Computers & Fluids 96 (2014) 264–275

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Design and verification methodology of boundary conditions for finite volume schemes ☆

D. Folkner [a,*], A. Katz [a], V. Sankaran [b]

[a] Department of Mechanical and Aerospace Engineering, Utah State University, Logan, UT 84322, USA
[b] Rocket Propulsion Division Air Force Research Laboratory (ARFL), Edwards AFB, CA 93524, USA

ARTICLE INFO

Article history:
Received 8 April 2013
Received in revised form 25 March 2014
Accepted 26 March 2014
Available online 5 April 2014

Keywords:
Boundary conditions
Finite volume
Lagrange multipliers
Manufactured solutions
Verification

ABSTRACT

Despite the critical importance of resolving flow at or near domain boundaries, boundary condition formulations are often implemented in an ad hoc fashion. The purpose of this work is to provide a general framework for the implementation and, importantly, the verification of boundary conditions for node- and cell-centered finite volume schemes. These conditions may include any combination of Dirichlet conditions, Neumann conditions, extrapolation, and the conservation equations themselves. Specific conditions for inviscid walls, inflow, and outflow are systematically tested using manufactured and exact solutions to assess well-posedness and stability. The procedures in this work provide a method for the verification of complex boundary conditions as well as shed physical insight for different problem configurations.

1. Introduction

Computational Fluid Dynamics (CFD) is increasingly being applied to applications involving greater complexity than ever before. Common to most of these applications is the requirement for high levels of accuracy at or near domain boundaries. Examples include calculation of aerodynamic lift and drag, computation of boundary layer characteristics, estimation of surface heating, and mass flux computation. For these and other cases, the regions near boundaries are the primary focus of the CFD simulation and require the greatest resolution and numerical accuracy.

Because of the need for high accuracy near boundaries, many researchers over the years have focused on proper implementation and analysis of boundary conditions. Such analysis, especially in the context of finite difference schemes, has a long history (see, for example, the works of Gustafsson [1] and Kreiss and Sherer [2]). The subsequent emergence of finite volume schemes raised new questions regarding the stability and well-posedness of boundary procedures. Numerous boundary treatments have been proposed for finite volume schemes in an attempt to maintain accuracy and stability. For example, many treatments have been proposed for inviscid walls. The method by Rizzi [3] involves the use of the momentum equation to obtain the pressure. Jameson proposed direct modification of the flux at an inviscid wall to produce zero convective flux contribution [4]. Dadone and Grossman advocate a curvature corrected symmetry condition for an inviscid wall [5]. Balakrishnan and Fernandez recommend a variety of other methods involving additional quantities such as entropy and enthalpy [6]. Numerous other strategies exist for inviscid walls as well as for other boundary conditions, such as inflow, outflow, and no-slip walls. The difficulty is that many of these methods have not been rigorously verified and may or may not be consistent with the interior discretization schemes. Addressing this issue, Choudhary et al. [7] recently proposed a boundary verification procedure using the method of manufactured solutions (MMS). Their approach requires carefully constructed manufactured solutions that already satisfy the boundary conditions, which means that a new manufactured solution needs to be constructed for each boundary condition and geometry. Nonetheless, their work represents an important step towards comprehensive code verification since most previous verification strategies have neglected boundary effects [8–12].

This work provides an alternate method of boundary verification by focusing on three main goals. First, we provide a general framework for implementing boundary conditions for both node- and cell-centered schemes. The framework applies to arbitrary boundary conditions for any physical system. Second, we provide a rigorous and general approach for the verification of these boundary conditions based on MMS wherein arbitrary manufactured

☆ Distribution statement A: Approved for public release; distribution is unlimited.

* Corresponding author.
E-mail addresses: David.Folkner@gmail.com (D. Folkner), aaron.katz@usu.edu (A. Katz), venkateswaran.sankaran@edwards.af.mil (V. Sankaran).

http://dx.doi.org/10.1016/j.compfluid.2014.03.028
0045-7930/

solutions are used in concert with the appropriate formulation of the boundary condition equations. This approach allows for direct measure of the combined discretization error of the interior and boundary treatments. This is essential for finite volume schemes, for which traditional finite difference-type truncation error analysis does not accurately represent the true order of discretization error [13]. Third, we demonstrate the importance of choosing physically correct boundary conditions for specific physical configurations. We find that often the choice of well-posed and stable boundary conditions is not unique. However, certain boundary conditions can lead to erroneous results, often in subtle ways.

In order to accomplish these goals, we explore the formulation of a variety of boundary types, including inviscid walls, inflow, and outflow, for both node- and cell-centered finite volume schemes. For each of these boundary types, we explore a variety of conditions that use a combination of Dirichlet, Neumann, and extrapolation conditions. Importantly, we highlight the fact that node-centered schemes easily allow the direct use of the governing equations of motion (mass, momentum, and energy) at the boundaries, while cell-centered schemes do not, instead relying on extrapolation or other conditions. Other boundary condition types and implementations beyond those discussed in this work are certainly possible. Here, the principal objective is to provide a framework for the implementation and verification of any boundary condition.

The paper is organized as follows: First, boundary condition formulations are explored for node- and cell-centered finite volume schemes. We present a common discretization framework to test a variety of governing boundary equations. We demonstrate how to verify these boundary conditions using MMS. We then present grid refinement and other qualitative studies. First, a quasi-1D nozzle is tested with both MMS and exact solutions. Next, we extend the use of MMS to two dimensions with wall boundary conditions. Finally we present error convergence results for Ringleb flow and a NACA 0012 subsonic airfoil. We then offer some conclusions derived from these studies.

2. Boundary condition implementation

In this work we test a variety of boundary condition implementations and assess the resulting impact on accuracy through rigorous verification studies. In the interior of the domain, we solve the steady Euler equations,

$$\frac{\partial Q}{\partial \tau} + \nabla \cdot \boldsymbol{F} = 0, \tag{1}$$

where $Q = (\rho, \rho u, \rho v, \rho e)^T$ is the vector of conserved variables, and $\boldsymbol{F}$ is the inviscid flux vector. Here, ρ is density, u and v are the Cartesian velocity components, and e is the total energy per unit mass. While we focus on inviscid flows in this work, the methods described herein apply equally well to viscous flows. We are interested in solving the steady equations to which we add the pseudo-time (τ) derivative for convenience in marching to steady state. We test both node- and cell-centered finite volume spatial discretizations, which result in a semi-discrete set of non-linear equations of the form

$$\frac{\partial Q}{\partial \tau} + R(Q) = 0, \tag{2}$$

where $R(Q)$ is the steady residual at either an interior node or an interior cell location depending on the discretization procedure. Both cell- and node-centered methods use linear least squares gradient procedures to reconstruct left and right states with CUSP artificial dissipation [14,15]. Limiters are not employed in this work because all test cases make use of smooth solutions in order to verify order of accuracy. Explicit Runge–Kutta time stepping is used to reach steady state for both node- and cell-centered codes.

In addition to the interior discretization scheme, we must incorporate boundary conditions to close the system of equations on a finite domain. Here we focus on inviscid wall, inflow, and outflow conditions in order to explore fundamental issues of stability, well-posedness, and numerical accuracy. The methodology developed here is quite general and directly applies to other boundary condition types as well. For node-centered discretizations, boundary conditions are enforced directly at the boundary nodes coincident with the physical boundary, shown in Fig. 1a. For cell-centered discretizations, we introduce additional unknowns in the form of ghost nodes located at the flux quadrature points of the boundary faces, shown in Fig. 1b. The ghost nodes are then used in an upwind flux formula to determine the numerical flux through the boundary face. In this manner, the cell-centered boundary formulation remains water-tight. The node-centered configuration, however, is not strictly water-tight since the fluxes surrounding the boundary nodes do not always cancel with nearby interior nodes.

For both node- and cell-centered formulations, we define a "boundary residual", $R_b(Q)$, which we drive to zero at steady state along with the interior residuals, $R(Q)$:

$$R_b(Q) = 0. \tag{3}$$

In a node-centered discretization, R_b replaces R at the boundary nodes. In cell-centered discretizations, R_b provides the governing equations for the ghost nodes. Boundary nodes and ghost nodes may then be used in any aspect of the discretization scheme, including flux computations and gradient reconstruction. In all, we test fifteen different boundary implementations, which are listed with a common notational convention in Table 1 for clarity. The methods involve a certain number of Dirichlet-specified quantities, with the state specification completed by additional methods, such as Neumann conditions, extrapolation, or in some cases, the equations of motion themselves. We will refer to this table as we develop various forms for R_b in the following sections.

2.1. Node-centered boundaries

All boundary conditions involve the specification of a certain number of Dirichlet (or Neumann) conditions augmented by additional information derived from the interior field. With node-centered schemes it is straightforward to select some combination of the governing equations of motion (mass, momentum, and energy) to enforce at boundary nodes. This is because the boundary nodes lie within control volumes for which we can easily implement flux balances. In contrast, cell-centered boundary conditions require the use of ghost nodes for which there is no natural control volume or flux balance. Thus, it becomes difficult to apply the equations of motion directly at the ghost nodes. Instead, we choose other methods to define the ghost node state such as solution extrapolation.

Enforcement of boundary conditions in a node-centered scheme involves the specification of the boundary residual, R_b, directly at the boundary nodes shown in Fig. 1a. In this section, we outline a procedure to obtain the boundary residual by multiplying the governing equations of mass, momentum, and energy by a selection matrix to complete the boundary conditions. A subset of this approach uses Lagrange multipliers as discussed by Allmaras [16]. In addition, we discuss a second method involving a commonly used weak boundary condition for an inviscid wall.

2.1.1. Selection matrix method

One method of obtaining a boundary residual is through a selection matrix that picks desired combinations of the equations of motion. The finite element community has used such an approach