given application is known as model validation, while the study of the remaining mathematical, numerical, and programming errors is known as verification [2]. Verification is further subdivided into code verification and solution verification, where code verification is used to show that the code or software solves a given mathematical model correctly within some domain of inputs, while solution verification estimates the expected numerical error in a solution to a specific problem or application.

One of the most powerful features of code verification is the ability to analyze the errors resulting from programming mistakes directly. Software debugging is the most difficult and time-consuming part of scientific code development, and modern scientific computing codes are so complex that it is impossible to reliably eliminate all coding mistakes. Thorough code verification allows scientists to at least eliminate any mistakes that introduce error in the computed solution. Unfortunately, thorough verification, or verification that exercises all aspects of a code, can be difficult. The measurement of convergence rates to complex, manufactured solutions is the best technique currently available [2–6], but it is limited in scope, and many important physical systems are simply unsuitable for verification with manufactured solutions as currently available.

One of the principal limitations of MMS is that it cannot be applied directly to problems that admit discontinuous solutions [3,4]. Discontinuities arise in many branches of physics and engineering, and they are an important aspect of many scientific models. These discontinuities can be part of the problem specification, such as material interfaces, or they can arise naturally from the mathematical model, such as aerodynamic shock waves. In order to establish trust in the codes used to study these phenomena, it is absolutely essential to develop verification tools and techniques that can be directly applied to simulations that contain them.

The root of the problem is that discontinuous solutions to systems of differential equations are not, in fact, solutions to differential equations. Rather, they are solutions to related systems of integral equations, and many computational frameworks for solving systems of this kind are themselves integral in nature. It is therefore natural to approach the problem of discontinuous manufactured solutions as an integral problem. In order to do this effectively, one must also have some means to accurately integrate complicated, multidimensional, discontinuous functions.

The remainder of this paper is structured as follows. In Sections 2 and 3, we will present a brief overview of code verification and MMS in general. In Section 4, we will discuss in detail one method for accurately integrating general discontinuous functions. In Section 5, we will describe a computational tool that can be used to automate the computation of integral manufactured source terms, and we will also use this tool to verify the accuracy of the integration method described in Section 4. Finally, in Section 6 we will demonstrate a simple application using MMS for verification of a computational fluid dynamic code.

2. Code verification and MMS

2.1. What is verification?

As discussed, verification is the study and quantification of errors resulting from mathematical and numerical approximations and the elimination of errors resulting from programming mistakes. Code verification is the specific process by which one ascertains that a code correctly solves its model equations within some domain of inputs. This is done by directly testing the results of a code for various problems, and the quality of verification depends on the type of testing that is done.

The most powerful, accurate, and reliable method of verification is known as code order verification [4]. For a code that is working as intended, error is expected to be dominated by discretization errors, which scale as Δx^n, where n is determined by the algorithms in use by the code. In code order verification, the code is used to compute a solution to a problem with a known exact solution under successive refinements of the code discretization. The actual rate at which the code converges to the correct solution is computed based on the error between the computational results and the exact solution, and this is compared to the nominal expected value. If the observed convergence rate matches the theoretical convergence rate, then the code is considered verified.

2.2. Manufactured solutions

Code order verification, while clearly defined and very reliable, has traditionally suffered from the scarcity of exact solutions for systems of interest. MMS was developed to resolve this problem, by allowing the simple, direct generation of complex solutions for the purpose of code verification [3–7].