Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 2560–2581
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers

Hiroaki Nishikawa [a,*], Keiichi Kitamura [a,b]

[a] *W.M. Keck Foundation Laboratory for Computational Fluid Dynamics, Department of Aerospace Engineering, University of Michigan, FXB Building, 1320 Beal Avenue, Ann Arbor, MI 48109-2140, USA*
[b] *Fluid Dynamics Laboratory, Department of Aerospace Engineering, Nagoya University, 1 Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan*

Received 8 August 2007; received in revised form 29 October 2007; accepted 4 November 2007
Available online 17 November 2007

Abstract

In this paper, we propose new Euler flux functions for use in a finite-volume Euler/Navier–Stokes code, which are very simple, carbuncle-free, yet have an excellent boundary-layer-resolving capability, by combining two different Riemann solvers into one based on a rotated Riemann solver approach. We show that very economical Euler flux functions can be devised by combining the Roe solver (a full-wave solver) and the Rusanov/HLL solver (a fewer-wave solver), based on a rotated Riemann solver approach: a fewer-wave solver automatically applied in the direction normal to shocks to suppress carbuncles and a full-wave solver applied, again automatically, across shear layers to avoid an excessive amount of dissipation. The resulting flux functions can be implemented in a very simple and economical manner, in the form of the Roe solver with modified wave speeds, so that converting an existing Roe flux code into the new fluxes is an extremely simple task. They require only 7–14% extra CPU time and no problem-dependent tuning parameters. These new rotated fluxes are not only robust for shock-capturing, but also accurate for resolving shear layers. This is demonstrated by an extensive series of numerical experiments with standard finite-volume Euler and Navier–Stokes codes, including various shock instability problems and also an unstructured grid case.

Keywords: Shock instability; Carbuncle; Hybrid schemes; Rotated Riemann solvers; Upwind schemes

1. Introduction

In this paper, we propose new flux functions for use in a finite-volume Euler/Navier–Stokes code, which are very simple, carbuncle-free, yet have an excellent boundary-layer-resolving capability, by combining two different Riemann solvers into one based on a rotated Riemann solver approach.

* Corresponding author. Present address: National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666-6147, USA. Tel.: +1 757 325 6906.
E-mail address: hiro@nianet.org (H. Nishikawa).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.11.003

Despite a great deal of progress made in the past decades, current finite-volume Euler codes have a hard time suppressing unacceptable flow fields, often referred to as the carbuncle, e.g., a distorted bow shock as if a thin plate were inserted at the center perpendicular to the shock [1,2]. This happens, unfortunately, with accurate (less dissipative) flux functions such as Roe's approximate Riemann solver [3], which recognizes all wave components in the Riemann problem and introduces minimal dissipation to better resolve discontinuities [4]. Several attempts have been made to understand and cure the phenomenon [4–10], but, although they offer several useful explanations and guides to prevent it, none of them have appeared to reach yet the complete understanding and the cure of the problem [11]. There is, however, a class of flux functions that is known to be 'carbuncle-free' (if not perfectly carbuncle-free). This includes the Rusanov solver and the simplest version of the HLL Riemann solvers [12,13]. These are approximate Riemann solvers with one-wave and two-wave approximations respectively. These flux functions are very robust for inviscid calculations involving shocks, but unacceptably dissipative for shear layers, particularly for boundary layer calculations with a Navier–Stokes code.

Here, we propose a method to combine these carbuncle-free flux functions and the Roe flux function, such that shear layers are better resolved while the robustness of the dissipative fluxes is retained for shock-capturing, in a very simple but a systematic manner. To combine two flux functions, we use the rotated Riemann solver approach [14–16], which adaptively selects a direction suitable for upwinding and applies a Riemann solver along that direction, in order to capture multidimensional flow features as accurately as possible by one-dimensional physics. Originally, this approach was proposed to better resolve shocks and shear layers [14,15], but in [15] it was concluded that the gain was not very impressive in second-order accurate computations. Later, Ren [16] took this approach, not to improve accuracy, but to gain robustness for shock-capturing. He decomposed a cell-face normal into two directions: one aligned with the velocity difference vector (normal to shocks and parallel to shears) and the other orthogonal to it, and then applied the Roe solver along them. This rotated flux was shown to suppress the carbuncle by an extra dissipation introduced by the rotated flux mechanism. This, of course, does not come free; an additional cost has to be paid. He claims that the factor of CPU time is not 2 but can be made 1.5 if smartly implemented.

We follow Ren's work, but there are significant differences. One is that we apply two different Riemann solvers in the two directions. We do not rely on the additional dissipation introduced by the rotation mechanism itself, but rather apply a carbuncle-free flux function directly to the velocity difference vector direction. In the other direction, we employ the Roe solver to prevent the resulting flux from being too dissipative. This means that the combined flux becomes the carbuncle-free flux normal to shocks and the less dissipative Roe flux across shear layers. Therefore, the resulting flux has what it takes to be carbuncle-free as well as boundary-layer-resolving. Moreover, we will show that the resulting fluxes can be implemented as simple and economical as the Roe flux with an entropy fix, and they require only about 7–14% more CPU time than the Roe flux (a factor of 1.07–1.14) including the cost of computing the velocity difference vector. Another difference lies in the definition of the decomposed normal directions. It has been customary to use a cell-face normal when the velocity difference vector is too small [16]. Here, we propose to take a cell-tangent instead of normal. This does not make any differences for Ren's rotated solver (because the same flux is used in both directions), but does bring a significant impact on our rotated solvers. With this modified definition, the Roe flux will be activated instead of more dissipative solvers, for smoothly varying flows, thus leading to accuracy improvement.

The idea of hybridizing the Roe flux and the HLL flux is not new. This has already been considered by Quirk [2] to cure the carbuncle. He used a pressure gradient to detect a shock, and then switch the flux from the Roe to the HLL to avoid the carbuncle. Although very effective, the resulting algorithm does not seem to have gained popularity, apparently because of its empirical nature in the shock-sensor and also the discontinuous switching between two different fluxes. Another hybridization was proposed later by Janhunen [17] for magneto-hydrodynamics simulations, based on a positivity check on the density and the pressure. Yet, there are still other fluxes that may be considered as a combination of these two fluxes, [8,18], for example. In this paper, we construct hybrid fluxes by using the rotated Riemann solver approach, which are as simple, economical and predigested as a single Riemann solver, and do not require any problem-dependent tuning parameters.

The paper is organized as follows. In the next section, a finite-volume formulation of the Euler equations is given. In Section 3, base Riemann solvers are described, which will be combined later to construct new Riemann