On the Acoustic Component of Active Flux Schemes for Nonlinear Hyperbolic Conservation Laws

by

Duoming Fan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering and Scientific Computing)
in the University of Michigan
2017

Doctoral Committee:

Professor Philip L. Roe, Chair
Assistant Professor Karthik Duraisamy
Associate Professor Krzysztof J. Fidkowski
Assistant Professor Eric Johnsen

Duoming Fan

dufan@umich.edu

ORCID iD: 0000-0002-3246-4315

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the people who have helped and encouraged me through my PhD studies here at the University of Michigan. I am foremost exceptionally grateful to my advisor, Phil Roe, for his continuous guidance and insight into the field of CFD. Our discussions were always helpful and usually ended with more questions than I had to begin with, but I believe this has made me a better researcher and problem solver. And while I was impressed by Phil's vast knowledge of the technical, and sometimes philosophical aspects of the field, I feel that I have been impacted more by his outlook and approach to life outside of the academic setting. He and his wife, Jacqui, have always made me feel very welcomed ever since the first time I met them, and I will always remember and appreciate those warm gestures.

I would also like to thank Prof. Karthik Duraisamy, Prof. Chris Fidkowski, and Prof. Eric Johnsen for agreeing to be on my dissertation committee, and providing valuable and thoughtful feedback on my research. A special thanks goes out to Chris and his student Kyle Ding for producing some of the DG results in this thesis.

I wish to thank my parents for their constant support and care throughout my graduate studies and beyond. I would not be who or where I am today without their unwavering encouragement and belief in the decisions I've made. I am also fortunate to have my boyfriend, Jimmy, share our experiences and hardships through graduate school together. And last, but not least, I want to thank all of my friends here in the Aerospace department who have made my journey through PhD very fun and memorable. I will always remember those football games, late-night karaokes, game nights, daily runs, and lots of other memories that we have made together. There are too many of you to list here, but know that I treasure each and every one of you and hope that we can continue to stay as life-long friends.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

Current numerical methods used in production-level CFD codes are found to be lacking in many respects; they are only second-order accurate, rely on inherently one-dimensional solvers, and are ill-equipped to handle more complex fluid flow problems such as turbulence, aeroacoustics and vortical flows just to name a few. Recently, a new class of third-order methods known simply as Active Flux (AF) has been introduced to address some of these issues. The AF method is best understood as a finite-volume method with additional degrees of freedom (DOF) at the interface to independently evolve interface fluxes. It is a fully discrete, maximally stable method that uses continuous data representation, and because the interface fluxes are computed independently from the cell-average values, true multidimensional solvers can be used.

This dissertation focuses on the development of the AF method aimed at solving conservation laws describing acoustic processes. The method is demonstrated for linear and nonlinear acoustic equations in two-dimensions as well as for the full Euler equations where we employ operator splitting between the advective and acoustic processes. Given its continuous representation, the AF method economically achieves third-order accuracy using only three DOF in two dimensions, which is comparable to the discontinuous Galerkin method using linear reconstruction (DG1). A direct comparison between the two methods for acoustic problems finds that the AF method is capable of matching the accuracy of DG1 with a mesh spacing about three times greater and uses time steps about 2.5 times longer. The AF solutions also display superior circular symmetry with significantly less scatter than DG1, which we attribute to the method being able to employ truly multidimensional solvers. In addition, we find that on the same grid and to achieve the same level of error, the computation time for the AF method is more than one magnitude less than DG1 and approximately 3 to 5 times less than DG with quadratic reconstruction (DG2). Finally, various boundary conditions are introduced and developed for the AF scheme including far-field and curved wall boundaries.

CHAPTER 1

Introduction

1.1 Motivation

The role of computational fluid dynamics (CFD) in engineering applications nowadays is so strong that it is considered by some to be the ‘third dimension’ in fluid dynamics, the other two dimensions being the classical predictions of pure experiment and pure theory [2]. With its ability to solve the governing equations without the use of simplified physical models as is done with pure theory, and to include detailed physical phenomena that may be difficult to incorporate in real-life experiments, CFD has rapidly become a popular tool in engineering analysis. The aerospace industry in particular has embraced CFD for decades in the prediction of various fluid dynamic problems that are too complicated to solve analytically. These predictions are then used to make decisions regarding the design of various parts of an aircraft, and is a much cheaper option than testing numerous designs in wind tunnel experiments.

In light of such importance in the aerospace industry, the advancement of CFD methods has been linked closely to performance and cost. The current industry standard for CFD applications is a second-order accurate finite volume (FV) code with a rigid code structure that only allows changes to its modules, either in spatial or temporal methodology. The reasons for keeping this particular form of code are numerous. The first is that the computational cost of a second-order accurate method is low enough to allow for the fast turnaround time required for design purposes. Its order of accuracy is also sufficient enough to solve the fluid dynamic problems that were studied in the earlier days of CFD. The code has proven to be robust and incorporates well-developed features such as limiters and compact support that allows for easy parallelization. It is also easily adapted across a wide variety of engineering fields. But perhaps the most pragmatic reason for the rigidity of the current CFD production codes is the extensive time investment that would be required to make major algorithm changes. Thus, without the emergence of a new competitive method that is robust, compact, and far more accurate and cost-efficient than the current FV code, it is unlikely that the current landscape of CFD production codes will change.

Despite the industry's cautious attitude towards adopting CFD methods that do not fit into their existing framework, we believe that a well-designed third-order accurate method may have the necessary advantages to change their view. The first issue with second-order methods is that while they are accurate enough to compute solutions for the types of problems that were analyzed a few decades ago, it is becoming increasingly difficult for them to handle the complexity of fluid dynamic problems that need to be solved today. This includes the addition of viscous effects, time-dependent physics, and three-dimensional phenomena such as turbulence. The consequence of its relatively low order of accuracy has not been strongly felt yet due to the massive advances in computational power and code parallelization, but it will eventually become a bottleneck to solving even more complex problems in the future. In addition, with the recent development of high-order methods, we find that they are in fact more cost-efficient than lower order methods to achieve the same level of error [1]. However, the advantages of the currently available high-order methods are not great enough or well-established enough to tempt designers into changing their tools, and many of these methods suffer from a lack of robustness, which is a quality that is required in production codes.

There is some debate about the optimum order of accuracy for a numerical scheme. Fourier analysis for the advection equation concludes that even-order schemes are prone to exhibiting dispersive leading term errors while odd-order are prone to dissipative error [3]. Therefore, even- and odd-ordered schemes behave very differently at sharp features or discontinuities; even-ordered schemes tend to result in much more oscillatory behavior than odd-ordered. Practically, there is much to be gained from increasing from second- to third-order accuracy; since the oscillations are more manageable in a third-order scheme, its limiters would need to do less work. There is little incentive to increase to fourth-order accuracy because it would bring the oscillations back, and the next odd-ordered scheme at fifth-order would require significantly more memory storage and computational cost to run. For all of the reasons given above, we believe that third-order methods are best suited for practical CFD applications.

1.2 Current practices of high-order methods

Many high-order methods have been developed in the research community in the past two decades, but they seem to be far from assuming any relevant role in aircraft design. We argue that this is in part due to the fact that they retain many of the standard features of conventional FV schemes and other part due to the inaccurate representation of the physics behind the governing equations being solved. The following sections describe some of the practices and beliefs of the current classes of high-order methods that may be questionable or even unsuitable for solving multidimensional conservation laws.

1.2.1 Discontinuous representation

Since its introduction in the early 1970's, the discontinuous Galerkin (DG) method has received a wide reception from the CFD community to solve both compressible and incompressible flow problems. DG methods seek to approximate the solution in each cell independently by using test functions that are restricted to the local elemental space. By breaking up the data at cell interfaces and having the ability to approximate the local solution to any order of accuracy using the test functions, a highly compact high-order method emerges. Compactness is considered to be a very important feature to both researchers and industry alike; researchers argue that hyperbolic problems are local phenomena that should not depend on solutions far away, and for practical applications, a compact method is easily scalable to parallel architectures. Further favorable properties of the DG method include being locally conservative and stable.

Despite their numerous advantages, DG methods have failed to be adopted by industry for two main reasons. The first is a lack of robustness for nonlinear problems [4]. The second, and perhaps more influential reason is their high memory requirement and computational expense. Storing discontinuous data local to each cell means that there are two independent values overlapping at every node along an edge, and on average six different values at a vertex even in two-dimensions. Significant savings in memory storage could be gained by a scheme employing continuous data instead, but that could result in the regular Galerkin method with a global mass matrix that would cost much more to invert and solve. In addition, savings in memory storage becomes particularly important as we move to exascale computing, where flops are cheap but memory storage is limited for each individual processor.

From a fundamental point of view, introducing discontinuities across cells invokes the necessity for the Riemann solver to resolve the differences. In essence, these discontinuities are being treated by the method as actual flow features when in reality they do not correspond to anything remotely resembling real flow features and result in producing numerical artifacts. In fact, Roe demonstrates in [5] that the highly praised discontinuous nature of DG methods is actually equivalent to the behavior of a class of continuous methods for the one-dimensional linear advection equation. We can then argue that it is unlikely that its use of discontinuous reconstruction is what makes the DG method somehow superior to other methods that use continuous representation.

1.2.2 Riemann solvers

The Riemann solver was introduced in Godunov's original shock-capturing method [6] and has since then become the standard device which almost all CFD schemes use to resolve solution discontinuities at the cell interface. It was surprisingly easy to extend to two- or three-dimensions by solving the Riemann problem normal to the cell face, but it is not without its flaws. For one, the assumption that all waves travel normal to cell interfaces usually results in a strong nonphysical wave being generated that is dependent on the grid geometry instead of being a real flow feature. Another shortcoming of Riemann solvers is dealing with multidimensional low speed flows where the pressure perturbations are of order $\mathcal{O}(M^2)$. But because Riemann solvers assume one-dimensional wave motion, it resolves pressure perturbations on the order of $\mathcal{O}(1)$ instead. This mismatch in the order of pressure perturbations has led to numerical difficulties for some high-order methods when solving flow problems at low Mach numbers. The final disadvantage of Riemann solvers that we would like to discuss is its representation of fluxes at cell interfaces. By solving the Riemann problem normal to the interface, the resulting fluxes represent the flow through only the cell face. However, fluxes in multidimensions have a tensorial nature and are associated more with points and vertices than with faces, as demonstrated by some truly multidimensional numerical methods in [7, 8, 9]. For the reasons given above, Riemann solvers, by their one-dimensional nature, are ill-suited for solving multidimensional problems.

1.2.3 Lack of multidimensional physics

As we mentioned previously, a great majority of numerical methods for solving hyperbolic conservation laws use Riemann solvers to resolve fluxes across cells. This includes popular high-order methods such as DG and weighted essentially non-oscillatory (WENO) [10]. This approach generally works well for smooth problems, but when there is a shock or discontinuity, artificial numerical features, such as carbuncles and oscillations behind an oblique shock, can be generated if the shock is not aligned with the grid. Numerous attempts have been made to mitigate these solution irregularities including using a combination of reduced-wave solvers such as HLL or Rusanov and full Riemann solvers [11], multidimensional corner treatment (MCT) method introduced by Colella [12], and various rotated Riemann methods [13, 14, 15] that decompose the data at the cell interface in order to solve two Riemann problems in different directions. While these techniques are effective at fixing the symptoms of the problem, they are often more computationally complex and do little to address the fact that the modified solvers are still based on one-dimensional physics.

Roe has argued in [16, 17] that multidimensional physical processes in nonlinear systems must be modeled by a numerical method that considers the proper domain of dependence. For an acoustic process that is omnidirectional, this means that the method must use information from a circular domain in two dimensions, and spherical in three, which is fundamentally impossible for an inherently one-dimensional scheme such as the Riemann solver to accomplish. There have been previous attempts at creating high-order schemes without the use of Riemann solvers, the most notable being fluctuation splitting (FS) or residual distribution (RD) methods first introduced by Roe for linear advection [18]. An extension of the method to hyperbolic-elliptic systems by Rad has been demonstrated to achieve third-order accuracy and be able to nearly recover potential flow around elliptical bodies for steady state problems [19, 20]. The RD methods relies on 'distributing' the solution residual to the nodes according to the type of physics involved in the process; upwinding for advection-type process or least-squares for Cauchy-Riemann-type. Unfortunately, some of the concepts of the method could not be extended to three-dimensions and a pseudo-time stepping technique may be needed for unsteady cases [21].

Multidimensional methods that do not use Riemann solvers have also been developed for solving acoustic equations. Most commonly, in two dimensions, the update at a point is found by using bicharacteristics of the solution [22]. The main disadvantages of this method are that they involve integrals that may be difficult to approximate and they are second-order accurate at best. A more recent method has emerged that uses the Poisson solution to derive an integral evolution formula for the scalar wave equation [23]. This method is perhaps the most similar to ours in application, and has been shown to be both robust and stable in two- and three-dimensions when applied to high-order polynomial approximations on square grids [24, 25].

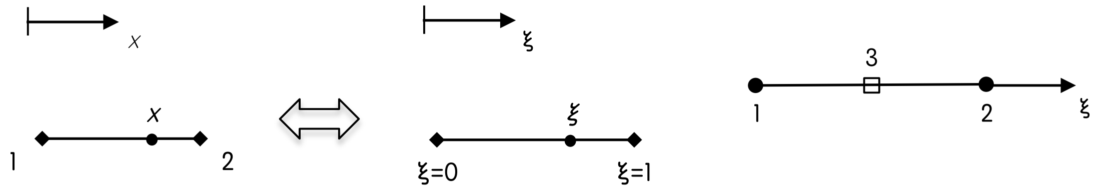
1.2.4 Separation of space and time

Expressing the governing equations in a semi-discrete form has been the welcomed standard in the aerospace community ever since the early 1980's. For unsteady flows, the evolution in time, usually by some variant of the Runge-Kutta (RK) method, is considered separately from the spatial discretization. The advantages of this formulation are that the methods are easy to code and the two modules consisting of spatial and temporal methods can be considered separately at different orders of accuracy. This also plays well into industry standards due to the modular nature of the formulation. However, the major drawback of this separation of space and time is that there is usually a much lower limit on stability, which becomes even more severe as the accuracy of the spatial discretization increases. In addition, one advancement step in time requires a number of stages that each need a full set of residual evaluations, which makes high-order methods computationally expensive. Taking the DG scheme using quadratic reconstruction (DG2) as an example, this third-order method uses RK5 to evolve in time and its stability limit is a CFL number of $1/(2p+1) = 0.2$. This means that the DG2 scheme requires approximately 25 times more computational cost than a fully discrete, maximally stable method to complete a single time step. It is for these reasons that we abandon the idea of using the semi-discrete formulation in our third-order method.

Instead, we choose to go back to early CFD codes that were based on the Lax-Wendroff method, which achieves accuracy by expanding the solution as a series in time and is fully discrete in nature. Although several drawbacks of the original scheme, such as restrictive stability limits and loss of accuracy for nonlinear problems, were eventually resolved, the new forms of Lax-Wendroff schemes are seldomly used in the CFD community today because they fail to produce the necessary multidimensional features. Part of the reason for their failure is that they frequently employed dimensional splitting, which is a very poor form of operator splitting due to the fact that the operators do not commute [26]. More recently, Lax-Wendroff rotated Richtmyer schemes [27, 28] have been shown to preserve discrete vorticity in acoustic problems and are stable at the maximum CFL condition of 1.0 on square grids. The discrete vorticity is preserved by tuning the free parameters in the two-dimensional Lax-Wendroff scheme to reflect the isotropic nature of the acoustic process. Our method is also of the Lax-Wendroff type, but the correct physics is incorporated using spectral integrals on unstructured grids instead.

1.3 A new approach: Active Flux method

The Active Flux (AF) method is a new *class* of methods that aims to resolve the issues with current high-order methods discussed in the previous sections. It is based on Scheme V, concluded by van Leer in his 1977 paper as the most economical of the six schemes he constructed to achieve better than first-order convergence for the one-dimensional advection problem [29]. Scheme V introduces edge values in addition to a cell-average value to each cell, and then uses the extra degrees of freedom to increase the accuracy of the solution to third-order. By construction, all of van Leer's methods are always stable up



(a) Mapping from physical space to reference space.

(b) Node placement for AF basis functions.

Figure 1.1: Geometry and data storage for AF method in one-dimension.

to a Courant number of 1.0. Furthermore, there are many other features of Scheme V that satisfies our criteria for building a new third-order scheme; it is fully discrete, compact, and continuous at the cell interface which reduces memory storage and circumvents the use of Riemann solvers. The method obtains its upwinding property by using a semi-Lagrangian method to advance the solution.

To extend Scheme V, we consider conservation laws of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0 \tag{1.1}$$

where $\mathbf{u}$ are the conservative variables. In general, we can also write Equation (1.1) in terms of primitive variables whose behaviors are either intuitively or physically easier to understand.

$$\frac{\partial \mathbf{q}}{\partial t} + \boldsymbol{A}^q \frac{\partial \mathbf{q}}{\partial x} + \boldsymbol{B}^q \frac{\partial \mathbf{q}}{\partial y} + \boldsymbol{C}^q \frac{\partial \mathbf{q}}{\partial z} = 0 \tag{1.2}$$

Here, $\boldsymbol{A}^q$, $\boldsymbol{B}^q$, $\boldsymbol{C}^q$ are the coefficient matrices of the primitive form of the conservation laws. To solve the same system described by both Equation (1.1) and (1.2), all AF methods follow the same sequence of steps: Reconstruction, Evolution, and Conservation. In the next three sections, we discuss each step in detail for the method in one- and two-dimensions.

1.3.1 Reconstruction

The first step in an AF method is to perform a quadratic reconstruction of primitive variables $\mathbf{q}$ within a cell. By choosing such a reconstruction, we guarantee the scheme is exact for quadratic data and hence achieve third-order spatial accuracy. We will start with the reconstruction stage in one-dimension, and then progress to two-dimensions.

In general, a mapping from physical space $\mathbf{x}$ to reference space ξ is required to simplify

Index	c_i	ϕ_i
1	q_1	$1 - 3\xi + 2\xi^2$
2	q_2	$2\xi^2 - \xi$
3	$\frac{1}{4}(6\overline{q} - q_1 - q_2)$	$-4\xi^2 + 4\xi$

Table 1.1: Coefficients and basis functions for reconstruction in 1-D.

the definition of the solution within a cell. The mapping in one-dimension is simple and illustrated in Figure 1.1(a).

$$\xi = \frac{x - x_1}{x_2 - x_1} = \frac{1}{\Delta x}(x - x_1) \tag{1.3}$$

Three data points are necessary to achieve a quadratic reconstruction in one-dimension, and the points we pick are the two nodes at the interfaces and a node located at the middle of the cell to represent the cell average, which is shown in Figure 1.1(b). As with standard finite element methods, the local solution is represented as a summation of basis functions,

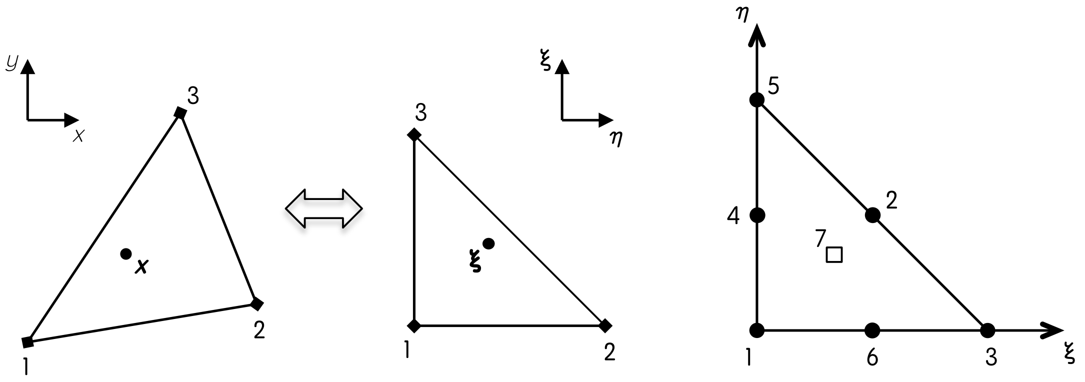
$$q(\xi) = \sum_{i=1}^{3} c_i \phi_i(\xi) \tag{1.4}$$

where q is the primitive variable, and the coefficients c and basis functions $\phi(\xi)$ are listed in Table 1.1. Note that interior node 3 is determined by integrating the average value in the cell.

$$\begin{aligned} \overline{q} &= \frac{1}{\Delta x} \int_{x_1}^{x_2} q(x)\,\mathrm{d}x \\ &= \int_{0}^{1} q(\xi)\,\mathrm{d}\xi \\ &= \sum_{i=1}^{3} c_i \int_{0}^{1} \phi_i(\xi)\,\mathrm{d}\xi \\ &= \frac{1}{6}(q_1 + 4q_3 + q_2) \end{aligned} \tag{1.5}$$

This concludes the reconstruction stage for the one-dimensional AF method.

In two dimensions, assuming counterclockwise ordering of the nodes, we can map the physical space to reference space by using a constant Jacobian matrix. Figure 1.2(a)



(a) Mapping from physical space to reference space.

(b) Node placement for AF basis functions.

Figure 1.2: Geometry and data storage for AF method in two-dimensions.

illustrates the mapping from $\mathbf{x} = (x, y)^T$ to $\boldsymbol{\xi} = (\xi, \eta)^T$. In matrix form,

$$\mathbf{x} = \mathbf{x}_1 + \boldsymbol{J}\boldsymbol{\xi} \tag{1.6}$$

where

$$\boldsymbol{J} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix} \tag{1.7}$$

We can also express the reference space in terms of physical coordinates by inverting the Jacobian matrix, which is useful in situations where the solution must be solved in physical space.

$$\boldsymbol{\xi} = \boldsymbol{J}^{-1}(\mathbf{x} - \mathbf{x}_1) \tag{1.8}$$

Quadratic reconstruction in two-dimensions require only six data points, which are naturally chosen to be the vertex and edge nodes of a cell. However, with these chosen points, there is no guarantee that the average of the resulting reconstruction matches the cell average of the initial and subsequent values obtained using a conservative scheme (see Section 1.3.3). One particularly compact solution is to add a third-order bubble function that does not influence the edge values, which is conveniently located at the centroid. The seven data points required for the full second-order reconstruction are illustrated in Figure 1.2(b), and the coefficients and basis functions are listed in Table 1.2. We can again use the definition of the cell average like we did with the one-dimensional AF method to determine

Index	c_i	ϕ_i
1	q_1	$1 - 3\xi + 2\xi^2 - 3\eta + 4\xi\eta + 2\eta^2$
2	q_2	$4\xi\eta$
3	q_3	$2\xi^2 - \xi$
4	q_4	$4\eta - 4\eta^2 - 4\xi\eta$
5	q_5	$2\eta^2 - \eta$
6	q_6	$4\xi - 4\xi^2 - 4\xi\eta$
7	$\frac{20}{9}\left[\overline{q} - \frac{1}{3}(q_2 + q_4 + q_6)\right]$	$27(\xi\eta - \xi^2\eta - \xi\eta^2)$

Table 1.2: Coefficients and basis functions for reconstruction in 2-D.

the appropriate coefficient for the bubble function.

$$\begin{aligned}
\overline{q} &= \frac{1}{\Omega_j} \iint\limits_{\Omega_j} q(x,y)\,\mathrm{d}\Omega \\
&= \frac{J}{\Omega_j} \int\limits_0^1 \int\limits_0^{1-\xi} q(\xi,\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \\
&= \sum_{i=1}^{7} c_i \int\limits_0^1 \int\limits_0^{1-\xi} \phi_i(\xi,\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \\
&= \frac{1}{3}(q_2 + q_4 + q_6) + \frac{9}{20} c_7
\end{aligned} \tag{1.9}$$

We would like to mention that although the bubble function is our first choice to reconcile the difference between the cell average and the average of the quadratic reconstruction using only the vertex and edge nodes, it is not the only option. Since the six points are sufficient for a quadratic reconstruction, other methods could be developed to obtain the correct cell average by changing the nodal values in such a way as to preserve accuracy.

1.3.2 Evolution

In the evolution stage, the interface update does not need to be conservative. In fact, as long as the method of evolution is consistent, any convenient scheme may be used to generate a point update. This is one of the most powerful aspect of AF because it gives us the flexibility to choose an update method that accurately describes a physical process in multidimensions. For example, the evolution of an acoustic process is isotropic in two- and three-dimensions while advection is directionally-based in any dimension. This idea of using nonconservative update methods opens a wide range of new possibilities for solving conservation laws.

1.3.3 Conservation

The third and final stage of the AF method is to construct average fluxes from the point data by integrating along the edges of a cell using a sufficiently high order approximation such as Simpsons rule. Referring to Figure 1.3, the one-dimensional average flux is simply

$$\bar{\mathbf{f}}_{1D} = \frac{1}{6}\left[\mathbf{f}(\mathbf{u}^n) + 4\mathbf{f}(\mathbf{u}^{n+1/2}) + \mathbf{f}(\mathbf{u}^{n+1})\right] \tag{1.10}$$

where $\mathbf{u}$ are the conservative states calculated from the primitive point values. The two-dimensional average flux can again be approximated by the two-dimensional version of Simpsons rule.

$$\begin{aligned}\bar{\mathbf{f}}_{2D} = \frac{1}{9}\Bigg[&\frac{1}{4}(\mathbf{f}_L^n + \mathbf{f}_R^n + \mathbf{f}_L^{n+1} + \mathbf{f}_R^{n+1}) \\ &+ (\mathbf{f}_M^n + \mathbf{f}_L^{n+1/2} + \mathbf{f}_R^{n+1/2} + \mathbf{f}_M^{n+1}) + 4\mathbf{f}_M^{n+1/2}\Bigg]\end{aligned} \tag{1.11}$$

These fluxes are then used to update the conserved cell-average in a similar fashion as FV methods. Because we integrate around the cell in a closed loop, the leading terms in the solution error cancel and we are left with a third-order accurate cell-average value. In one-dimension, the cell-average is updated by

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^n - \frac{\Delta t}{\Delta x}(\bar{\mathbf{f}}_R - \bar{\mathbf{f}}_L) \tag{1.12}$$

and in two-dimensions,

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^n - \frac{\Delta t}{\Omega_j}\sum_{k=1}^{3}(\bar{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k) l_k \tag{1.13}$$

where Ω_j is the area of the element j, and $\hat{\mathbf{n}}_k$ and l_k are the normal vector and length associated with edge k.

One last detail worth mentioning is the fact that the cell-average data updated using Equations (1.12) and (1.13) depict conservative states while the point data updated in the Evolution stage most likely represent primitive states. For scalar conservation laws, it is common to find that the conserved variables are also the primitive ones, but this is generally not the case for nonlinear systems. The coefficient of the bubble function is expressed

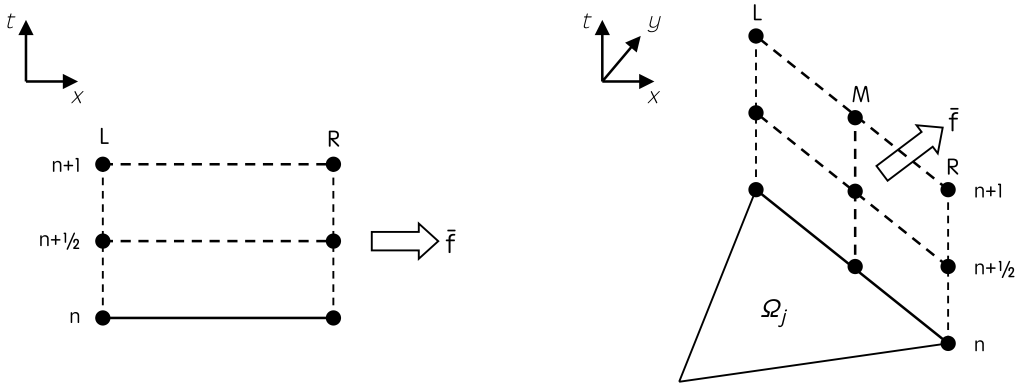


Figure 1.3: Nomenclature for conservation stage of AF method in one-dimension (left) and two-dimensions (right).

using primitive cell-average, which should be computed carefully from the conservative cell-average as to preserve the order of accuracy.

1.4 Thesis overview

This dissertation focuses on the development of the Active Flux method for conservation laws that either fully represent or contain acoustic processes. It also includes discussions of various boundary conditions for the AF method that are scattered throughout the chapters. Chapter 2 introduces the linear acoustic equations and the nonconservative update method used in the Evolution stage of the AF scheme, as well as provides numerical results supporting the advantages of the method. Nonlinear extensions are discussed in Chapter 3 along with open boundary conditions. Chapter 4 discusses and shows how the correct form of operator splitting on the full Euler equations can result in an AF scheme that accurately represents the physical behaviour of advective and acoustic processes in inviscid flows. Then, boundary conditions are formulated and implemented for the Euler system in Chapter 5. We conclude the dissertation with suggestions for future research topics in Chapter 6.

CHAPTER 2

Linear Acoustics

We will start our understanding of acoustic systems by analyzing the linear acoustics equations derived by taking only the acoustic terms from the Euler equations. Linearizing the acoustic subsystem by assuming constant coefficients ρ_0 and c_0, and relating the pressure and density through the isentropic assumption $c^2 = p/\rho$, the acoustic portion of the Euler equations can be written as:

$$\frac{\partial p^*}{\partial t} + c_0 \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} + \frac{\partial w^*}{\partial z} \right) = 0 \tag{2.1a}$$

$$\frac{\partial u^*}{\partial t} + c_0 \frac{\partial p^*}{\partial x} = 0 \tag{2.1b}$$

$$\frac{\partial v^*}{\partial t} + c_0 \frac{\partial p^*}{\partial y} = 0 \tag{2.1c}$$

$$\frac{\partial w^*}{\partial t} + c_0 \frac{\partial p^*}{\partial z} = 0 \tag{2.1d}$$

where the non-dimensional pressure $p^* = p/\rho_0 c_0^2$ and velocity $\mathbf{u}^* = \mathbf{u}/c_0$ are used for visual cleaniness.

Taking the second time-derivative of Equation (2.1), it becomes obvious that both pressure and velocity obey a form of the wave equation.

$$\frac{\partial^2 p^*}{\partial t^2} - c_0^2 \nabla^2 p^* = 0 \tag{2.2}$$

$$\frac{\partial^2 \mathbf{u}^*}{\partial t^2} - c_0^2 \nabla^2 \mathbf{u}^* = c_0^2 (\nabla \times \nabla \times \mathbf{u}^*) \tag{2.3}$$

We have used the vector identity $\nabla(\nabla \cdot \mathbf{u}^*) = \nabla^2 \mathbf{u}^* + \nabla \times \nabla \times \mathbf{u}^*$ to show that the velocity obeys the wave equation with an additional term that is dependent on vorticity $\omega = \nabla \times \mathbf{u}^*$. For this subset of problems, we can find an exact solution to update the pressure and velocity coefficients of the quadratic basis functions used in the Active Flux

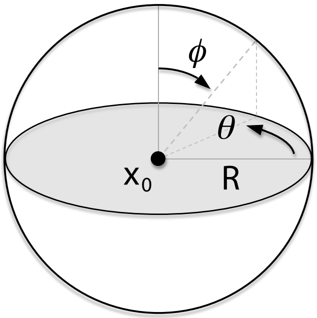


Figure 2.1: Sphere of influence about point $\mathbf{x}_0$ with radius R.

method, which we then use to construct an average flux at each interface as previously outlined in Section 1.3. The details for finding an analytical solution to Equation (2.1) are discussed in Section 2.1. The rest of the chapter will explore the performance of the AF scheme applied to various linear acoustic problems in terms of accuracy, stability, and robustness. Finally, simple numerical conditions are applied to the acoustics solver at flat wall boundary.

2.1 Exact solutions using spherical means

2.1.1 Spherical means integral

The spherical means of a function f at an arbitrary point (x_0, y_0, z_0) is defined as the average value over the surface of a sphere of radius R.

$$M_R^{3D}\{f\}(x_0,y_0,z_0) = \frac{1}{4\pi R^2}\int_0^{2\pi}\int_0^{\pi} f(x_0 + R\sin\phi\cos\theta, y_0 + R\sin\phi\sin\theta, z_0 + R\cos\phi)R^2\sin\phi\,\mathrm{d}\phi\,\mathrm{d}\theta \tag{2.4}$$

The sphere of influence in three-dimensions is shown in Figure 2.1. Using the method of descent, we can treat the two-dimensional spherical means as a special case where there is no dependence on the z-coordinate. The sphere of influence then becomes a disc of influence where the function is now averaged over the area of a disc.

$$M_R^{2D}\{f\}(x_0,y_0) = \frac{1}{2\pi R}\int_0^{2\pi}\int_0^{R} f(x_0 + r\cos\theta, y_0 + r\sin\theta)\frac{r}{\sqrt{R^2 - r^2}}\,\mathrm{d}r\,\mathrm{d}\theta \tag{2.5}$$

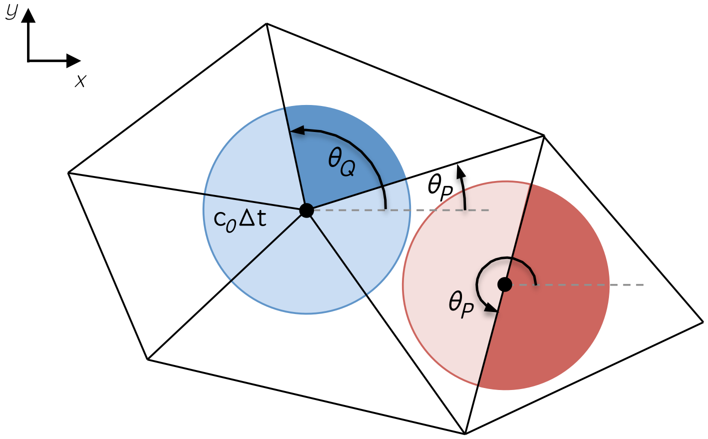


Figure 2.2: Integration of spherical means with radius $R = c_0\Delta t$ at vertex (blue) and edge (red) nodes inside neighboring triangular elements.

Note that the disc and sphere of radius R is also the domain of dependence in two- and three-dimensions.

In the Active Flux framework, the two-dimensional spherical means are integrated over discs of radius $R = c_0\Delta t$ centered at the vertex and edge nodes, as illustrated in Figure 2.2. Partial integration of the disc is performed in each element separately, and the angular limits of integration are set by the intersection of the integral disc with the element edges. At the vertex nodes, we define both a starting angle θ_P and a final angle θ_Q with the constraint $\theta_Q > \theta_P$, whereas we only require θ_P for edge nodes since $\theta_Q = \theta_P + \pi$ is always implied due to geometry.

As previously mentioned in Section 1.3, the AF solution is approximated by quadratic reconstruction functions within each triangular element. Thus, the spherical means integral of solution state u at node position (x_n, y_n) becomes:

$$\begin{aligned} M_R\{u\}(x_n, y_n) &= \frac{1}{2\pi R}\int_{\theta_P}^{\theta_Q}\int_0^R u(x_n + r\cos\theta, y_n + r\sin\theta)\frac{r}{\sqrt{R^2 - r^2}}\,dr\,d\theta \\ &= \frac{1}{2\pi R}\sum_{i=0}^{6} u_i \int_{\theta_P}^{\theta_Q}\int_0^R \phi_i(\,\xi(x_n + r\cos\theta, y_n + r\sin\theta), \\ &\qquad\qquad \eta(x_n + r\cos\theta, y_n + r\sin\theta)\,)\frac{r}{\sqrt{R^2 - r^2}}\,dr\,d\theta \end{aligned} \tag{2.6}$$

The above integral may seem daunting at first glance. However, considering that the basis functions $\phi_i(\xi, \eta)$ are quadratic functions expressed in reference coordinates that correlate linearly to the spatial coordinates $x = r\cos\theta$ and $y = r\sin\theta$ with respect to the Jacobian matrix, we would only need to integrate at most quadratic expressions.

For example, to integrate functions that use only powers of x, the integrals that have to be computed are all of the form

$$I_p = \int_{\theta_P}^{\theta_Q} \int_0^R \frac{r^{p+1} cos^p \theta}{\sqrt{R^2 - r^2}} \,\mathrm{d}r\,\mathrm{d}\theta \tag{2.7}$$

where p takes all integer values less than or equal to the order of the function. It should also be noted that with the use of a bubble function, cubic functions will need to be integrated in addition to quadratic. Therefore, the integrals we will need are as follows.

$$\begin{aligned} I_0 &= R(\theta_Q - \theta_P) \\ I_1 &= \frac{\pi R^2}{4}(\sin\theta_Q - \sin\theta_P) \\ I_2 &= \frac{R^3}{3}(\theta_Q - \theta_P + \cos\theta_Q \sin\theta_Q - \cos\theta_P \sin\theta_P) \\ I_3 &= \frac{\pi R^4}{16}(\sin\theta_Q(3 - \sin^2\theta_Q) - \sin\theta_P(3 - \sin^2\theta_P)) \end{aligned} \tag{2.8}$$

Further simplifications can be made to the integrals in Equation (2.6) by using Laplacian point evaluations in addition to some correction terms (see Appendix A for more details). The two-dimensional spherical means integral can also be evaluated using area coordinates instead of reference coordinates, as derived by Eymann in [30].

2.1.2 Application of spherical means to linear acoustics equations

Let the state vector be $\mathbf{q} = (p^*, u^*, v^*, w^*)^T$. Going back to the linear acoustics equations, we can rewrite Equation (2.1) in its matrix form as

$$\frac{\partial \mathbf{q}}{\partial t} + c_0 \boldsymbol{L}\mathbf{q} = \mathbf{0} \tag{2.9}$$

where

$$\boldsymbol{L} = \begin{pmatrix} 0 & \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ \dfrac{\partial}{\partial x} & 0 & 0 & 0 \\ \dfrac{\partial}{\partial y} & 0 & 0 & 0 \\ \dfrac{\partial}{\partial z} & 0 & 0 & 0 \end{pmatrix} \tag{2.10}$$

Equation (2.9) has a formal solution

$$\begin{aligned} \mathbf{q}(\mathbf{x}, t) &= e^{-c_0 \boldsymbol{L} t}\mathbf{q}(\mathbf{x}, 0) \\ &= \sum_{p=0}^{\infty} \frac{(-c_0 \boldsymbol{L} t)^p}{p!}\mathbf{q}(\mathbf{x}, 0) \end{aligned} \tag{2.11}$$

An interesting and useful property of the gradient matrix $\boldsymbol{L}$ is that it satisfies

$$\boldsymbol{L}^{p+2} = \nabla^2 \boldsymbol{L}^p \qquad (p > 0)$$

where ∇^2 is the Laplacian operator in three-dimensional space. Note that $\boldsymbol{L}^2 \neq \nabla^2 \mathbf{I}$, and in fact

$$\boldsymbol{L}^2 = \begin{pmatrix} \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} + \dfrac{\partial^2}{\partial z^2} & 0 & 0 & 0 \\ 0 & \dfrac{\partial^2}{\partial x^2} & \dfrac{\partial^2}{\partial x \partial y} & \dfrac{\partial^2}{\partial x \partial z} \\ 0 & \dfrac{\partial^2}{\partial x \partial y} & \dfrac{\partial^2}{\partial y^2} & \dfrac{\partial^2}{\partial y \partial z} \\ 0 & \dfrac{\partial^2}{\partial x \partial z} & \dfrac{\partial^2}{\partial y \partial z} & \dfrac{\partial^2}{\partial z^2} \end{pmatrix} \tag{2.12}$$

Using this special property of $\boldsymbol{L}$, we can manipulate Equation (2.11) into the summation of even and odd terms.

$$\mathbf{q}(t) = \mathbf{q}(0) - \sum_{p=1,3,5} \frac{(c_0 \boldsymbol{L} t)^p}{p!}\mathbf{q}(0) + \sum_{p=2,4,6} \frac{(c_0 \boldsymbol{L} t)^p}{p!}\mathbf{q}(0)$$

$$= \mathbf{q}(0) - c_0 \boldsymbol{L} t \left\{ \sum_{p=1,3,5} \frac{(c_0 \boldsymbol{L} t)^{p-1}}{p!} \right\} \mathbf{q}(0) + (c_0 \boldsymbol{L} t)^2 \left\{ \sum_{p=2,4,6} \frac{(c_0 \boldsymbol{L} t)^{p-2}}{p!} \right\} \mathbf{q}(0)$$

$$= \mathbf{q}(0) - c_0 \boldsymbol{L} t \left\{ \sum_{p=1,3,5} \frac{(c_0 t)^{p-1}}{p!} \nabla^{p-1} \right\} \mathbf{q}(0) + (c_0 \boldsymbol{L} t)^2 \left\{ \sum_{p=2,4,6} \frac{(c_0 t)^{p-2}}{p!} \nabla^{p-2} \right\} \mathbf{q}(0)$$

$$= \mathbf{q}(0) - t \left\{ \sum_{p=0}^{\infty} \frac{(c_0 t)^{2p}}{(2p+1)!} \nabla^{2p} \right\} c_0 \boldsymbol{L} \mathbf{q}(0) + t^2 \left\{ \sum_{p=0}^{\infty} \frac{(c_0 t)^{2p}}{(2p+2)!} \nabla^{2p} \right\} c_0^2 \boldsymbol{L}^2 \mathbf{q}(0)$$

$$= \mathbf{q}(0) + t \left\{ \sum_{p=0}^{\infty} \frac{R^{2p}}{(2p+1)!} \nabla^{2p} \right\} \partial_t \mathbf{q}(0) + t^2 \left\{ \sum_{p=0}^{\infty} \frac{R^{2p}}{(2p+2)!} \nabla^{2p} \right\} \partial_{tt} \mathbf{q}(0) \quad (2.13)$$

The radius of the spherical means integral is $R = c_0 t$ and we have used the fact that the linear acoustics equations obey the time-derivatives $\partial_t \mathbf{q} = -c_0 \boldsymbol{L} \mathbf{q}$ and $\partial_{tt} \mathbf{q} = c_0^2 \boldsymbol{L}^2 \mathbf{q}$.

The formal solution was expanded in the above form because we noticed that the spherical means formula can be written as the sum of point evaluations of increasing powers of the Laplacian,

$$M_R = \sum_{p=0}^{\infty} \frac{R^{2p}}{(2p+1)!} \nabla^{2p} \quad (2.14)$$

which corresponds to the odd terms in Equation (2.13). This follows from Darboux's equation that the following expression holds true for any function $f(\mathbf{x})$,

$$\partial_{rr} M_r\{f\}(\mathbf{x}) + \frac{2}{r} \partial_r M_r\{f\}(\mathbf{x}) = M_r\{\nabla^2 f\}(\mathbf{x}) \quad (2.15)$$

which has the formal solution

$$M_r\{f\}(\mathbf{x}) = \frac{\sinh(\nabla r)}{\nabla r} f(\mathbf{x}) \quad (2.16)$$

We can then substitute the expression directly into the following integral to get a term corresponding to the even terms in Equation (2.13).

$$\begin{aligned} \frac{2}{R^2} \int_0^R r M_r\{f\}(\mathbf{x}) \, \mathrm{d}r &= \frac{2}{R} \int_0^R \frac{r \sinh(\nabla r)}{\nabla r} f(\mathbf{x}) \, \mathrm{d}r \\ &= \frac{2 \cosh \nabla R - 1}{\nabla^2 R^2} f(\mathbf{x}) \\ &= 2 \sum_{p=0}^{\infty} \frac{R^{2p}}{(2p+2)!} \nabla^{2p} f(\mathbf{x}) \end{aligned} \quad (2.17)$$

Hence, the formal solution to the linear acoustics system can be expressed using Equations (2.14) and (2.17) as

$$\mathbf{q}(t) = \mathbf{q}(0) + tM_R\{\partial_t \mathbf{q}\}(0) + \frac{1}{c_0^2}\int\limits_0^R rM_r\{\partial_{tt}\mathbf{q}\}(0)\,\mathrm{d}r \tag{2.18}$$

and applies to all four components of $\mathbf{q}$.

From a numerical analysis point of view, Equation (2.18) is quite interesting. The first term is just the current solution, while the second and third terms are of order t and t^2, respectively. This numerical form strongly resembles the Lax-Wendroff formula. In fact, if we evaluate each term pointwise, we would recover the Lax-Wendroff finite difference approximation.

$$\begin{aligned}\mathbf{q}(t) &= \mathbf{q}(0) + t\partial_t\mathbf{q}(0) + \frac{1}{c_0^2}\int\limits_0^{c_0 t} r\,\mathrm{d}r\;\partial_{tt}\mathbf{q}(0)\\ &= \mathbf{q}(0) + t\partial_t\mathbf{q}(0) + \frac{t^2}{2}\partial_{tt}\mathbf{q}(0)\end{aligned} \tag{2.19}$$

However, the formula becomes exact if the terms are evaluated exactly using the spherical means integrals in Equation (2.18).

For the first component p^*, some simplification is possible since $\partial_{tt}p^* = c_0^2\nabla^2 p^*$.

$$p^*(t) = p(0) + tM_R\{\partial_t p^*\}(0) + \int\limits_0^R rM_r\{\nabla^2 p^*\}(0)\,\mathrm{d}r \tag{2.20}$$

Combining the first and third term gives us the well-known exact solution for the scalar wave equation from textbooks on partial differential equations [31, 32]:

$$p^*(t) = \partial_t(tM_R\{p^*\}) + tM_R\{\partial_t p^*\} \tag{2.21}$$

An alternate form of the above expression can be achieved by expanding the first term, replacing the time derivative with a derivative with respect to R, and substituting the time derivative with spatial derivatives from Equation (2.1a) , as derived by Eymann [30].

$$\begin{aligned}p^*(t) &= M_R\{p^*\} + R\partial_R M_R\{p^*\} + tM_R\{\partial_t p^*\}\\ &= M_R\{p^*\} + R\partial_R M_R\{p^*\} - RM_R\{\nabla\cdot\mathbf{u}^*\}\end{aligned} \tag{2.22}$$

The same simplification process can be applied to the velocity components of $\mathbf{q}$ by using

$\partial_{tt}\mathbf{u}^* = c_0^2\nabla^2\mathbf{u}^* + c_0^2\nabla\times\nabla\times\mathbf{u}^*$, resulting in

$$\mathbf{u}^*(t) = M_R\{\mathbf{u}^*\} + R\partial_R M_R\{\mathbf{u}^*\} - RM_R\{\nabla p^*\} + \int\limits_0^R rM_r\{\nabla\times\nabla\times\mathbf{u}^*\}\,\mathrm{d}r \qquad (2.23)$$

2.1.3 Vorticity term

As we can clearly see from Equation (2.23), the velocity solution contains an additional term dependent on the curl of the flow vorticity that does not exist in the pressure solution. At this time, we would like to prove that the spherical means integral of this vorticity term in fact vanishes on a sphere with an arbitrary radius r.

Claim: The integral term vanishes

$$\int\limits_0^R rM_r\{\nabla\times\nabla\times\mathbf{v}\}\,\mathrm{d}r = 0$$

for arbitrary velocity field $\mathbf{v}$.

Proof: Consider the volume integral

$$\mathbf{V} = \iiint\limits_{\Omega} \nabla\times\nabla\times\mathbf{v}\,\mathrm{d}V$$

where $\mathbf{v}(\mathbf{x})$ is an arbitrary velocity field. First, we apply the generalized form of Stokes' theorem to get

$$\mathbf{V} = \iiint\limits_{\Omega} \nabla\times(\nabla\times\mathbf{v})\,\mathrm{d}V = \oiint\limits_{\partial\Omega} \hat{\mathbf{n}}\times(\nabla\times\mathbf{v})\,\mathrm{d}S$$

Then, applying classical Stokes' theorem to the above expression, it can be shown that the flux integral of a curl field over a closed surface is zero because a closed surface has no boundary. Thus,

$$\mathbf{V} = -\oiint\limits_{\partial\Omega} \nabla\times(\mathbf{v}\times\hat{\mathbf{n}})\,\mathrm{d}S = -\int\limits_0 (\mathbf{v}\times\hat{\mathbf{n}})\,\mathrm{d}s = \mathbf{0}$$

Now, the radial change in the volume integral is related to the spherical means integral as follows

$$\partial_r\mathbf{V} = 4\pi r^2 M_r\{\nabla\times\nabla\times\mathbf{v}\} = \mathbf{0}$$

from which it follows that $M_r\{\nabla \times \nabla \times \mathbf{v}\} = \mathbf{0}$ and the integral under consideration vanishes also.

■

Hence, the vorticity of the flow field is preserved by the velocity solution in Equation (2.23) after disregarding the last term. To summarize, the analytical solution to the linear acoustics system described by Equation (2.9) is

$$\mathbf{q}(\mathbf{x},t) = M_R\{\mathbf{q}\}(\mathbf{x},0) + R\partial_R M_R\{\mathbf{q}\}(\mathbf{x},0) - RM_R\{\boldsymbol{L}\mathbf{q}\}(\mathbf{x},0) \tag{2.24}$$

The spherical means integrals in the above equation are evaluated at each node by calculating and adding together the partial integrals from every neighboring element, as described previously in Section 2.1.1.

2.2 Evaluation of one-dimensional data

One of the perceived strengths of using spherical means to calculate the solution to the linear acoustics equations is that the spherical means integral naturally satisfies one-, two-, and three-dimensional physics without additional mathematical considerations. In one dimension, the spherical means integral is very simple:

$$M_R\{f\}(x) = \frac{1}{2R}\int\limits_{-R}^{R} f(x)\,\mathrm{d}x \tag{2.25}$$

Subsequently, in one-dimensional flow, the method of spherical means in fact collapses to the method of characteristics [31].

$$p^*(x,t) = \frac{1}{2}\left(p^*(R,0) + p^*(-R,0)\right) - \frac{R}{2}\left(u^*(R,0) - u^*(-R,0)\right) \tag{2.26a}$$

$$u^*(x,t) = \frac{1}{2}\left(u^*(R,0) + u^*(-R,0)\right) - \frac{R}{2}\left(p^*(R,0) - p^*(-R,0)\right) \tag{2.26b}$$

To demonstrate and verify this assertion, we will test two different sets of one-dimensional data using the one-dimensional upwind method of characteristics proposed by Eymann in [30] and the two-dimensional spherical means method described by Equation (2.24). These two methods will be referred to as the 1-D and 2-D Active Flux schemes, respectively.

2.2.1 Nonsimple wave

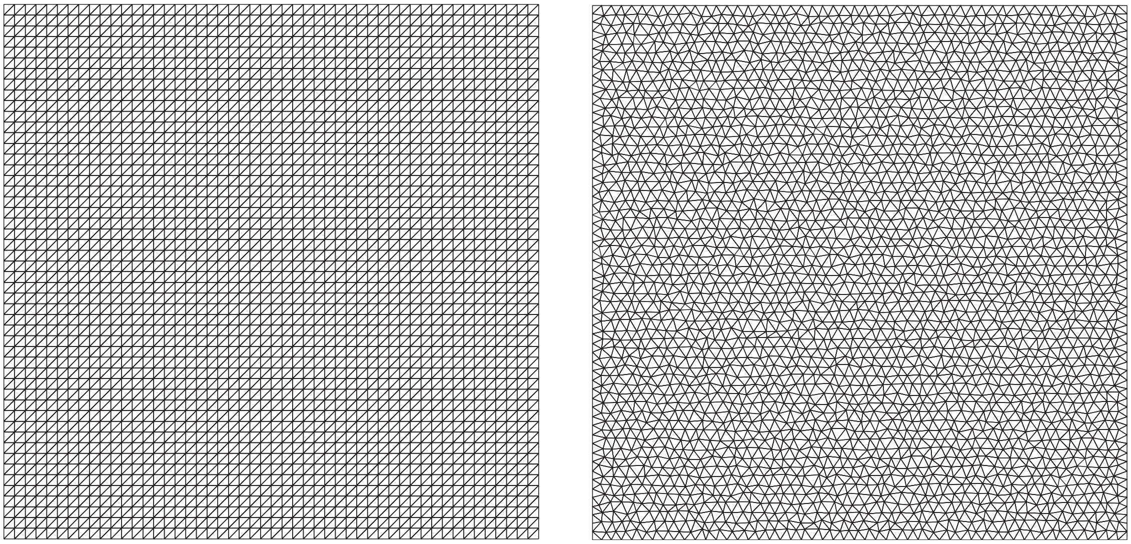
We used the nonsimple wave problem from [30] to assess the difference, if any, between the results from the 1-D and 2-D Active Flux solvers on smooth data. The initial conditions are

$$\begin{aligned} p(x,0) &= \frac{1}{4} + \frac{1}{80}\sin(2\pi x) \\ u(x,0) &= \frac{1}{4} - \frac{1}{10}\sin(\pi x) \\ c_0 &= 1 \\ \rho_0 &= 1/2 \end{aligned} \tag{2.27}$$

and the exact solution described by Equation (2.28) has a simple form due to the linear nature of the problem.

$$\begin{aligned} p(x,t) &= \frac{1}{2}\left[p(x-c_0t,0) + u(x-c_0t,0) + p(x+c_0t,0) - u(x+c_0t,0)\right] \\ u(x,t) &= \frac{1}{2}\left[p(x-c_0t,0) + u(x-c_0t,0) - p(x+c_0t,0) + u(x+c_0t,0)\right] \end{aligned} \tag{2.28}$$

Figure 2.4 compares the exact, one-dimensional numerical, and two-dimensional numerical solutions at time $t = 6$ on the structured and unstructured grids illustrated in Figure 2.3 with 50 elements along the x-axis. All of the results generated on the two-dimensional grids have been plotted to show the one-dimensional quality of the 2-D solution. In partic-



(a) Structured grid.

(b) Unstructured grid.

Figure 2.3: Grids used to evaluate one-dimensional problems.

ular, the solution on the unstructured grid follows the exact solution so closely that it forms a line of its own. We observe that, as expected, the 2-D AF solution computed on the structured grid is almost indistinguishable from the 1-D solution The deviations between the two solutions range from $2.46 \times 10^{-5}\%$ to $8.47 \times 10^{-4}\%$. In addition, all three computed results nearly match the exact solution, with the 2-D AF solution from the unstructured grid having a maximum error of 0.0023%. These observations still hold true later into the simulation at time $t = 11.4$ with deviations between the two methods ranging from $3.73 \times 10^{-6}\%$ to 0.0022%, as shown in Figure 2.5.

2.2.2 Square wave

After verifying the two AF solvers are able to produce almost identical results for smooth data, we move onto comparing their solutions for a discontinuous problem. The initial

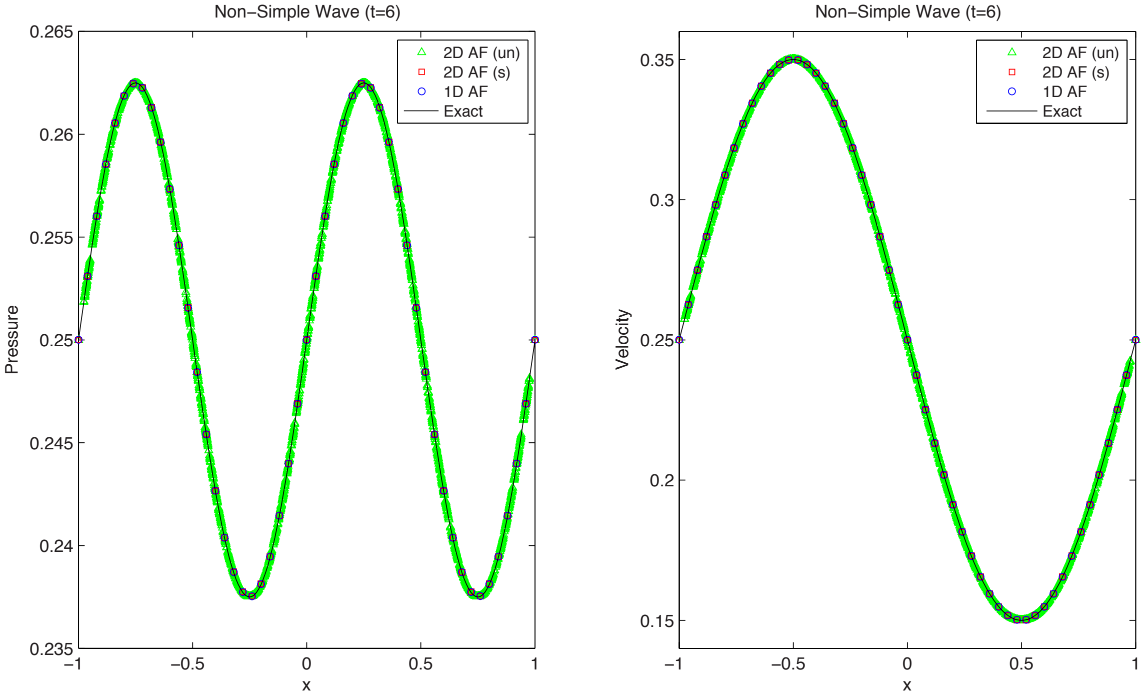


Figure 2.4: Pressure and velocity solutions for nonsimple wave problem; $h = 0.04$, $t = 6$.

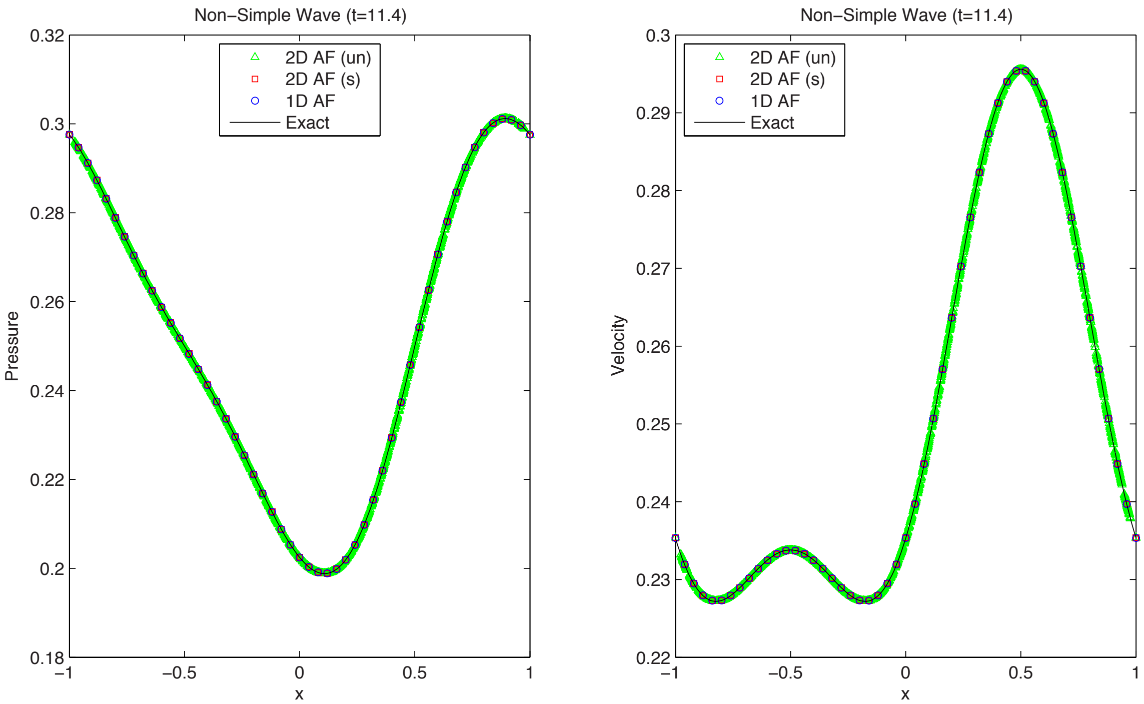


Figure 2.5: Pressure and velocity solutions for nonsimple wave problem; $h = 0.04$, $t = 11.4$.

conditions for a simple square wave problem is

$$\begin{aligned} p(x,0) &= \begin{cases} 1/2 & \text{if } -1 \leq x \leq 1 \\ 0 & \text{elsewhere} \end{cases} \\ u(x,0) &= 0 \\ c_0 &= 1 \\ \rho_0 &= 1/2 \end{aligned} \tag{2.29}$$

where the domain is $x \in [-5, 5]$, and a uniform structured grid with 100 elements along the x-axis is used. Again, the exact solution for this problem follows the same simple form as Equation (2.28).

Figure 2.6 compares the exact solution and computed results with no limiting at time $t = 3$. The square wave has split apart into two subwaves that travel in opposite directions until they have almost reached the boundaries. The 1-D and 2-D AF results are again almost indistinguishable in the smooth regions of the solution. In regions close to the discontinuities, the two computed results differ only slightly in the maximum and minimum overshoots, with the maximum difference of 2.3% occurring at the shock fronts. Both 1-

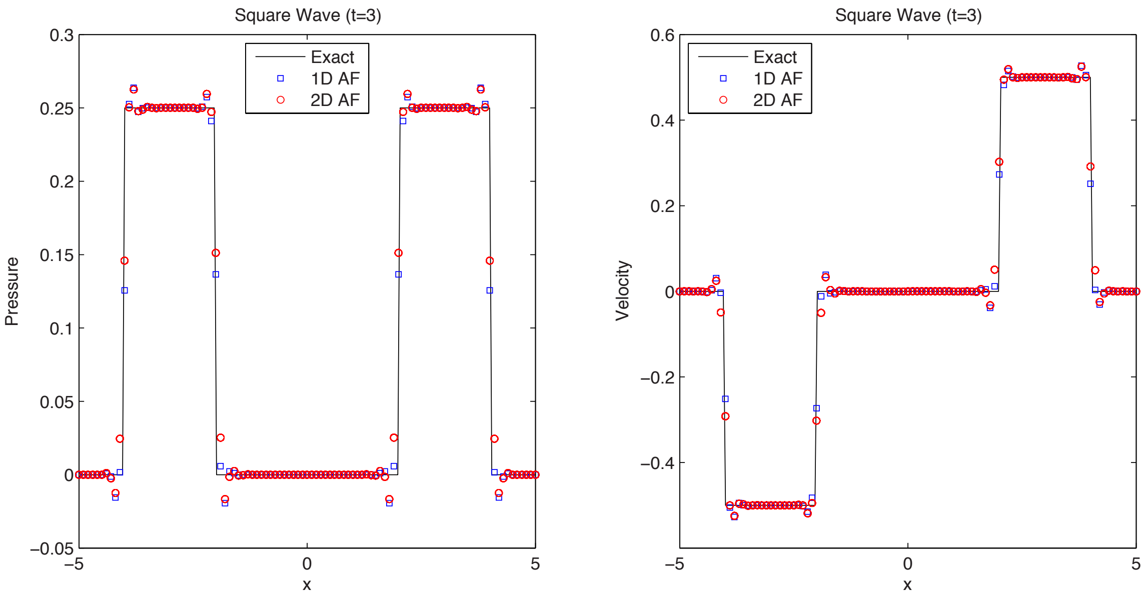


Figure 2.6: Pressure and velocity solutions for square wave problem; $h = 0.1$, $t = 3$.

D and 2-D results approximate the exact solution well given that no limiting has been enforced, with maximum overshoots of 12.6% and 14.6%, respectively.

2.3 Convergence and accuracy

We can confirm and compare the accuracy and order of convergence of the Active Flux method with the Discontinuous Galerkin method using linear reconstruction (DG1) by using the initial conditions proposed by Lukáčová-Medivid'ová, Morton, and Warnecke [33]:

$$\begin{aligned} p(x,y,0) &= \frac{1}{c_0}\left[\sin(2\pi x) + \sin(2\pi y)\right] \\ u(x,y,0) &= 0 \\ v(x,y,0) &= 0 \end{aligned} \tag{2.30}$$

This double sine waves problem has an analytical solution of

$$\begin{aligned} p(x,y,t) &= \frac{1}{c_0}\cos(2\pi c_0 t)\left[\sin(2\pi x) + \sin(2\pi y)\right] \\ u(x,y,t) &= \frac{1}{c_0}\sin(2\pi c_0 t)\cos(2\pi x) \\ v(x,y,t) &= \frac{1}{c_0}\sin(2\pi c_0 t)\cos(2\pi y) \end{aligned} \tag{2.31}$$

For our test, the sound speed is set to $c_0 = 1$ and a final time $t = 1.0$, by which time the original data should repeat itself. The Courant number is set to the maximally stable value of 1.0.

The state errors are computed based on area integrals,

$$L_1(\mathbf{q}) = \frac{1}{|\Omega|}\int_\Omega |\mathbf{q}(t) - \mathbf{q}_{\text{ex}}(t)|\,\mathrm{d}\Omega \tag{2.32}$$

$$L_2(\mathbf{q}) = \left[\frac{1}{|\Omega|}\int_\Omega |\mathbf{q}(t) - \mathbf{q}_{\text{ex}}(t)|^2\,\mathrm{d}\Omega\right]^{1/2} \tag{2.33}$$

where $\mathbf{q} = (p, u, v)$ and $\mathbf{q}_{\text{ex}}$ are the exact solutions of the state variables as described in Equation (2.31). Figures 2.7 and 2.8 both show that the Active Flux method converges at third-order for all state variables, whereas DG1 converges in general at only second-order, with the exception of cell-averaged pressure in Figure 2.8, which displays superconvergence for this linear problem. The AF errors are also lower than those of DG1; most notably, the velocity errors of the AF method is more than two orders of magnitude smaller than those of DG1.

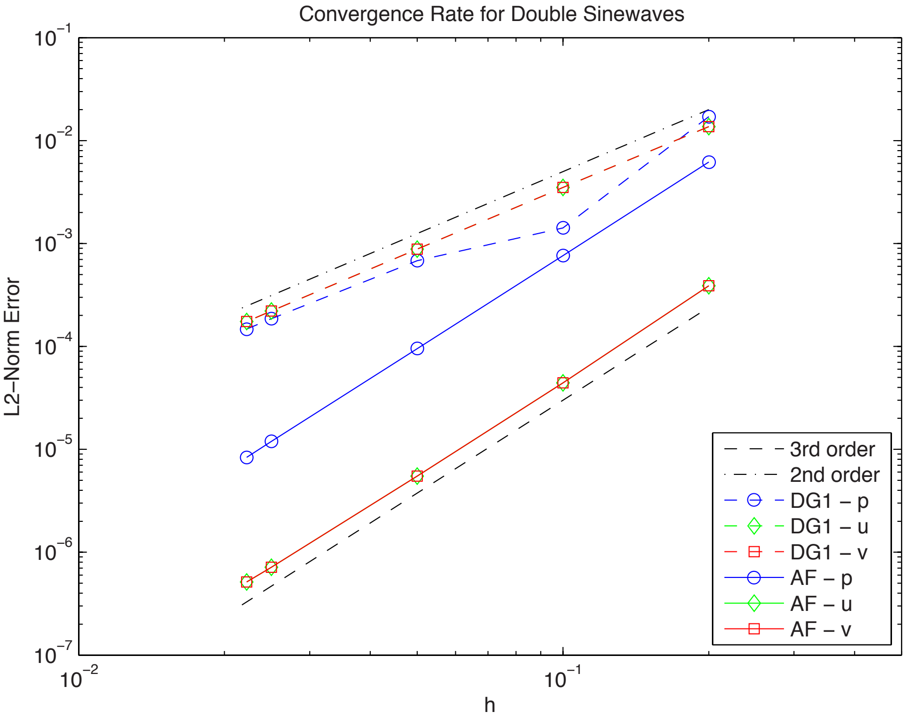


Figure 2.7: Convergence plot of AF and DG1 state errors for double sine waves problem.

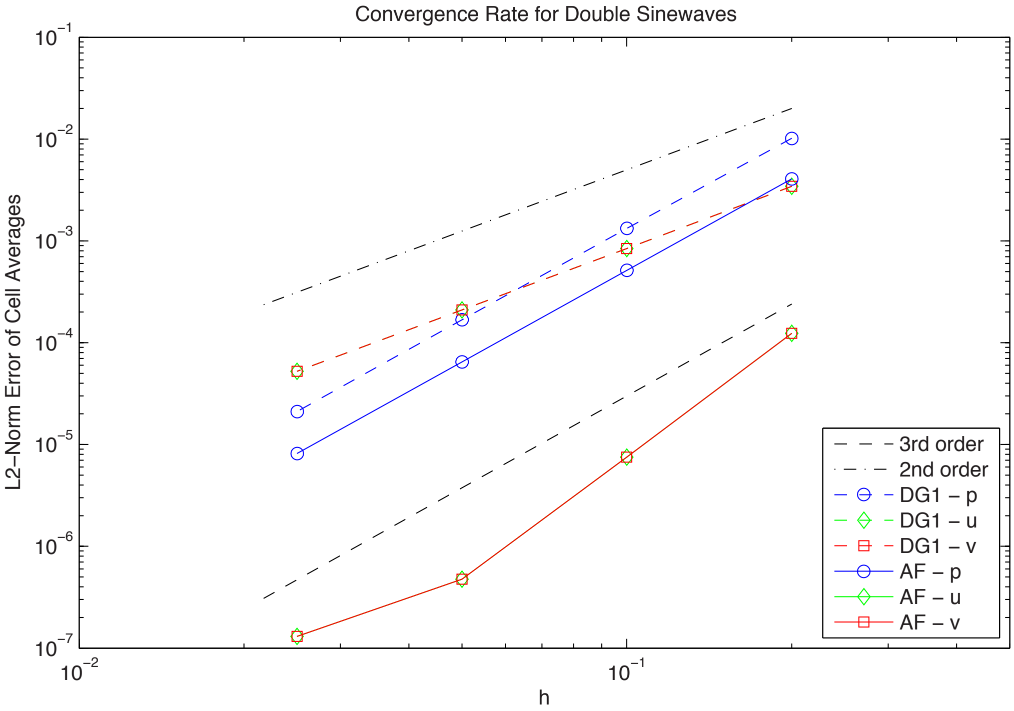


Figure 2.8: Convergence plot of AF and DG1 cell-average state errors for double sine waves problem.

2.3.1 Mesh robustness

In the construction of the Active Flux scheme, we are striving to build a method that is based on the physics of multidimensional flow. In acoustic problems, the physics of the flow indicates that the method should take into account flow properties from all spatial directions surrounding a point of interest. Unlike conventional methods that generalize multidimensional flows using only one-dimensional results, we are hoping that by mimicking the physical properties of acoustic flow using the method of spherical means, we can achieve accurate solutions regardless of the quality of mesh in multidimensions.

To test the ability of the AF method to operate on a severely distorted "torture" grid, we used the square grid $(x, y) \in [-1, 1]$ shown in Figure 2.9. The grid is composed of structured elements everywhere except in the North-East quadrant, which was deliberately polluted with small, badly formed elements with high aspect ratios up to $\mathcal{AR}_{\max} \approx 288$. We will again use the double sine waves problem described by Equation (2.31) to compare the solutions obtained using the AF scheme and the DG1 method which uses one-dimensional Riemann fluxes along cell edges.

Figure 2.10 shows that the AF method gave better results on the pressure predictions with errors that are approximately two-thirds of DG1. There was a much larger difference

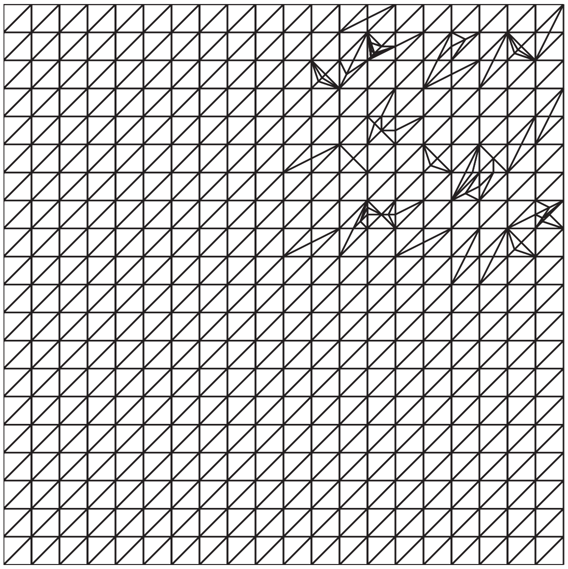
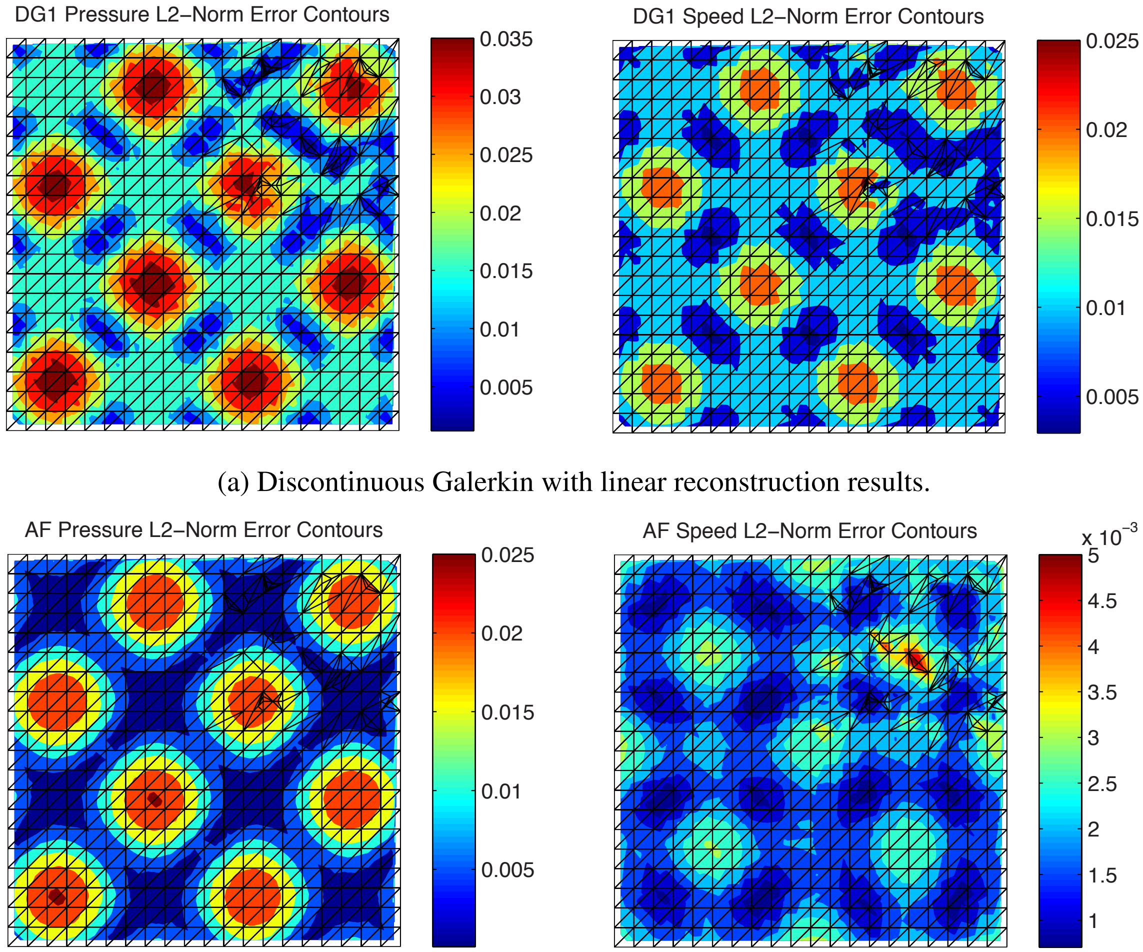


Figure 2.9: Grid with severely distorted elements in the North-East quadrant; used to test tolerance of poor meshes.

in the prediction of speed, where the AF errors were less by a factor of 20. To offer a more fair comparison to the DG1 method, we have also plotted the cell-averaged solutions in Figure 2.11, which shows that when integrated over each individual element, the AF and DG1 method gave similar pressure results. However, the AF errors in speed were still less by a factor of 4 compared to DG1.

2.3.2 Vorticity preservation

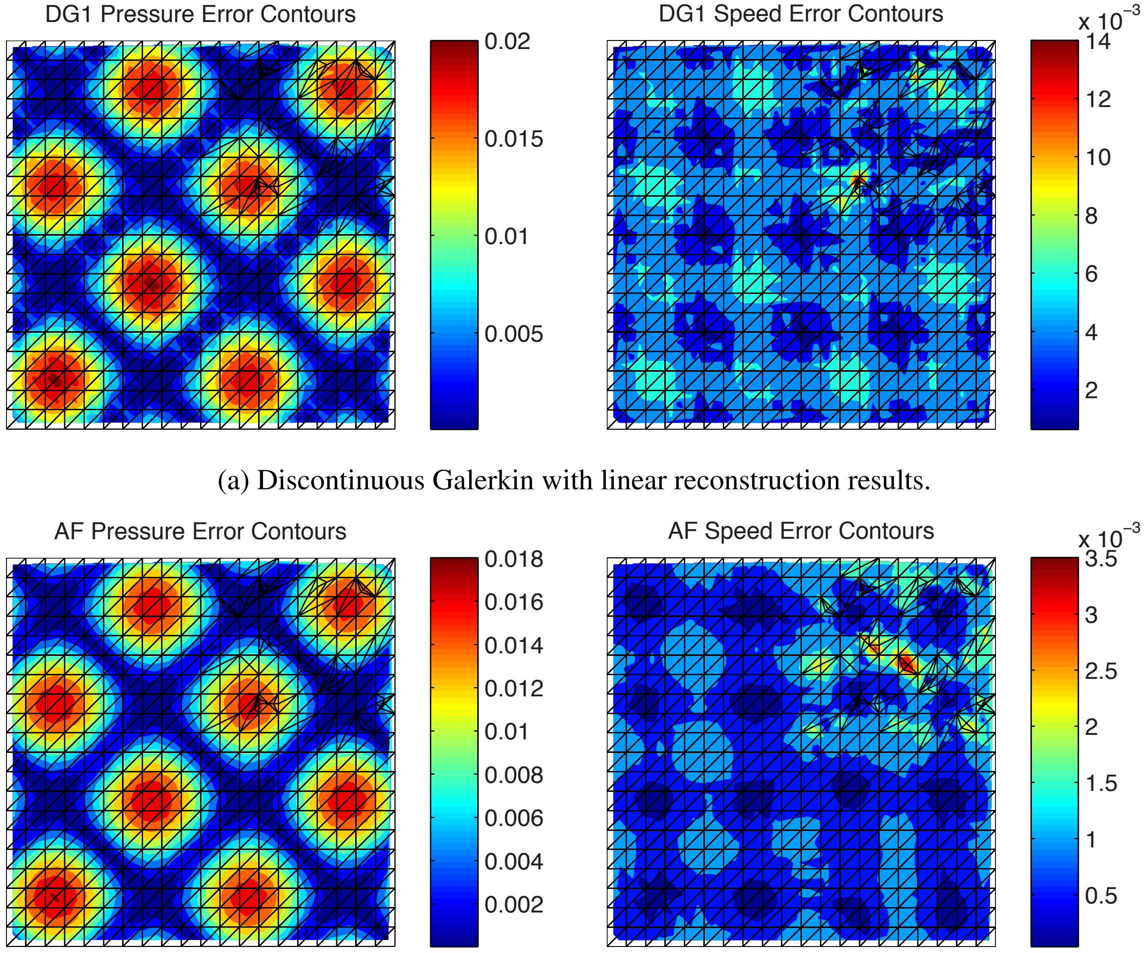
Another advantage of incorporating multidimensional physics into the AF method is that the vorticity of the flow field is naturally preserved. Any irrotational velocities should remain vorticity-free, while rotational velocities will not change in time given that no additional vorticity is created within the flow field. The same problem described by Equation (2.31) is tested with the addition of a stationary vortex centered at the origin on a square



(a) Discontinuous Galerkin with linear reconstruction results.

(b) Active Flux results.

Figure 2.10: Contours of pressure and speed errors for the double sine waves problem computed on "torture" grid; $t = 1.0$.



DG1 Pressure Error Contours

DG1 Speed Error Contours

(a) Discontinuous Galerkin with linear reconstruction results.

AF Pressure Error Contours

AF Speed Error Contours

(b) Active Flux results.

Figure 2.11: Contours of cell-average pressure and speed errors for the double sine waves problem computed on "torture" grid; $t = 1.0$.

grid $(x, y) \in [-2, 2]$:

$$u(x,y,0) = \begin{cases} 1.6yr(4r-3) & \text{if } r < 0.5 \\ -0.8y(r-1.5)^2 & \text{if } 0.5 \le r \le 1.5 \\ 0 & \text{elsewhere} \end{cases}$$
$$v(x,y,0) = \begin{cases} 1.6xr(-4r+3) & \text{if } r < 0.5 \\ 0.8x(r-1.5)^2 & \text{if } 0.5 \le r \le 1.5 \\ 0 & \text{elsewhere} \end{cases} \tag{2.34}$$

Again, the original data should repeat itself at the final time of $t = 2$.

Figure 2.12 reinforces the notion that the Active Flux scheme is able to preserve vorticity as the velocity profile returns to its initial data after a half period across the domain.

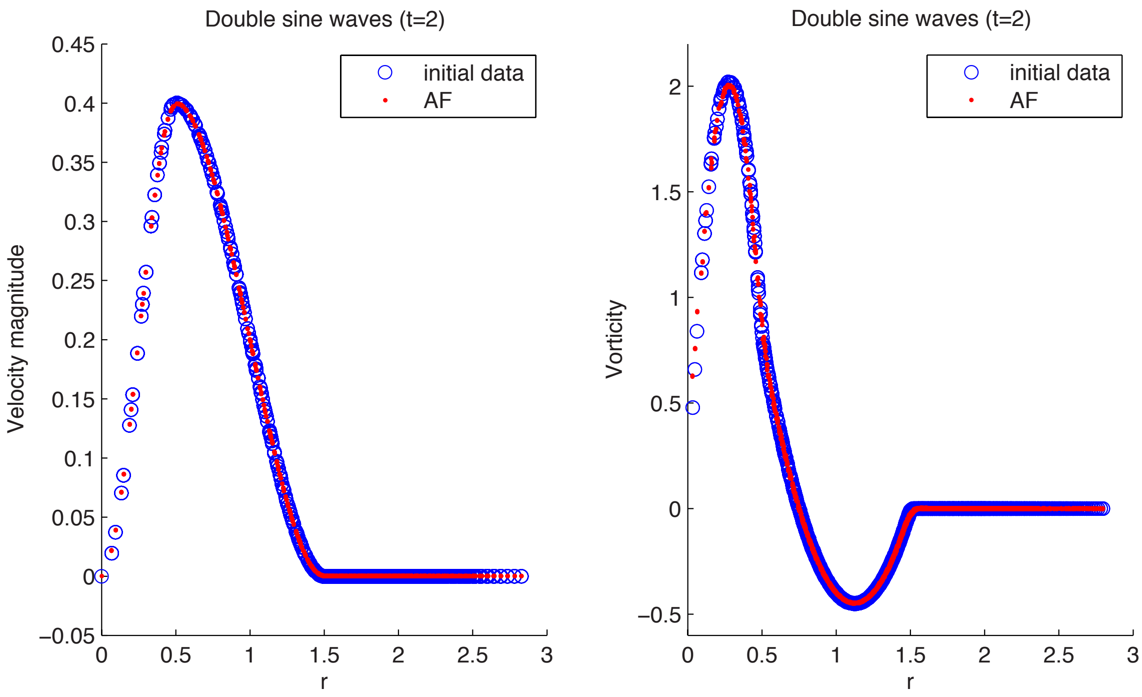


Figure 2.12: Velocity and vorticity profile along the radial direction for double sine waves problem with stationary vortex; $h = 0.05$, $t = 2$.

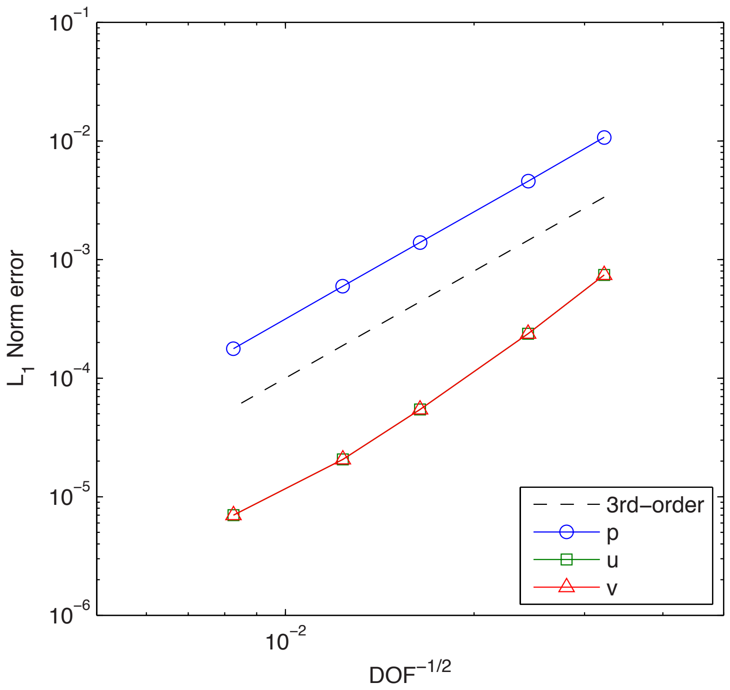


Figure 2.13: Convergence plot of AF errors for double sine waves problem with stationary vortex; $t = 2$.

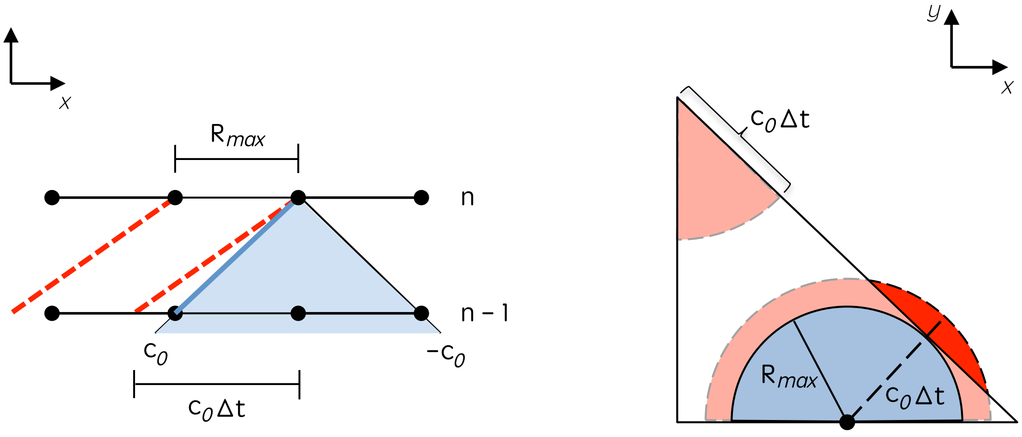
The vorticity is calculated as the average value in each element,

$$\begin{aligned}\overline{\omega} &= \frac{1}{\Omega_j}\int_{\Omega_j} \nabla \times \mathbf{v}\,\mathrm{d}\Omega \\ &= \frac{1}{\Omega_j}\int_{\Omega_j} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)\,\mathrm{d}\Omega\end{aligned} \tag{2.35}$$

and is slightly overestimated near the origin due to the velocity being nondifferentiable there. The velocity solution displays very convincing radial symmetry and the AF method achieves third-order convergence for all states, as confirmed in Figure 2.13.

2.4 Stability

The method of spherical means is an analytical solution to the homogeneous scalar wave problem $\partial_{tt}u = c^2\nabla^2 u$, and therefore its solution u at any given point is unique and dependent only on the prescribed initial values that are within the domain (of dependence) which is cut out by the backward characteristic ray cone through the said given point [31]. The physical requirement for stability is that the convex hull of the point values used to update a given point value must include this domain of dependence. We define a local Courant num-



(a) 1-D; bounded by characteristic lines.

(b) 2-D; bounded by characteristic disc.

Figure 2.14: Domains of dependence bounded by characteristic speeds of c_0. The blue regions within these domains are stable, while the red regions exterior are unstable.

ber at each point of $\text{CFL}_i = c_0\Delta t/|\Delta \mathbf{x}_i|$, which equals unity when the physical requirement is first violated. In one-dimension, the domain of dependence is bounded on the right and left by characteristic wavespeeds of $\pm c_0$, as shown in Figure 2.14(a). To satisfy the physical domain of dependence, the maximum timestep allowed is $\Delta t_{max} = \min(|\Delta x_i|)/c_0$.

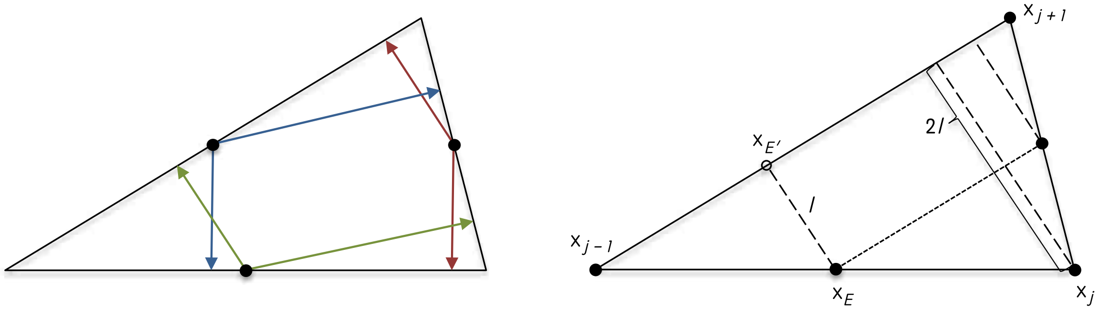
In two-dimensions, to satisfy the physical domain of dependence means that no disc centered at any point can cross over any element to which that point belongs, as demonstrated geometrically in Figure 2.14(b). This translates to a physically-stable maximum radius of integration $R_{\max} = c_0\Delta t_{\max}$, where we can again define a CFL condition:

$$\Delta t_{\max} = \frac{l_{\min}}{c_0} \tag{2.36}$$

where $l_{\min}$ is the minimum distance from a node to its opposing edges in a triangular element. Clearly, the distance from the edge midpoints are the most restrictive, and Figure 2.15(a) illustrates the various possibilities for $l_{\min}$. In practice, we only need to check one normal distance per edge [30]. Figure 2.15(b) shows how we can use the formula for the area of a triangle to determine the shorter normal length between the edge midpoint $\mathbf{x}_E$ and the opposite edge point $\mathbf{x}_{E'}$.

$$l_{\min} = \min_j \left(\frac{S}{|\mathbf{x}_{j+1} - \mathbf{x}_{j-1}|} \right) \tag{2.37}$$

Here, S is the element area. Equations (2.36) and (2.37) give a precise upper bound on the timestep. We will conduct numerical experiments to show that this upper bound is



(a) Minimum distances from edge midpoints.

(b) Nomenclature for minimum distance calculation.

Figure 2.15: Relevant distances for CFL calculation.

attainable.

To show numerical verification of these stability constraints, we have tested both one- and two-dimensional Active Flux solvers using various Courant numbers on the same one-dimensional nonsimple wave problem previously defined in Section $2.2.1$. We will attempt to increase the Courant number past its stable regime of 1 and observe the (possible) onset of numerical instability. The measure of stability we will use is the total acoustic energy of the system defined as

$$E^* = \frac{1}{2}(p^{*2} + |\mathbf{v}^*|^2) = \frac{1}{2}(p^{*2} + u^{*2} + v^{*2} + w^{*2}) \tag{2.38}$$

Taking the time-derivative of Equation (2.38) and substituting in the acoustic equations (2.1), it becomes clear that E^* is a conserved quantity for the linear acoustic system, and can be regarded as both an energy and entropy.

$$\frac{\partial E^*}{\partial t} = -c_0 \left(\frac{\partial (p^* u^*)}{\partial x} + \frac{\partial (p^* v^*)}{\partial y} + \frac{\partial (p^* w^*)}{\partial z} \right) \tag{2.39}$$

We may describe the method as "energy stable" if $\int E^*(t)\,\mathrm{d}x$ remains bounded.

We first observe the one-dimensional Active Flux scheme using the upwind method of characteristics is very sensitive to the Courant number, becoming unstable as soon as CFL > 1.0. Figure 2.16 shows that the total acoustic energy decreases at a steady rate in the stable regime of CFL < 1.0, is conserved when CFL is at unity, and increases sharply when CFL is larger than one. Interestingly, the same sensitivity to the Courant number cannot be concluded for the 2-D AF scheme using spherical means. Figure 2.17 shows that the acoustic energy does not increase before $t = 10$ unless the Courant number is close to 1.128 on a grid size of 50×50.

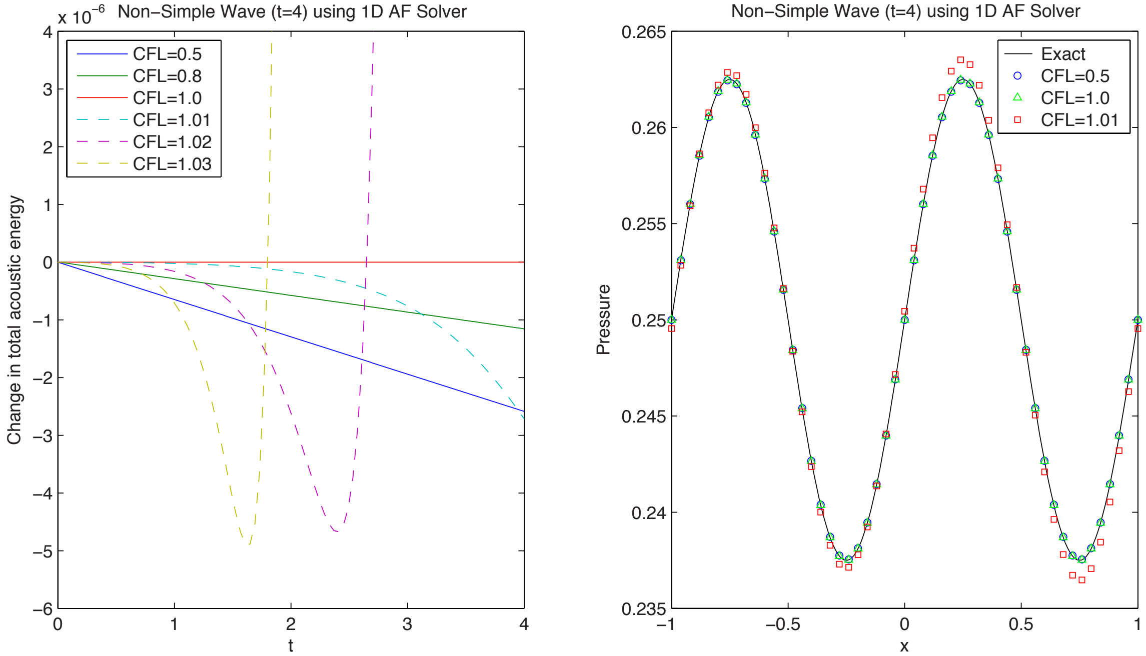


Figure 2.16: Nonsimple wave problem solved using 1-D AF method, $h = 0.04$, $t = 4$; shows the changes in total acoustic energy of the system (left) and pressure results deviating from the exact solution when CFL > 1 (right).

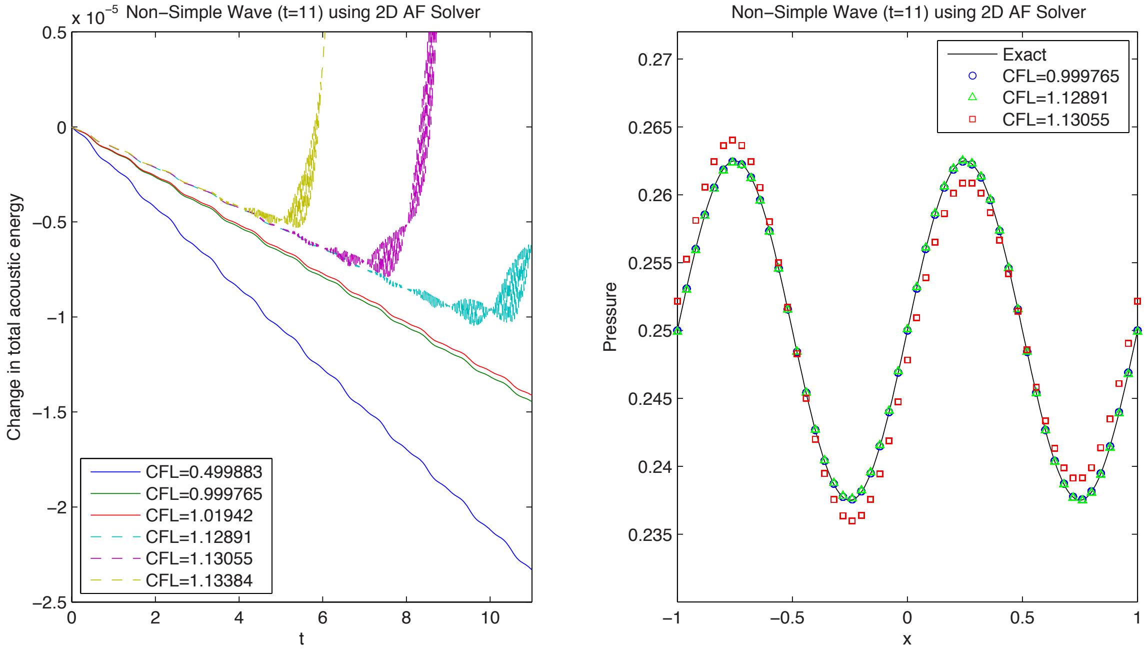


Figure 2.17: Nonsimple wave problem solved using 2-D AF method, $h = 0.04$, $t = 11$; shows the changes in total acoustic energy of the system (left) and pressure results deviating from the exact solution (right).

The two-dimensional results may seem to be inconsistent with theory at first. However, if we consider the regions in which the data points are used to update the solutions, the discrepancy may be explained. Figure 2.14 illustrates the theoretically stable and unstable regions from where the data is used to update the solution in one element. In the one-dimensional case, as the Courant number for any given point increases past unity, the data used to update the solution leaves the cell containing that point, and this violates the physical domain of dependence. In addition, the other point in the cell is also evaluated outside its own domain of dependence. This leads to the solution becoming unstable as soon as the CFL condition is violated.

By comparison, the two-dimensional AF method uses disc-like regions to update the solutions. The CFL condition is based on the minimum distance from an edge midpoint to its opposing edge, which is always a shorter length than the minimum distance from a vertex node to its opposing edge. As a result, only some of the nodes in a triangular element may be violating its physical domain of dependence as the Courant number increases above one. In addition, the smoothness of the initial data limits the maximum error that can be achieved when we update using point values outside the stability regime. Hence, instability will often be observed much later in time as the error very slowly builds up.

To verify this argument, we now test the 2-D AF method using higher frequency data.

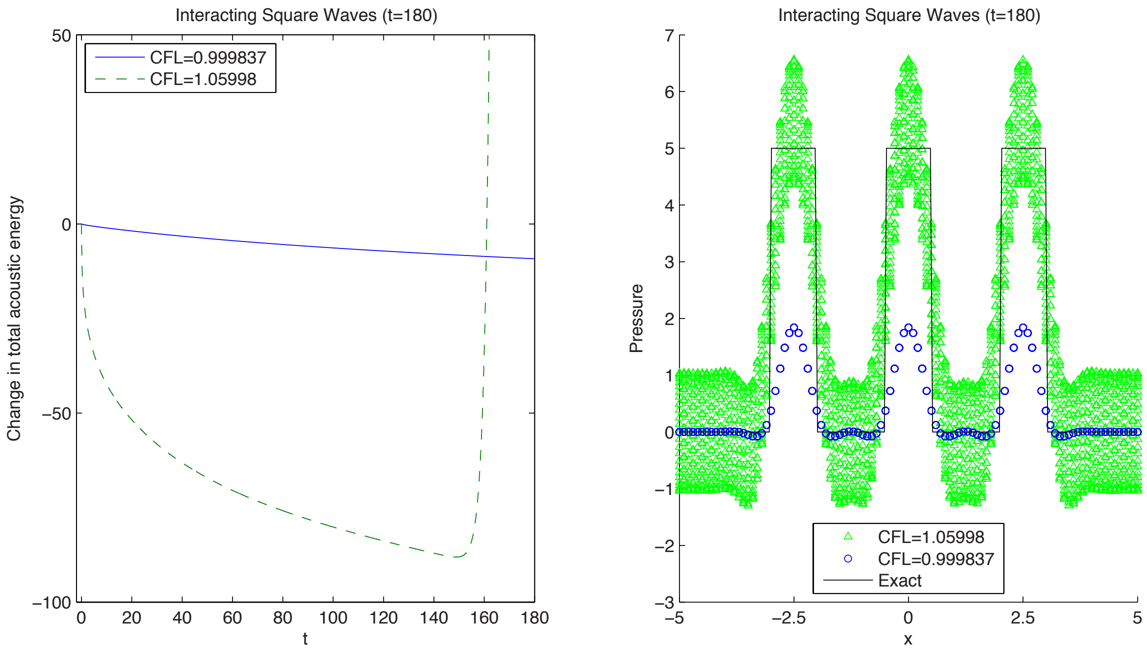


Figure 2.18: Three square waves problem solved using 2-D AF method, $h = 0.1$, $t = 180$; shows the changes in total acoustic energy of the system (left) and pressure results deviating from the exact solution (right).

As a variation of the square wave problem previously defined in Section 2.2.2, the initial data on the same structured grid now consists of three square waves of width $\Delta x = 1$ centered at $x = (-2.5, 0, 2.5)$ with a wave magnitude of $p = 5$. We notice that even with this discontinuous data, instability only sets in after a long runtime of $t > 160$ with a lower Courant number of 1.06, as shown in Figure 2.18. We also observe that the stable solution has smoothed out and decreased the amplitude of the square waves by a significant amount, which could explain the late onset of instability. It still exhibits one-dimensional solution in the x-direction and its acoustic energy remains bounded and decreases at a relatively slow rate. On the other hand, the unstable solution has lost its one-dimensional quality and its total acoustic energy increases rapidly after $t = 160$. These results verify that instability will eventually occur if the stability limit is violated, granted that may take a very long time if the data is or becomes smooth.

2.5 Boundary considerations

The test problems we have considered so far have used periodic boundaries, which do not require any additional numerical considerations, to evaluate the accuracy of the AF method. In order to be able to test more complicated problems, we will now introduce two new types of boundaries: open and flat wall, both of which can be implemented in the AF scheme in a simple and natural way based on the correct physics.

2.5.1 Open boundary

Quite often, open or nonreflecting boundaries are the sources of the most significant numerical errors in a computational acoustics or computational fluid dynamics method [34]. Numerous types of nonreflecting boundary conditions have been developed to cope with the open-domain problem, including far-field asymptotic solutions, the buffer zone technique, the perfectly matched layer technique, and characteristic-based inflow and outflow boundary conditions. The last technique is desirable for schemes with upwinding features, and often works best when the wave angle is normal to the boundary. However, its performance can deteriorate when the wave angle deviate from normal, with errors of up to 3–5% for larger angles of incidence [35, 36]. For example, Thompson showed that a traveling shockwave moving at a 45° angle with respect to the x-axis produces a pressure error of approximately 4% after the shockwave has left the domain [37].

Our approach to handling open boundaries is similar to the characteristic-based outflow boundary condition. However, instead of applying the one-dimensional characteristics to multidimensions as is traditionally done [36], we will directly use the multidimensional physics of acoustic waves. The data that influence the flow at an open boundary is simply the point values inside the domain. As such, we need only to evaluate the spherical means integral over the portion of the disc that is inside the domain and disregard the portion that is outside, as illustrated in Figure 2.19 for both edge and vertex nodes. Verification of this boundary condition will be provided later for the nonlinear acoustics system in Section 3.3.

2.5.2 Flat wall boundary

The implementation of flat wall boundaries is quite straightforward. Physically, the changes in pressure and tangential velocity due to the external flow past a flat wall are doubled while the normal velocity is cancelled. At a right corner, there is a stagnation point where the velocity is zero and the changes in pressure is quadrupled. Taking these conditions into account, we can test a more complicated problem consisting of four sinusoidal waves with a stationary vortex as described in Equation (2.40). Each sinusoidal wave is centered at $x = \pm 10$ and $y = \pm 10$, and the vortex is centered at $(0, 0)$.

$$\Delta p(x, y, 0) = \begin{cases} 0.5 + 0.5\cos(0.5\pi(x-10)) & \text{if } 8 \leq x \leq 12 \\ 0.5 + 0.5\cos(0.5\pi(x+10)) & \text{if } -12 \leq x \leq -8 \\ 0.5 + 0.5\cos(0.5\pi(y-10)) & \text{if } 8 \leq y \leq 12 \\ 0.5 + 0.5\cos(0.5\pi(y+10)) & \text{if } -12 \leq y \leq -8 \\ 0 & \text{elsewhere} \end{cases}$$

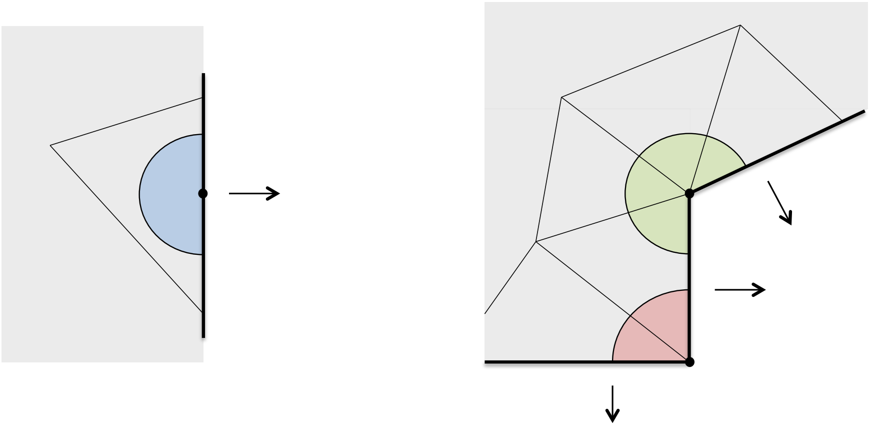


Figure 2.19: Partial discs of integration at open boundaries. The interior domain is colored gray.

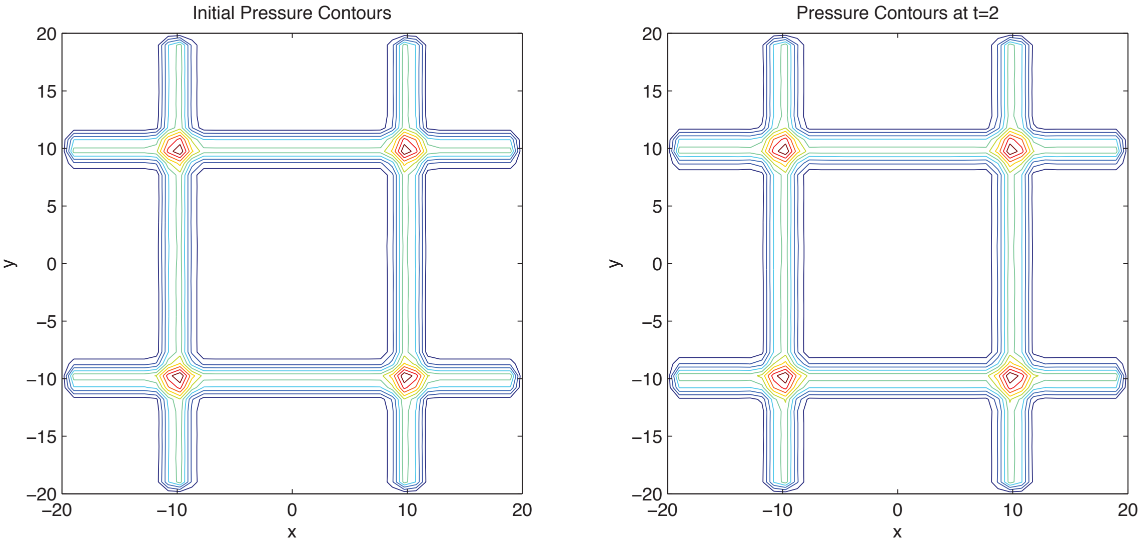


Figure 2.20: Pressure contours for the interacting sine wave problem at $t = 0$ (left) and $t = 2$ (right); 80×80 grid.

$$
\begin{aligned}
p(x,y,0) &= 1.0 + \Delta p(x,y,0) \\
u(x,y,0) &= \begin{cases} -0.0125\, y(r-6)^2 & \text{if } r \le 6 \\ 0 & \text{elsewhere} \end{cases} \\
v(x,y,0) &= \begin{cases} 0.0125\, x(r-6)^2 & \text{if } r \le 6 \\ 0 & \text{elsewhere} \end{cases}
\end{aligned}
\qquad (2.40)
$$

The sound speed is set to $c_0 = 10$ and the AF scheme is run until $t = 2$, by which point the initial data repeats itself, as shown in Figure 2.20. A snapshot of an intermediate solution at $t = 0.8$ is also shown in Figure 2.21 to illustrate the symmetry and complexity of this particular problem.

Figure 2.22 shows that the final velocity profile follows closely to the initial data, and in fact, this problem was continued until $t = 40$ without any further visual change in the vorticity. However, we observe the pressure waves have decreased in amplitude. We also note that there are oscillations in the pressure profile at the foot of the sine waves. This effect is due to the steep gradients in the pressure data at that location. Unfortunately, without limiting at this time, it is difficult to achieve non-oscillatory solutions with such data. As a result, the rate of convergence for this problem is less than three for all three states. However, the maximum pressure peaks are not as affected by the gradients and do in fact converge at third-order, as shown in Figure 2.23. This promising result provides incentive for us to incorporate limiting into the AF scheme in order to properly handle data

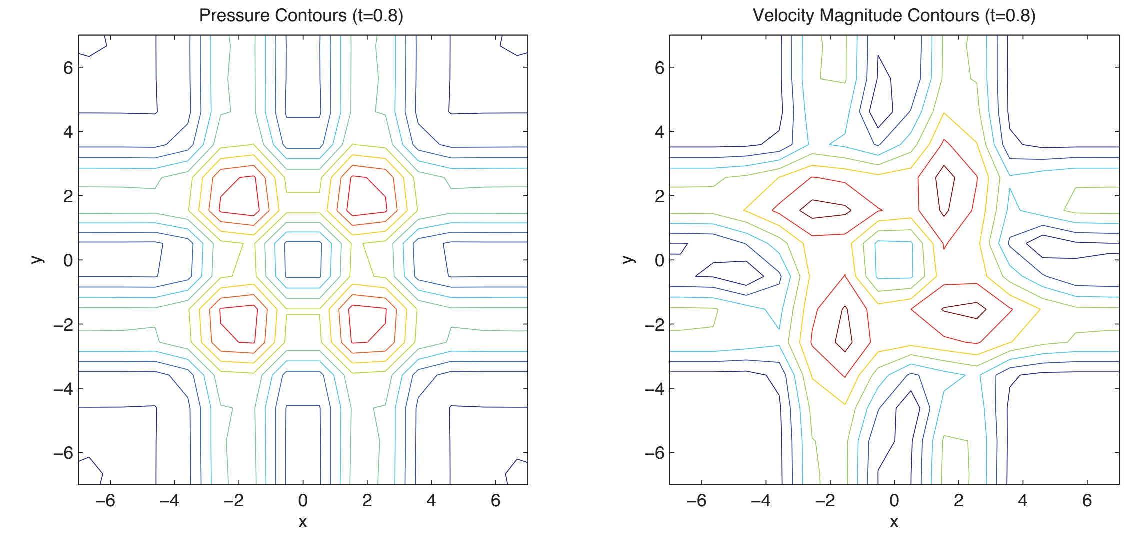


Figure 2.21: Pressure (left) and velocity (right) contours for the interacting sine wave problem at intermediate time $t = 0.8$; 80×80 grid.

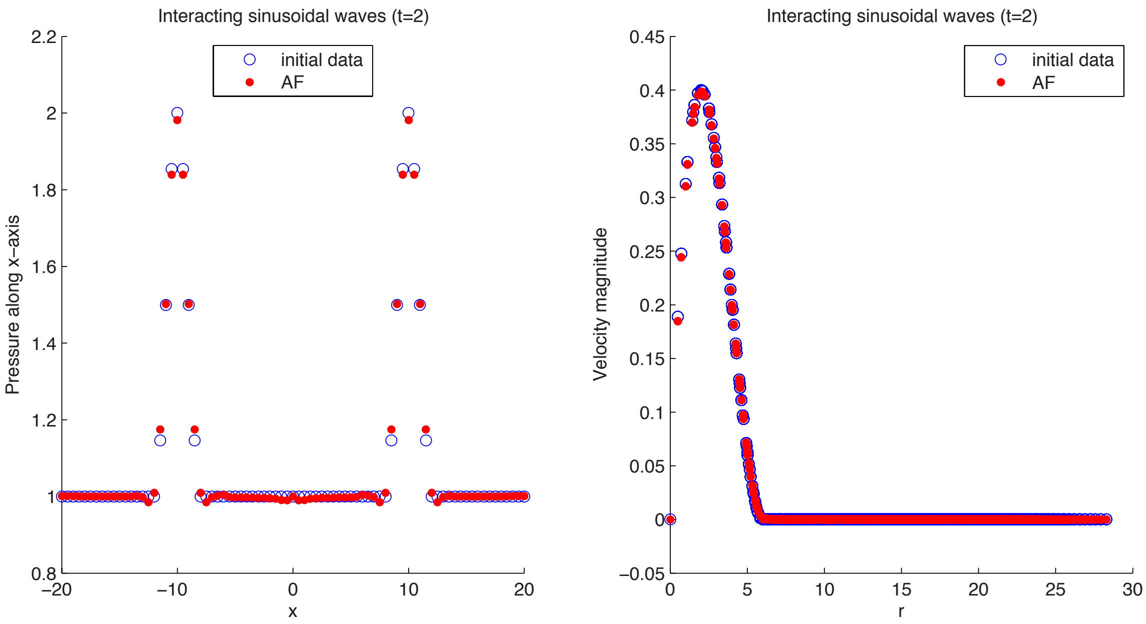


Figure 2.22: Pressure (left) and velocity (right) profiles for the interacting sine wave problem on 80×80 grid; $t = 2$.

with steep gradients.

One final aspect we can test with this problem is the ability of the AF scheme to preserve vorticity in more complex flows. The problem is run for 20 periods ($t = 40$) and the maximum pressure, velocity magnitude, and vorticity are plotted in Figure 2.24. Note that the vorticity values are computed using Equation (2.35). As expected, the pressure peaks

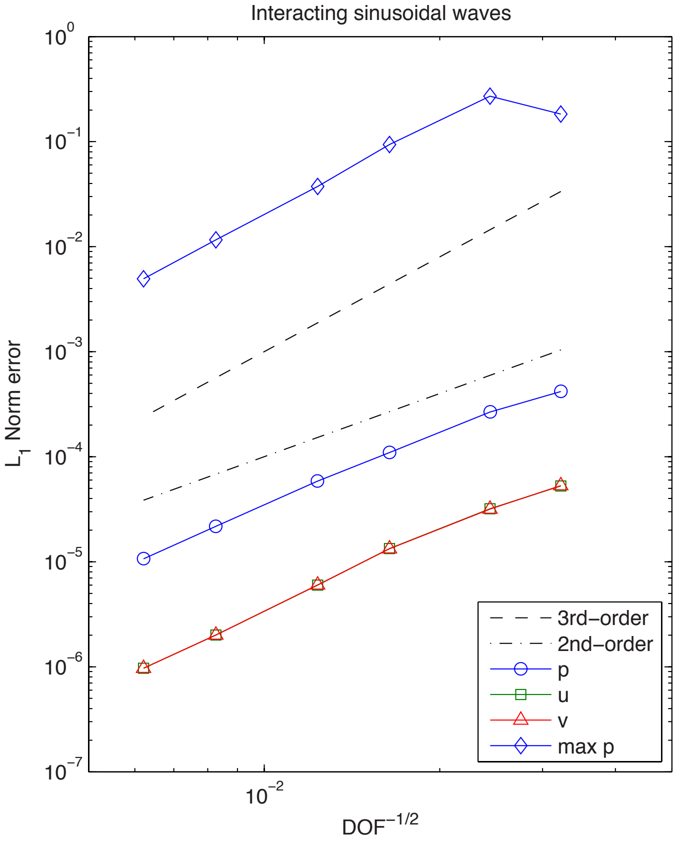


Figure 2.23: Convergence plot of pressure, velocities, and maximum pressure peak at $t = 2$.

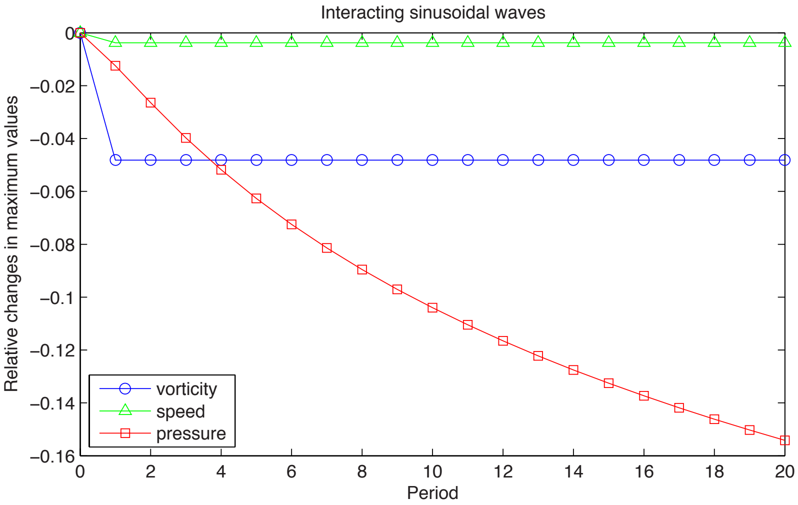


Figure 2.24: Time evolution of maximum pressure peak, vorticity, and velocity magnitude; 80×80 grid.

continue to decrease in time due to the lack of limiting. However, both maximum velocity magnitude and vorticity remain constant after an initial dip at $T = 1$. These are very convincing results suggesting that the AF scheme is capable of preserving a certain property of the acoustic process. While it does not seem to preserve the exact initial vorticity of the problem, the method may be preserving a discrete version of vorticity instead. This attribute of the AF method could potentially be a large advantage when solving other hyperbolic conservation laws where certain properties need to be preserved.

CHAPTER 3

Nonlinear Acoustics

In this chapter, we will explore the Active Flux method applied to nonlinear acoustic waves by studying the p-system of hyperbolic conservation laws, where pressure is a function only of density. Note that this is not a true limit of the Euler equations, but is often used to illustrate mathematical principles because of its simple behavior [38].

$$\frac{\partial \rho}{\partial t} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \tag{3.1a}$$

$$\frac{\partial u}{\partial t} + \frac{\partial p(\rho)}{\partial x} = 0 \tag{3.1b}$$

$$\frac{\partial v}{\partial t} + \frac{\partial p(\rho)}{\partial y} = 0 \tag{3.1c}$$

$$\frac{\partial w}{\partial t} + \frac{\partial p(\rho)}{\partial z} = 0 \tag{3.1d}$$

We then take the second time-derivatives of Equation (3.1) to obtain inhomogeneous wave equations for the density and velocity states.

$$\begin{aligned} \frac{\partial^2 \rho}{\partial t^2} - p'(\rho)\nabla^2 \rho &= \rho(\nabla \cdot \mathbf{u})^2 + \left(p''(\rho) - \frac{p'(\rho)}{\rho} \right) \nabla \rho \cdot \nabla \rho \\ \frac{\partial^2 \mathbf{u}}{\partial t^2} - p'(\rho)\nabla^2 \mathbf{u} &= p'(\rho)\nabla \times \nabla \times \mathbf{u} + p''(\rho)(\nabla \cdot \mathbf{u})\nabla \rho \end{aligned} \tag{3.2}$$

The propagation of signals in Equation (3.2) takes place at a variable sound speed of $c = \sqrt{p'(\rho)}$. We will determine $p(\rho)$ by making the isentropic assumption $p = K\rho^\gamma$, where γ is the specific heat ratio and K is the isentropic constant. Hence, the formula for the sound speed becomes the familiar expression $c = \sqrt{\gamma p/\rho}$. Equation (3.2) can then be re-written in terms of c, where the pressure derivatives $p'(\rho) = c^2$ and $p''(\rho) = (\gamma - 1)c^2/\rho$

have been used.

$$
\begin{aligned}
\frac{\partial^2 \rho}{\partial t^2} - c^2 \nabla^2 \rho = \rho(\nabla \cdot \mathbf{u})^2 + \frac{\gamma - 2}{\rho} c^2 \nabla \rho \cdot \nabla \rho \\
\frac{\partial^2 \mathbf{u}}{\partial t^2} - c^2 \nabla^2 \mathbf{u} - c^2 \nabla \times \nabla \times \mathbf{u} = \frac{\gamma - 1}{\rho} c^2 (\nabla \cdot \mathbf{u}) \nabla \rho
\end{aligned}
\tag{3.3}
$$

Clearly, the left-hand side of Equation (3.3) is formally identical to Equation (2.2) describing linear acoustics in the special case of $p = \rho$. The only differences between the linear and nonlinear systems are that the sound speed is no longer constant and that there are additional second-order terms on the right-hand side of the nonlinear acoustic equations. This similarity in mathematical structure allows us to apply the same numerical analysis to obtain an analytical solution to Equation (3.1) to be used in the Active Flux scheme. The AF method is then compared to the first-order Discontinuous Galerkin (DG1) method in terms of accuracy and convergence. We will end this chapter with a discussion on the implementation of open boundaries in the AF scheme.

3.1 Approximate solutions to the p-system

Let the state vector now be $\mathbf{q} = (\rho, u, v, w)^T$. We can rewrite the p-system described by Equation (3.1) as

$$
\frac{\partial \mathbf{q}}{\partial t} + c\boldsymbol{L}\mathbf{q} = \mathbf{0} \tag{3.4}
$$

where the sound speed is defined as $c = \sqrt{p'(\rho)}$, and

$$
\boldsymbol{L} = \begin{pmatrix}
0 & \frac{1}{c}\frac{\partial}{\partial x} & \frac{1}{c}\frac{\partial}{\partial y} & \frac{1}{c}\frac{\partial}{\partial z} \\
c\frac{\partial}{\partial x} & 0 & 0 & 0 \\
c\frac{\partial}{\partial y} & 0 & 0 & 0 \\
c\frac{\partial}{\partial z} & 0 & 0 & 0
\end{pmatrix} \tag{3.5}
$$

Equation (3.4) is formally identical to the linear acoustics system described by Equation

(2.9).

$$\begin{aligned}\mathbf{q}(\mathbf{x},t) &= e^{-c\boldsymbol{L}t}\mathbf{q}(\mathbf{x},0)\\ &= \sum_{p=0}^{\infty}\frac{(-c\boldsymbol{L}t)^p}{p!}\mathbf{q}(\mathbf{x},0)\end{aligned} \tag{3.6}$$

The gradient matrix $\boldsymbol{L}$ also satisfies the properties

$$\boldsymbol{L}^{p+2} = \nabla^2\boldsymbol{L}^p \qquad (p > 0)$$

and

$$\boldsymbol{L}^2 = \begin{pmatrix} \frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2} & 0 & 0 & 0 \\ 0 & \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial x\partial y} & \frac{\partial^2}{\partial x\partial z} \\ 0 & \frac{\partial^2}{\partial x\partial y} & \frac{\partial^2}{\partial y^2} & \frac{\partial^2}{\partial y\partial z} \\ 0 & \frac{\partial^2}{\partial x\partial z} & \frac{\partial^2}{\partial y\partial z} & \frac{\partial^2}{\partial z^2} \end{pmatrix}$$

Following the same derivation as with the linear acoustics equations, we can expand the formal solution as the sum of even and odd terms.

$$\mathbf{q}(t) = \mathbf{q}(0) - t\left\{\sum_{p=0}^{\infty}\frac{(ct)^{2p}}{(2p+1)!}\nabla^{2p}\right\}c\boldsymbol{L}\mathbf{q}(0) + t^2\left\{\sum_{p=0}^{\infty}\frac{(ct)^{2p}}{(2p+2)!}\nabla^{2p}\right\}c^2\boldsymbol{L}^2\mathbf{q}(0) \tag{3.7}$$

The odd terms can again be replaced by the spherical means formula. We then note that the second time-derivatives can be written as

$$\partial_{tt}\mathbf{q} + \partial_t(c\boldsymbol{L})\mathbf{q} - c^2\boldsymbol{L}^2\mathbf{q} = \mathbf{0} \tag{3.8}$$

which can be substituted into Equation (3.7) to obtain an exact solution to Equation (3.1) using spherical means.

$$\mathbf{q}(t) = \mathbf{q}(0) + tM_R\{\partial_t\mathbf{q}\}(0) + \frac{1}{c^2}\int_0^R rM_r\{\partial_{tt}\mathbf{q} + \partial_t(c\boldsymbol{L})\mathbf{q}\}(0)\,\mathrm{d}r \tag{3.9}$$

Now the only difference between the exact solution to the linear problem and Equation (3.9) is the addition of the time derivative term consisting of the gradient matrix multiplied by the sound speed. At this time, we would like to mention that for a third-order scheme, the fluxes are only required to be second-order accurate, which means that only the leading nonlinear correction term is needed. Hence, the fourth term, which is of quadratic order, need only be evaluated using the local sound speed. This means that $c\boldsymbol{L}$ remains a constant when evaluating the integral, and thus the last term vanishes and we are left with the following second-order solution to the nonlinear acoustics system.

$$\mathbf{q}(t) = \mathbf{q}(0) + tM_R\{\partial_t \mathbf{q}\}(0) + \frac{1}{c^2}\int_0^R rM_r\{\partial_{tt}\mathbf{q}\}(0)\,\mathrm{d}r \tag{3.10}$$

Note that this again resembles the second-order Lax-Wendroff method. Simplifying the nonlinear acoustic solutions using Equations (3.1) and (3.2), we get the following density and velocity evolution formulas.

$$\begin{aligned}\rho(\mathbf{x},t) = \rho(\mathbf{x},0) - tM_R\{\nabla\cdot\mathbf{u}\} + \frac{1}{c^2}\int_0^R rM_r\{c^2\nabla^2\rho\}\,\mathrm{d}r \\ + \frac{1}{c^2}\int_0^R rM_r\left\{\rho(\nabla\cdot\mathbf{u})^2 + \frac{(\gamma-2)c^2}{\rho}\nabla\rho\cdot\nabla\rho\right\}\mathrm{d}r\end{aligned} \tag{3.11}$$

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = \mathbf{u}(\mathbf{x},0) - tM_R\{\nabla p\} + \frac{1}{c^2}\int_0^R rM_r\{c^2\nabla^2\mathbf{u} + c^2\nabla\times\nabla\times\mathbf{u}\}\,\mathrm{d}r \\ + \frac{\gamma-1}{c^2}\int_0^R rM_r\left\{\frac{c^2}{\rho}(\nabla\cdot\mathbf{u})\nabla\rho\right\}\mathrm{d}r\end{aligned} \tag{3.12}$$

3.1.1 Approximation of second-order terms

Recall that for a third-order scheme, only the leading nonlinear correction is needed. Therefore, the point values are evolved according to Equations (3.11) and (3.12) by evaluating the spherical means integral using a radius proportional to the local sound speed at each node, $R = c_i t = (\sqrt{\gamma p_i/\rho_i})t$, as illustrated in Figure 3.1. This is rather similar to conventional numerical methods that use a linearized Riemann solver.

$$\rho(\mathbf{x},t) = \rho(\mathbf{x},0)\ - tM_R\{\nabla\cdot\mathbf{u}\} + \int_0^R rM_r\{\nabla^2\rho\}\,\mathrm{d}r$$

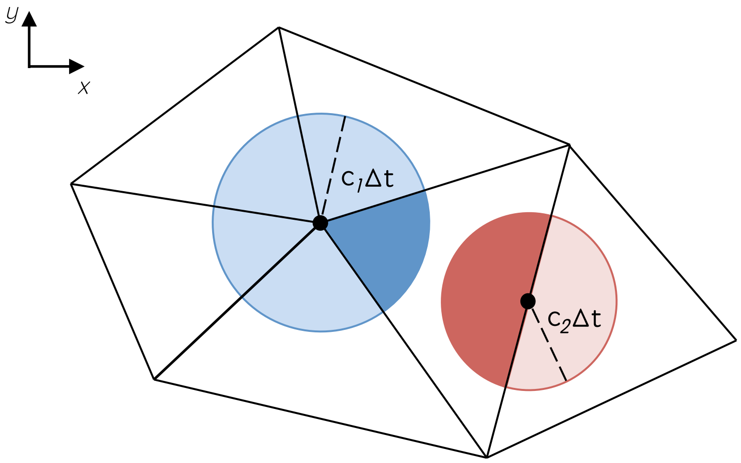


Figure 3.1: Discs of integration with varying local radii.

$$+\frac{\rho_i}{c_i^2}\int_0^R rM_r\{(\nabla\cdot\mathbf{u})^2\}\,\mathrm{d}r+\frac{\gamma-2}{\rho_i}\int_0^R rM_r\{\nabla\rho\cdot\nabla\rho\}\,\mathrm{d}r \tag{3.13a}$$

$$\mathbf{u}(\mathbf{x},t)=\mathbf{u}(\mathbf{x},0)\;-tM_R\{\nabla p\}+\int_0^R rM_r\{\nabla^2\mathbf{u}\}\,\mathrm{d}r+\cancelto{\mathbf{0}}{\int_0^R rM_r\{\nabla\times\nabla\times\mathbf{u}\}\,\mathrm{d}r}$$

$$+\frac{\gamma-1}{\rho_i}\int_0^R rM_r\left\{(\nabla\cdot\mathbf{u})\nabla\rho\right\}\,\mathrm{d}r \tag{3.13b}$$

where the crossed-out term vanishes due to the analysis given in Section 2.1.3. Note also that the gradient of pressure can be written as a gradient of density by using the isentropic relation, $\nabla p = c^2\nabla\rho$.

The solution given by Equation (3.13) is a combination of the linear acoustics solution described by the first lines, and second-order nonlinear terms in the second lines. These additional terms may seem to be much more complicated to compute because they are the product of two gradient functions and need to be integrated twice. However, some simplifications can be made since only the leading term in these nonlinear terms needs to be evaluated to retain second-order accuracy.

Let us start by first observing that given quadratic data for the state solutions, the state gradients are linear functions. All of the nonlinear terms are a product of two gradient functions, which results in the leading term being a constant. It then follows that the nonlinear terms can be approximated using two linear functions $g_1(x, y)$ and $g_2(x, y)$ thusly:

$$\begin{aligned}\int_0^R rM_r\{g_1(x,y)g_2(x,y)\}\,\mathrm{d}r &= \int_0^R r\,\mathrm{d}r\,M_R\{g_1(x,y)g_2(x,y)\} + \mathcal{O}(R^3)\\ &\approx \frac{R^2}{2}\left(M_R\{g_1\}M_R\{g_2\} + \mathcal{O}(R)\right)\\ &\approx \frac{R^2}{2}M_R\{g_1\}M_R\{g_2\}\end{aligned} \tag{3.14}$$

Taking advantage of the first-order gradient terms that have already been calculated to further reduce the computational cost of the nonlinear terms, the final form of the solution becomes

$$\begin{aligned}\rho(\mathbf{x},t) = \rho(\mathbf{x},0)\; - tM_R\{\nabla\cdot\mathbf{u}\} + \int_0^R rM_r\{\nabla^2\rho\}\,\mathrm{d}r&\\ + \frac{t^2}{2}\rho_i\left(M_R\{\nabla\cdot\mathbf{u}\}\right)^2 + \frac{t^2}{2}\frac{(\gamma-2)c_i^2}{\rho_i}M_R\{\nabla\rho\}\cdot M_R\{\nabla\rho\}&\end{aligned} \tag{3.15a}$$

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = \mathbf{u}(\mathbf{x},0) - tM_R\{\nabla p\} + \int_0^R rM_r\{\nabla^2\mathbf{u}\}\,\mathrm{d}r&\\ + \frac{t^2}{2}\frac{(\gamma-1)c_i^2}{\rho_i}M_R\{\nabla\cdot\mathbf{u}\}M_R\{\nabla\rho\}&\end{aligned} \tag{3.15b}$$

We shall end this section by making the observation that Equation (3.15) is in some ways a combination of the spherical means method and the Lax-Wendroff approximation. The first three terms describe linear acoustics and are evaluated exactly using spherical means, which would retain favorable wave properties such as radial symmetry and vorticity preservation. We can also combine the first and third term using Equation (3.16) as we did previously for the linear acoustics solution, if desired.

$$\mathbf{q}(\mathbf{x}) + \int_0^R rM_r\{\nabla^2\mathbf{q}\}(\mathbf{x})\,\mathrm{d}r = M_R\{\mathbf{q}\}(\mathbf{x}) + R\partial_R M_R\{\mathbf{q}\}(\mathbf{x}) \tag{3.16}$$

By comparison, to maintain second-order accuracy, the nonlinear terms need only be approximated by the leading constant, which means we can simply evaluate their point values as we would do in the Lax-Wendroff method. However, while the data is continuous at every nodal point in the AF method, the gradients are in general discontinuous, but this does not prevent us from integrating them. Hence, an average gradient value at each node is used instead by taking advantage of the definition of the spherical means integral.

3.2 Convergence and accuracy

The Active Flux method in Equation (3.15) for the nonlinear acoustics system described by Equation (3.1) is tested against the Discontinuous Galerkin method with linear reconstruction (DG1) in terms of accuracy and convergence. We have chosen to compare results with the DG1 method because it uses the same number of degrees of freedom as AF. It has also been shown to sometimes superconverge at third-order when solution variables are averaged or integrated, which means that, in this context, we consider it to be the closest competitor to the AF method.

3.2.1 Radial symmetry

We would like to investigate the ability of the Active Flux method to maintain radial symmetry. If the solution is perfectly symmetric, a scatter plot of the solution at individual nodal values should collapse to a single line. The initial conditions are described by Equation (3.17) and is a variant of the linear Gaussian pulse used by Eymann [30], where the far-field density is $\rho_\infty = 1.0$, $K = 1/\gamma$, and $\gamma = 1.4$.

$$\begin{aligned} \rho(\mathbf{x},0) &= \rho_\infty + 2\exp\{-50(x^2+y^2)\} \\ u(\mathbf{x},0) &= 0 \\ v(\mathbf{x},0) &= 0 \\ p(\mathbf{x},0) &= K\rho^\gamma(\mathbf{x},0) \end{aligned} \tag{3.17}$$

The final time is set to $t = 1.25$ to ensure that no shocks have yet developed. In addition, the maximally stable Courant number of 1.0 is used for this problem.

The AF scheme is compared to the DG1 method on grids of three sizes: 40×40 (coarse), 80×80 (medium), and 160×160 (fine). Solutions of both density and velocity magnitude are plotted along the radial direction in Figures 3.2 to 3.4. Clearly, the AF solutions display better radial symmetry than DG1 in all of the plots as well as show smaller magnitudes of undershoots and overshoots, which is especially obvious on the coarse and medium size grids. The AF method was also better at predicting the maximum density and velocity values of 1.24 and 0.252 at around $r = 1.38$ on the coarser grids.

To determine the order of convergence, we computed a scalar output $I(t)$ consisting of

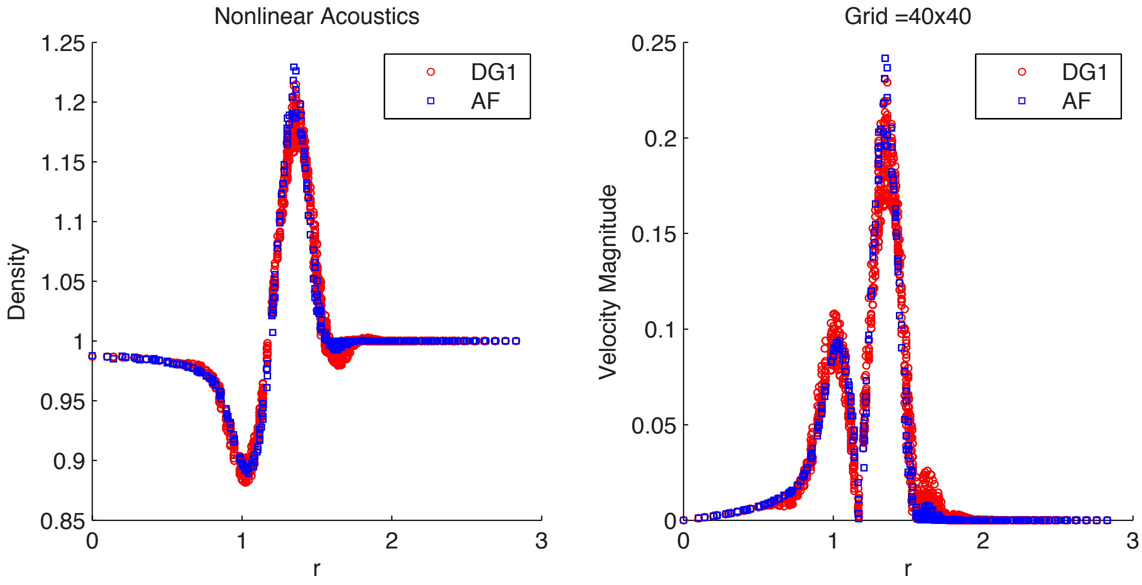


Figure 3.2: AF solution for Gaussian pulse on coarse grid; $t = 1.25$.

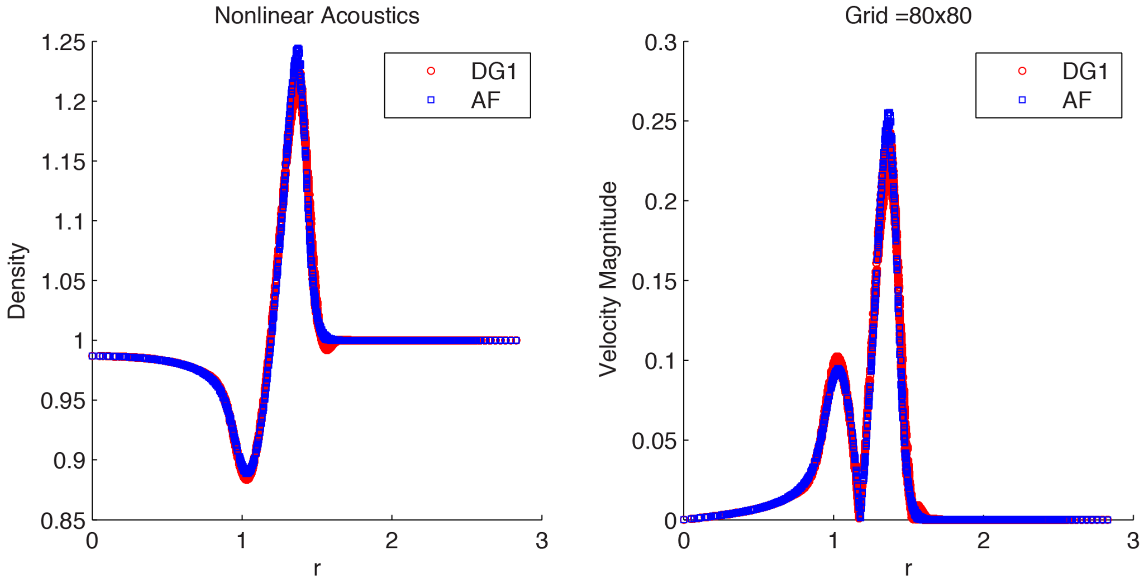


Figure 3.3: AF solution for Gaussian pulse on medium grid; $t = 1.25$.

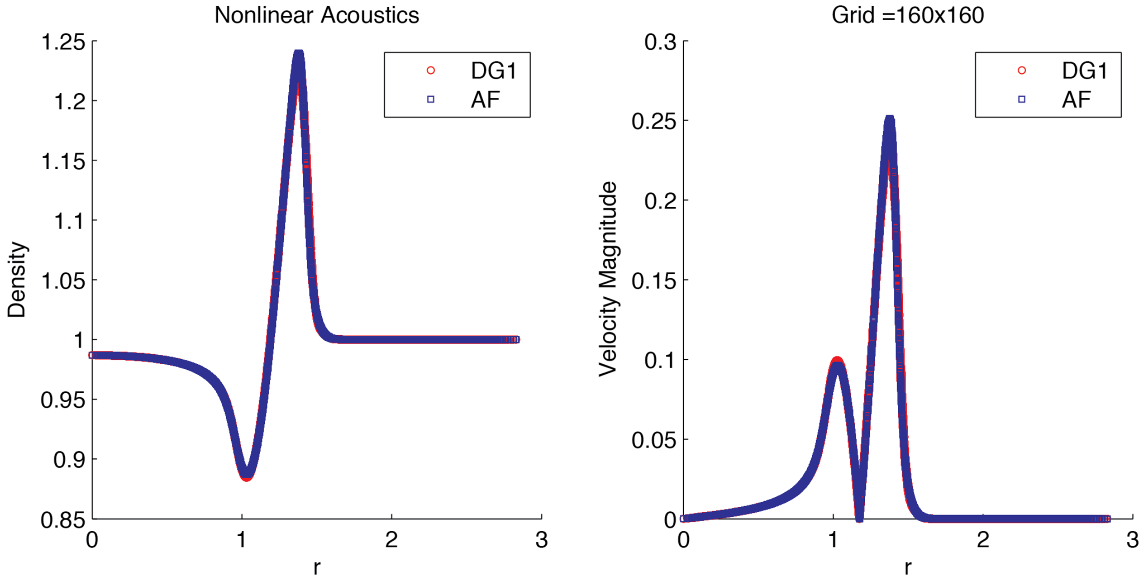


Figure 3.4: AF solution for Gaussian pulse on fine grid; $t = 1.25$.

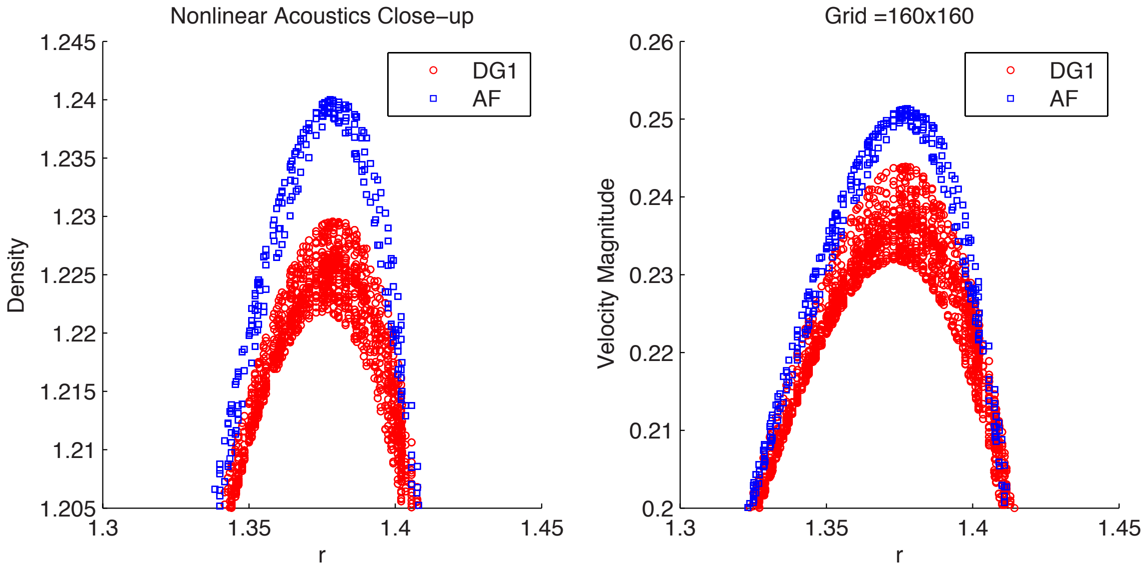


Figure 3.5: Close-up of solution on fine grid.

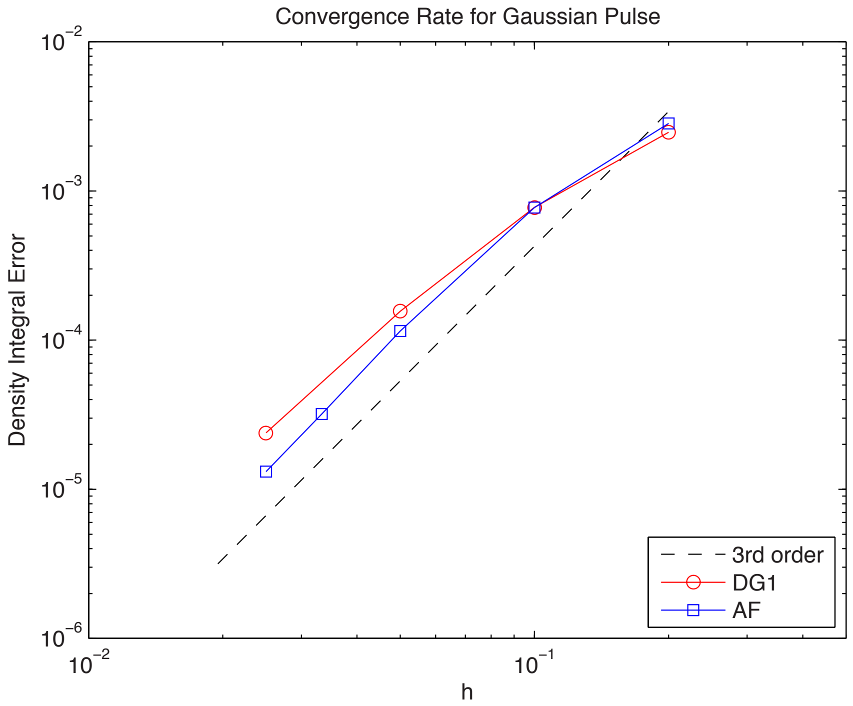


Figure 3.6: Convergence plot of AF and DG1 solutions for Gaussian pulse problem.

h	Active Flux		DG1	
	$\|I - I_{ref}\|$	$\mathcal{O}(\Delta I)$	$\|I - I_{ref}\|$	$\mathcal{O}(\Delta I)$
2.0000×10^{-1}	2.8393×10^{-3}	–	2.4765×10^{-3}	–
1.0000×10^{-1}	7.7461×10^{-4}	1.8740	7.7582×10^{-4}	1.6745
5.0000×10^{-2}	1.1504×10^{-4}	2.7513	1.5628×10^{-4}	2.3116
3.3333×10^{-2}	3.1982×10^{-5}	3.1571	–	–
2.5000×10^{-2}	1.3126×10^{-5}	3.0958	2.3918×10^{-5}	2.7079

Table 3.1: Convergence errors of scalar output I.

a weighted integral of the density distribution.

$$I(t) = \int_{\Omega} (\rho(\mathbf{x}, t) - \rho_\infty) w(\mathbf{x}) \, \mathrm{d}\Omega \tag{3.18}$$

The weight function is radially symmetric,

$$w(x, y) = \frac{128}{3}(x^2 + y^2)^2 \exp(-4\sqrt{x^2 + y^2})$$

and normalized, in the sense that

$$\int_0^\infty \int_0^\infty w(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{128}{3}\int_0^\infty r^4 \exp(-4r)\,\mathrm{d}r = 1$$

The error in the scalar output calculated using Equation (3.18) was determined by comparing with a reference solution computed using the fifth-order DG method using quadratic reconstruction (DG2) on a 160×160 grid. Both Figure 3.6 and Table 3.1 show that the AF method clearly converges at third-order while DG1 does not quite reach third-order convergence. Additionally, it is relevant to note that DG1 was initially showing only second-order convergence, and the current result was only obtained after carefully choosing the initialization of the data. We did not encounter this restriction with the AF method.

In terms of the error values of the density integral output itself, the DG1 method seems to perform almost on par with AF, which is slightly puzzling given their solutions from Figures 3.2 to 3.5. One possible explanation for this inconsistency is to analyze the weight function, which is plotted in Figure 3.7. The maximum value of $w(r)$ occurs at $r = 1$, which happens to be the approximate location of the minimum density. The DG1 solution at this location is rather smooth and much closer to the AF result than at the maximum density peak. Hence, the density integrals computed from DG1 results are comparable to those from Active Flux.

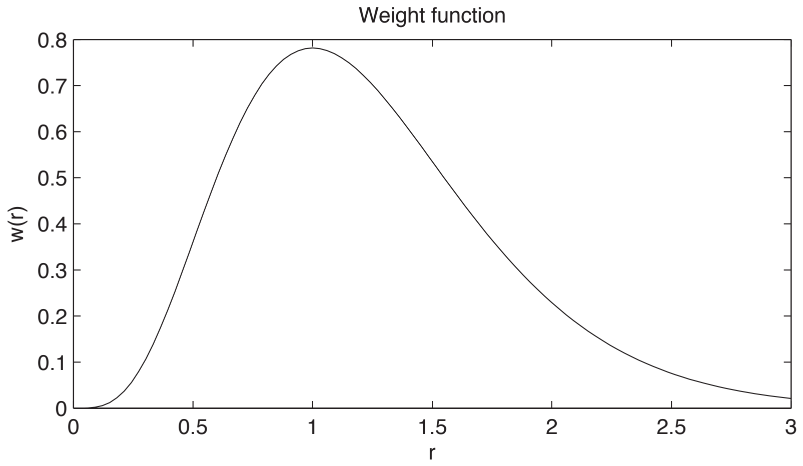


Figure 3.7: Weight function used in density integral calculation.

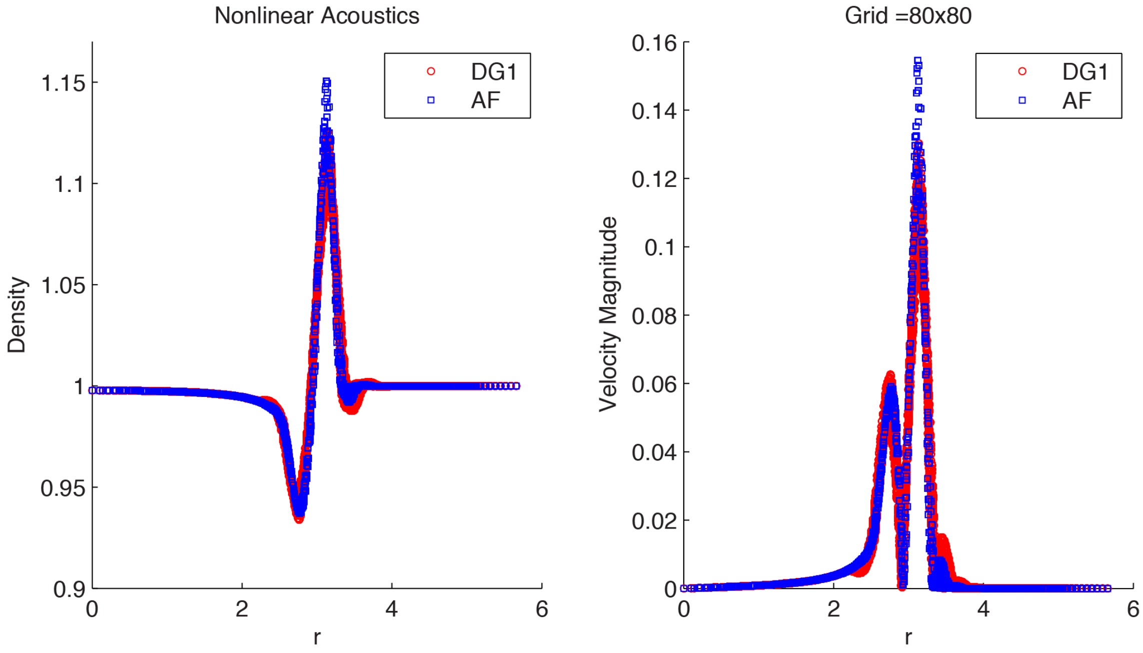


Figure 3.8: AF and DG1 solutions for Gaussian pulse at $t = 3.0$.

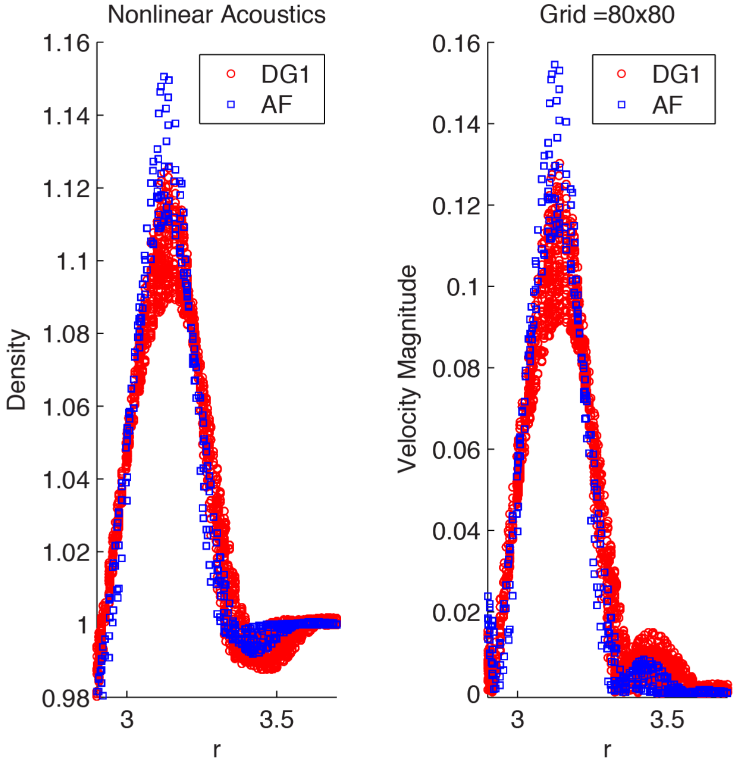


Figure 3.9: Close-up of AF and DG1 solutions for Gaussian pulse at $t = 3.0$.

3.2.2 Resolving shock waves

As this problem is run for a longer time, a shock wave starts to form as the compression wave moves out and steepens. Figure 3.8 compares the AF and DG1 results at $t = 3.0$ just after the shock has formed. We notice that there is increased scatter in the results of both methods as shown in the closeup plot in Figure 3.9. The reason for this decrease in radial symmetry is that with the chosen structured grid, the grid resolution varies in different directions, which means that some regions of the shock are better resolved than others. However, even with the increased scatter, the AF method produces a more defined shock than DG1. The closeup plot also confirms that the peak density value from the AF method, which ranges from $1.11 - 1.15$, is a better estimate of the exact value of 1.15 than the DG1 method whose peak density ranges from $1.09 - 1.13$. These results support the conclusion that the AF method is able to represent shocks on under-resolved grids with less scatter and better predictions of the extrema than DG1.

3.3 Open boundary results

Recall that in our brief discussion on open boundaries in Section 2.5.1, we claimed that the boundary condition is to simply evaluate the spherical means integral over the portion of the disc that is inside the domain and disregard the portion that is outside. We will now demonstrate the effectiveness of this simple open boundary technique. The data is initialized using the Gaussian pulse described by Equation (3.17) and the AF scheme is run until $t = 3.5$. We perform the calculation twice: once on the rectangular subdomain with its vertical boundaries treated as open boundaries and handled as described above, and again on the square domain where these boundaries are not present, as illustrated by the left picture in Figure 3.10. The boundary errors are then computed by comparing the two solutions at the vertical boundary. The maximum error norm relative to the solution at the open boundaries for two grids of different sizes – coarse and fine – are shown in Figure 3.11.

We first observe that the errors increase as the exit angle deviates from normal with a maximum density error of approximately 1% on the coarse grid and 1.6% on the fine grid at the maximum exit angle of -55 degrees. These results show that while the errors do increase from a coarse to a fine mesh, the boundary solutions are only weakly dependent on the grid size. In addition, the density and u-velocity errors essentially stay constant at minimum error values of 0.15% and 0.2% between exit angles of -25 and 25 degrees. We also observe that the results are not quite symmetric about the normal angle, which is

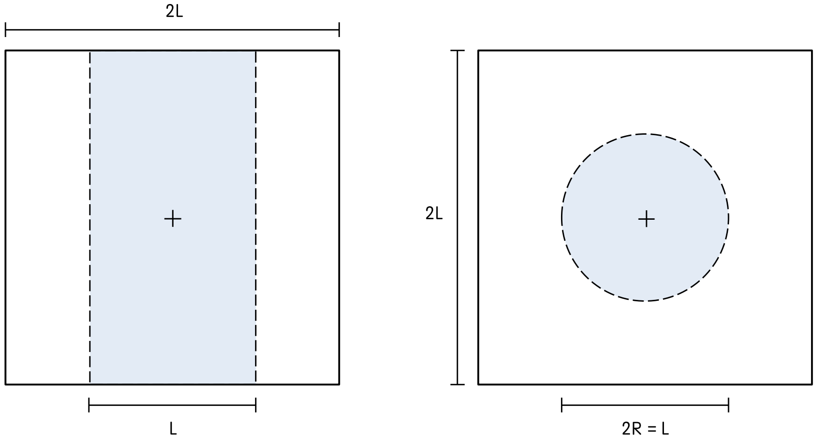
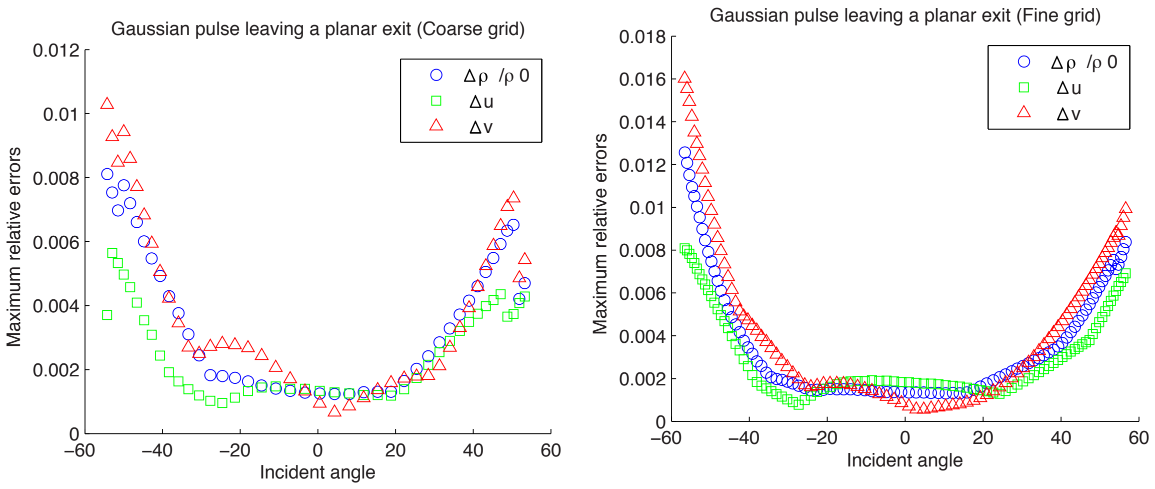


Figure 3.10: Sub-domains used in testing open boundaries.

especially evident in the v-velocity errors. This can be explained by considering the grids used were comprised of elements whose diagonals were all aligned in one direction, which means that the solutions computed along the diagonals were solved at a coarser resolution than in other directions and would result in asymmetry at the boundary.

Looking at the density distribution at an intermediate time of $t = 3.0$ as shown in Figure 3.12, we can see that the solution is no longer radially symmetric near the boundaries. This suggests that the boundary errors are causing small waves to be reflected back into the domain and interfere with the interior solution. In order to determine the magnitude of the reflected waves, we reran the Gaussian pulse problem on a circular sub-domain with 2774 elements, as illustrated by the right picture in Figure 3.10. This will ensure that the reflected waves are all of the same strength since the exit angle is now zero all along the boundary. The reference solution has been kept on the same larger square domain. Note that this is a rather severe test because the reflected waves will focus to a point as they approach the origin.

Figure 3.13 uses arrows to show that a very small reflected wave has already developed at $t = 2.5$ and at $t = 3$, this wave is traveling towards the center of the domain (see Figure 3.14). A closeup of the reflected wave is shown in Figure 3.15, from which solution errors can be computed using the reference solution. Judging by Figure 3.16, a generous estimation of the maximum amplitude of error is 0.3% for both density and velocity. This interior error is larger than the boundary error by at most a factor of 2, which means that the interior errors will stay within 3% even after encountering larger exit angles.



(a) Coarse grid of 15×30.

(b) Fine grid of 30×60.

Figure 3.11: Maximum percentage of solution errors at open boundaries for Gaussian pulse.

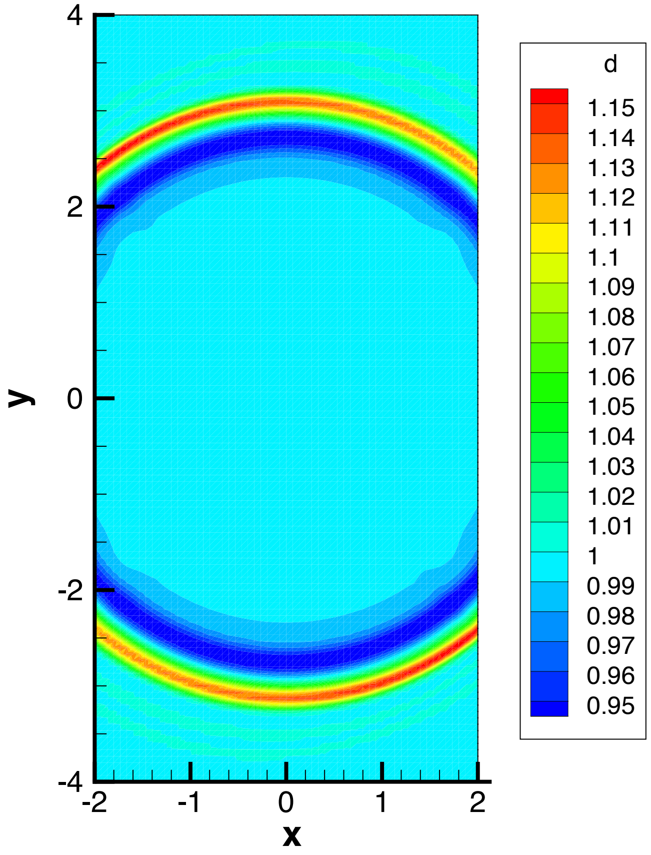


Figure 3.12: Density distribution for Gaussian pulse with open boundary condition; $t = 3.0$.

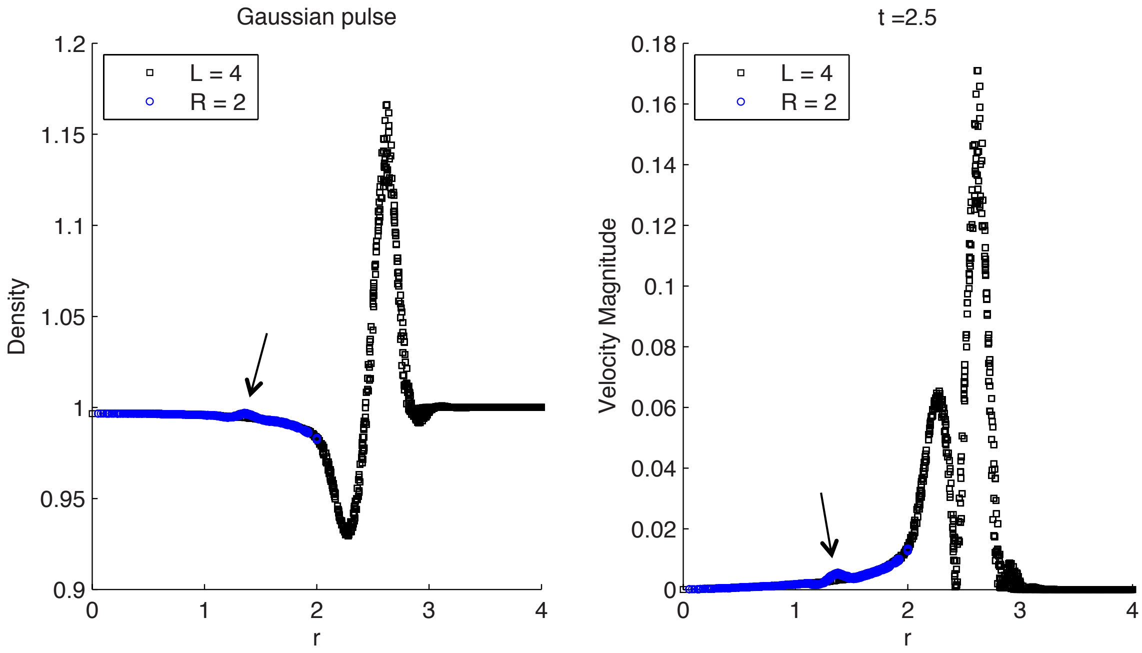


Figure 3.13: AF solutions for Gaussian pulse with open BC at $t = 2.5$ on a circular domain.

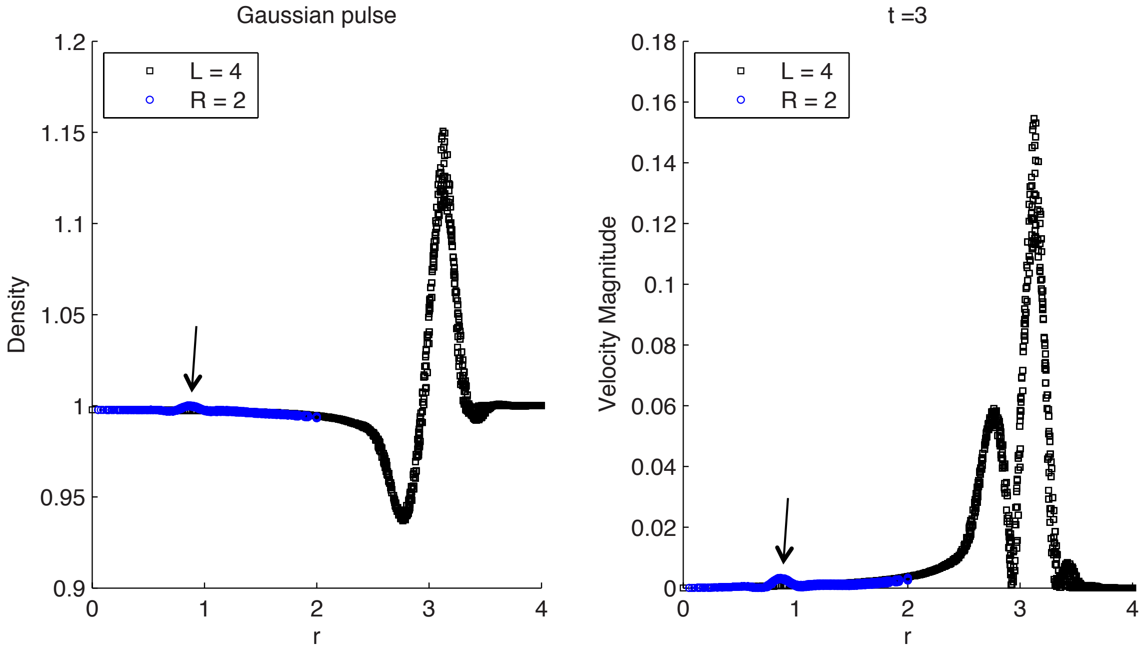


Figure 3.14: AF solutions for Gaussian pulse with open BC at $t = 3.0$ on a circular domain.

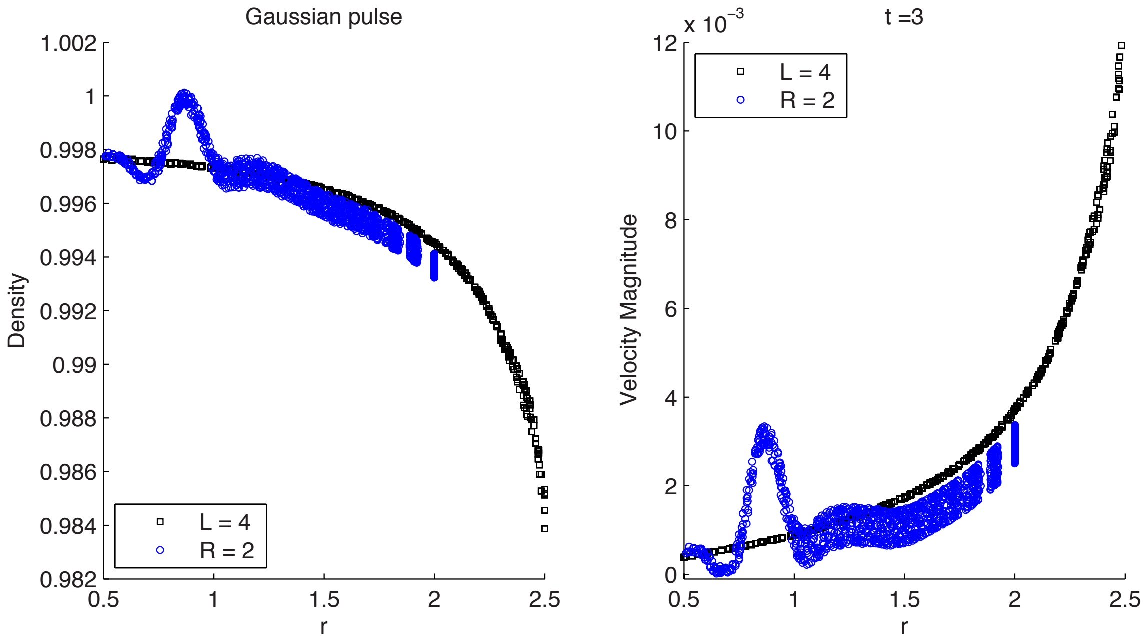


Figure 3.15: Close-up of AF solutions at $t = 3.0$.

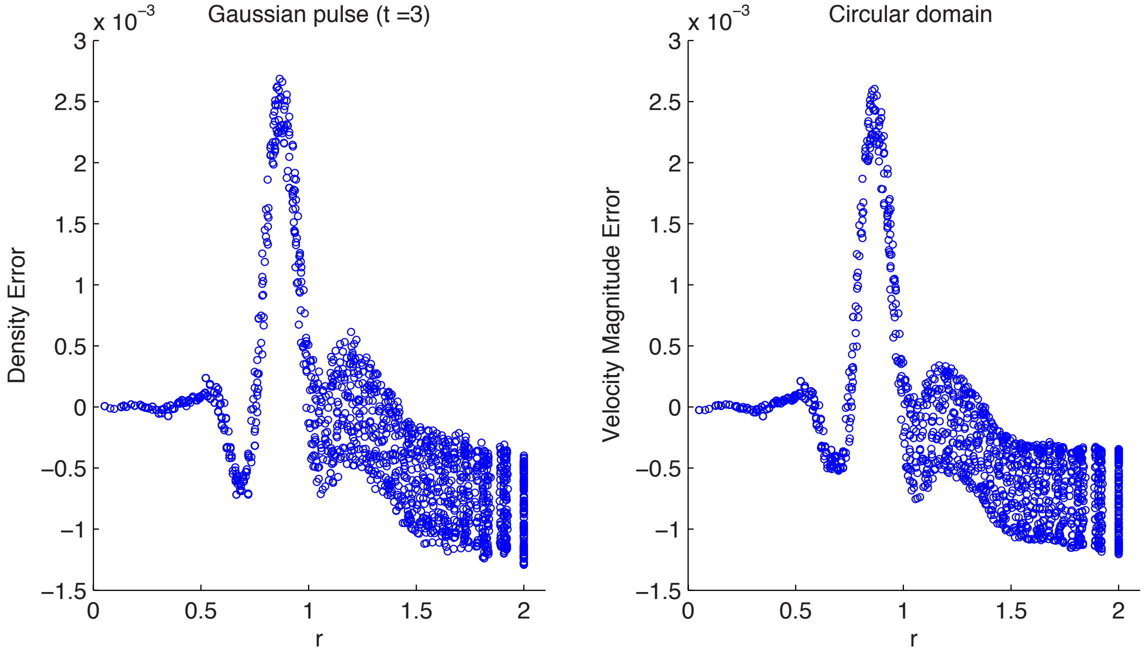


Figure 3.16: AF errors of solution with open boundary condition; $t = 3.0$.

One consequence of using this open boundary condition seems to be that there is a slight decrease in the radial symmetry of the solutions, as shown most prominently in Figure 3.15. The effect is not terrible, as this is a rather coarse grid, and we expect the scatter to decrease on a finer grid. In addition, even with this small drawback, the AF method still has the advantage that no special treatment is required for the open boundary; all that is necessary is to limit the angles of integration when computing the spherical means, and this comes naturally from the grid description. Finally, the nice behavior of the AF scheme at open boundaries would allow the use of smaller domains, hence reducing computational cost.

CHAPTER 4

Euler Equations

We have thus far seen the development of the Active Flux method for the linear and nonlinear acoustics equations. This is done as one half of our goal to develop an AF method for the multidimensional Euler equations. Our approach to solving the Euler equations is based on a splitting of advective and acoustic terms on the account of their very different numerical requirements caused by the differences in their underlying physics. On the advective side, Maeng has developed and tested the AF method to solve the multidimensional nonlinear advection equations [39]. After both halves of the Euler solver have been established independently, we then attempt to bring them together in a manner where each term in the Euler equations is solved using the appropriate advection or acoustics method. One final note is that we have opted to restrict the Euler equations to two dimensions in order to make the analysis more manageable, but the final results will be expressed in terms of operators such as ∇, $\nabla\cdot$ and $\nabla\times$ that apply in all dimensions.

We begin with the conservative Euler equations in two-dimensions

$$\partial_t \mathbf{u} + \partial_x \mathbf{f} + \partial_y \mathbf{g} = \mathbf{0} \tag{4.1}$$

where

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho H u \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho H v \end{pmatrix} \tag{4.2}$$

and E is the specific total energy and H the specific total enthalpy.

$$E = \frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{1}{2}|\mathbf{v}|^2 \tag{4.3}$$

$$H = E + \frac{p}{\rho} \tag{4.4}$$

Equation (4.1) is used in the last step of the AF method where the fluxes at interfaces are computed and used to update the cell average state. The evolution of the nodal point values, however, uses the primitive form of the Euler equations.

$$\begin{array}{lllll}
\rho_t + & \mathbf{v}\cdot\nabla\rho & + & \rho\nabla\cdot\mathbf{v} & = 0 \\
u_t + & \mathbf{v}\cdot\nabla u & + & \frac{1}{\rho}p_x & = 0 \\
v_t + & \mathbf{v}\cdot\nabla v & + & \frac{1}{\rho}p_y & = 0 \\
p_t + & \mathbf{v}\cdot\nabla p & + & \rho c^2\nabla\cdot\mathbf{v} & = 0
\end{array} \tag{4.5}$$

In this form, we are able to clearly identify that the terms in the orange box are advected by a velocity $\mathbf{v} = (u, v)^T$, and the terms in the blue box are acoustic in nature.

At this point, we would like to mention that the idea of operator splitting itself is far from original. The most common approach, for example in [40, 41, 42], has been to write the flux as $\mathbf{F} = \mathbf{F}_{ad} + \mathbf{F}_{ac}$, in one dimension, with

$$\mathbf{F}_{ad} = \begin{pmatrix} \rho u \\ \rho u^2 \\ uE \end{pmatrix}, \qquad \mathbf{F}_{ac} = \begin{pmatrix} 0 \\ p \\ up \end{pmatrix}$$

Although this makes intuitive sense, the Jacobian matrices $\mathbf{A}_{ad}$ and $\mathbf{A}_{ac}$ have eigenstructures that are not closely related to the physics. We have the advective Jacobian

$$\mathbf{A}_{ad} = \frac{\partial \mathbf{F}_{ad}}{\partial \mathbf{u}} = \begin{pmatrix} 0 & 1 & 0 \\ -u^2 & 2u & 0 \\ -uE/\rho & E/\rho & u \end{pmatrix}$$

which is nilpotent with three eigenvalues all equal to u, and two regular eigenvectors

$(0,0,1)^T$ and $(1,u,0)^T$. The acoustic Jacobian, on the other hand, is

$$\mathbf{A}_{ac} = \frac{\partial \mathbf{F}_{ac}}{\partial \mathbf{u}} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2}(\gamma-1)u^2 & -(\gamma-1)u & \gamma-1 \\ -\frac{pu}{\rho} + \frac{\gamma-1}{2}u^2 & \frac{p}{\rho} - (\gamma-1)u^2 & (\gamma-1)u \end{pmatrix}$$

whose eigenvalues are $\left(\pm\sqrt{\frac{(\gamma-1)p}{\rho}}, 0\right)$ with eigenvectors $\left(0, \frac{(\gamma-1)\pm\sqrt{(\gamma-1)p\rho}}{(\gamma-1)\rho u^2-p}, 1\right)^T$ and $(1,u,\frac{1}{2}u^2)^T$. This splitting gets some things correct. A purely advective disturbance – one that is proportional to $(1,u,\frac{1}{2}u^2)$ – does not lead to an acoustic response and will propagate at the correct speed. However, the acoustic wavespeed is poorly defined, and acoustic waves do not influence the density.

In fact, there is no satisfactory splitting of the conservative fluxes into advective and acoustic components. The problem lies with the term in the energy equation $\nabla \cdot (p\mathbf{v})$. This splits into two nonconservative terms $\mathbf{v}\cdot\nabla p$ and $p\nabla\cdot\mathbf{v}$. The first part of these is an advective term, and the second describes an acoustic effect. Hence, the splitting cannot result in a conservative formulation. It is for this reason that our method employs a nonconservative step to evaluate the fluxes.

This chapter describes in detail the process and results of applying the second-order Lax-Wendroff method to the primitive Euler system described by Equation (4.5) to update the nodal point values. Every acoustic term is evolved using the method of spherical means that was developed for the nonlinear acoustic equations in Chapter 3. Similarly, every advective term is updated using the method of characteristics, details of which is not included in this work but is the main topic of Brad Maeng's thesis [39]. We then introduce a new approach to achieve third-order convergence without using the bubble function, which removes the need for third-order terms in the data reconstruction inside each element, but also reduces the compactness of the method. Finally, AF results are compared to DG1 and DG2 for the moving vortex problem in terms of accuracy, convergence, and computational time.

4.1 Lax-Wendroff approximation

Let the primitive state vector be $\mathbf{q} = (\rho, u, v, p)^T$. We can rewrite the primitive Euler equations as

$$\partial_t \mathbf{q} + \boldsymbol{A}\partial_x \mathbf{q} + \boldsymbol{B}\partial_y \mathbf{q} = \mathbf{0} \tag{4.6}$$

where

$$\boldsymbol{A} = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & \dfrac{1}{\rho} \\ 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & u \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & \dfrac{1}{\rho} \\ 0 & 0 & \gamma p & v \end{pmatrix} \tag{4.7}$$

so that the second time derivatives are

$$\partial_{tt}\mathbf{q} + \partial_t\boldsymbol{A}\partial_x\mathbf{q} + \boldsymbol{A}\partial_{xt}\mathbf{q} + \partial_t\boldsymbol{B}\partial_y\mathbf{q} + \boldsymbol{B}\partial_{yt}\mathbf{q} = \mathbf{0} \tag{4.8}$$

The Lax-Wendroff method can then be approximated to second-order accuracy using Equations (4.6) and (4.8).

$$\begin{aligned}
\mathbf{q}^{n+1} &= \mathbf{q}^n + t\partial_t\mathbf{q}^n + \frac{1}{2}t^2\partial_{tt}\mathbf{q}^n \\
&= \mathbf{q}^n - t(\boldsymbol{A}\partial_x\mathbf{q} + \boldsymbol{B}\partial_y\mathbf{q}) - \frac{1}{2}t^2(\partial_t\boldsymbol{A}\partial_x\mathbf{q} + \boldsymbol{A}\partial_{xt}\mathbf{q} + \partial_t\boldsymbol{B}\partial_y\mathbf{q} + \boldsymbol{B}\partial_{yt}\mathbf{q}) \\
&= \mathbf{q}^n - t\left(\boldsymbol{A} + \frac{1}{2}t\partial_t\boldsymbol{A}\right)\partial_x\mathbf{q} - t\left(\boldsymbol{B} + \frac{1}{2}t\partial_t\boldsymbol{B}\right)\partial_y\mathbf{q} \\
&\qquad\qquad + \frac{1}{2}t^2\boldsymbol{A}\partial_x(\boldsymbol{A}\partial_x\mathbf{q} + \boldsymbol{B}\partial_y\mathbf{q}) + \frac{1}{2}t^2\boldsymbol{B}\partial_y(\boldsymbol{A}\partial_x\mathbf{q} + \boldsymbol{B}\partial_y\mathbf{q}) \\
&= \mathbf{q}^n - t\left(\boldsymbol{A} + \frac{1}{2}t(\partial_t\boldsymbol{A} - \boldsymbol{A}\partial_x\boldsymbol{A} - \boldsymbol{B}\partial_y\boldsymbol{A})\right)\partial_x\mathbf{q} \\
&\qquad - t\left(\boldsymbol{B} + \frac{1}{2}t(\partial_t\boldsymbol{B} - \boldsymbol{A}\partial_x\boldsymbol{B} - \boldsymbol{B}\partial_y\boldsymbol{B})\right)\partial_y\mathbf{q} \\
&\qquad\qquad + \frac{1}{2}t^2(\boldsymbol{A}^2\partial_{xx}\mathbf{q} + (\boldsymbol{B}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{B})\partial_{xy}\mathbf{q} + \boldsymbol{B}^2\partial_{yy}\mathbf{q}) \\
&= \mathbf{q}^n - t(\boldsymbol{A}^*\partial_x\mathbf{q} + \boldsymbol{B}^*\partial_y\mathbf{q}) \\
&\qquad + \frac{1}{2}t^2(\boldsymbol{A}^2\partial_{xx}\mathbf{q} + (\boldsymbol{B}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{B})\partial_{xy}\mathbf{q} + \boldsymbol{B}^2\partial_{yy}\mathbf{q})
\end{aligned} \tag{4.9}$$

where the modified coefficient matrices are

$$\mathbf{A}^* = \mathbf{A} + \boxed{\frac{1}{2}t(\mathbf{A}_t - \mathbf{A}\mathbf{A}_x - \mathbf{B}\mathbf{A}_y)} \tag{4.10}$$

$$\mathbf{B}^* = \mathbf{B} + \boxed{\frac{1}{2}t(\mathbf{B}_t - \mathbf{A}\mathbf{B}_x - \mathbf{B}\mathbf{B}_y)} \tag{4.11}$$

The second line in the final Equation (4.9) simply contains the second-order terms of the solution to the linearized Euler equations. The first line consists of the first-order part of the solution with coefficient matrices that have been modified to include the nonlinear terms in the boxes shown in Equations (4.10) and (4.11). In the special case of linearized Euler equations, the terms in the boxes vanish, and the advective and acoustic terms can be evolved separately using operator splitting. The challenge now is to determine which process, either advective or acoustic, each second-order nonlinear term is associated with in the full Euler equations.

Before moving onto the discussion of the nonlinear terms, we would like to mention that there are a few different ways to perform operator splitting to the primitive Euler equations, all of which are discussed by Maeng [39]. In this work, we focus solely on the nonsymmetric operator splitting that applies the advective operator first, followed by the acoustic operator. The reason behind choosing this particular operator splitting is based on the intuitive application of both advective and acoustic conditions at the boundaries, which will be discussed further in Chapter 5.

4.1.1 Advective terms

Including only the advective terms reduces Equation (4.5) to

$$\begin{aligned} \rho_t + \mathbf{v} \cdot \nabla \rho &= 0 \\ u_t + \mathbf{v} \cdot \nabla u &= 0 \\ v_t + \mathbf{v} \cdot \nabla v &= 0 \\ p_t + \mathbf{v} \cdot \nabla p &= 0 \end{aligned} \tag{4.12}$$

The second-order advective solution can be determined at

$$\mathbf{q}(\mathbf{0}, t) = \mathbf{q}(\mathbf{x}^{**}, 0) \tag{4.13}$$

where the modified characteristic origin is defined as

$$\begin{aligned} \mathbf{x}^{**} &= \mathbf{x}^{*} + t^2 \frac{1}{2\rho} \nabla p \\ &= -t(\mathbf{v}\mathbf{I} - t\mathbf{v} \cdot \nabla \mathbf{v}) + t^2 \frac{1}{2\rho} \nabla p \end{aligned} \tag{4.14}$$

Applying the nonlinear advective solution to the Lax-Wendroff approximation of the Euler equations results in the following update.

$$\begin{aligned}\mathbf{q}^{n+1} = \mathbf{q}^n &- t\left\{\left(\mathbf{A} - t(\mathbf{v}\cdot\nabla u + \tfrac{1}{2\rho}p_x)\mathbf{I}\right)\partial_x\mathbf{q} + \left(\mathbf{B} - t(\mathbf{v}\cdot\nabla v + \tfrac{1}{2\rho}p_y)\mathbf{I}\right)\partial_y\mathbf{q}\right\}\\ &+ \frac{1}{2}t^2\left\{\mathbf{A}^2\partial_{xx}\mathbf{q} + (\mathbf{AB}+\mathbf{BA})\partial_{xy}\mathbf{q} + \mathbf{B}^2\partial_{yy}\mathbf{q}\right\}\\ &+ t^2(\mathbf{A}^{***}\partial_x\mathbf{q} + \mathbf{B}^{***}\partial_y\mathbf{q})\end{aligned} \tag{4.15}$$

The first line of Equation (4.15) describes the second-order solution with nonlinear advection and linear acoustics. The second line remains unchanged, and third line contains the nonlinear terms left to evaluate, which can be simplified using the isentropic assumption $\rho\,\mathrm{d}(c^2) = (\gamma-1)\,\mathrm{d}p$ to the following terms.

$$\mathbf{A}^{***}\partial_x\mathbf{q} + \mathbf{B}^{***}\partial_y\mathbf{q} =$$
$$\begin{pmatrix} \frac{1}{2}\rho\left(u_x^2 + v_y^2 + 2u_yv_x\right) - \frac{1}{2\rho}\nabla\rho\cdot\nabla p + (\nabla\cdot\mathbf{v})(\mathbf{v}\cdot\nabla\rho) \\ \frac{1}{2\rho}\nabla\cdot\mathbf{v}(\rho c^2)_x - \frac{1}{2\rho}(p_xv_y - p_yv_x) + \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right)p_x \\ \frac{1}{2\rho}\nabla\cdot\mathbf{v}(\rho c^2)_y - \frac{1}{2\rho}(p_xu_y - p_yu_x) + \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right)p_y \\ \frac{1}{2}\rho c^2\left(u_x^2 + v_y^2 + 2u_yv_x\right) - \frac{c^2}{2\rho}\nabla\rho\cdot\nabla p + (\nabla\cdot\mathbf{v})(\mathbf{v}\cdot\nabla(\rho c^2)) + (\gamma-1)\rho c^2(\nabla\cdot\mathbf{v})^2 \end{pmatrix} \tag{4.16}$$

Details on the derivation of Equations (4.14) and (4.15) can be found in Appendix B.

4.1.2 Acoustic terms

Taking the Euler equations to include only the acoustic terms reduces Equation (4.5) to

$$\begin{aligned}\rho_t + \rho\nabla\cdot\mathbf{v} &= 0\\ u_t + \frac{1}{\rho}p_x &= 0\\ v_t + \frac{1}{\rho}p_y &= 0\\ p_t + \rho c^2\nabla\cdot\mathbf{v} &= 0\end{aligned} \tag{4.17}$$

It is important to remember that the operator splitting method requires the second operator to be applied to the intermediate solution computed from the first operator. This results in some nonlinear second-order terms $\Delta\mathbf{q}_{os}$ being produced.

$$\begin{aligned}
\Delta\rho_{os} &= \rho(u_x^2 + v_y^2 + 2u_y v_x) \\
\Delta u_{os} &= \frac{1}{\rho}\left(p_x u_x + p_y v_x\right) \\
\Delta v_{os} &= \frac{1}{\rho}\left(p_x u_y + p_y v_y\right) \\
\Delta p_{os} &= \rho c^2\left(u_x^2 + v_y^2 + 2u_y v_x\right)
\end{aligned} \tag{4.18}$$

In addition, the second derivatives with respect to time of Equation (4.17) are

$$\begin{aligned}
\rho_{tt} &= \nabla^2 p - \frac{1}{\rho}\nabla\rho\cdot\nabla p + \rho(\nabla\cdot\mathbf{v})^2 \\
u_{tt} &= c^2(\nabla\cdot\mathbf{v})_x + \frac{1}{\rho}\nabla\cdot\mathbf{v}(\rho c^2)_x - \frac{1}{\rho}(\nabla\cdot\mathbf{v})p_x \\
v_{tt} &= c^2(\nabla\cdot\mathbf{v})_y + \frac{1}{\rho}\nabla\cdot\mathbf{v}(\rho c^2)_y - \frac{1}{\rho}(\nabla\cdot\mathbf{v})p_y \\
p_{tt} &= c^2\nabla^2 p - \frac{c^2}{\rho}\nabla\rho\cdot\nabla p + \gamma\rho c^2(\nabla\cdot\mathbf{v})^2
\end{aligned} \tag{4.19}$$

Recall that the Lax-Wendroff approximation is of the form

$$\mathbf{q}^{n+1} = \mathbf{q}^n + t\partial_t\mathbf{q} + \frac{1}{2}t^2\partial_{tt}\mathbf{q}$$

and thus many of the second-order terms in Equation (4.19) that show up in Equation (4.16) are purely acoustic in nature.

After removing the second-order terms in Equations (4.18) and (4.19) from Equation

(4.16), the remaining terms are

$$\begin{pmatrix} (\nabla\cdot\mathbf{v})(\mathbf{v}\cdot\rho) \\ \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right)p_x \\ \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right)p_y \\ (\nabla\cdot\mathbf{v})(\mathbf{v}\cdot(\rho c^2)) \end{pmatrix} - \begin{pmatrix} \frac{1}{2}\rho\left(u_x^2+v_y^2+2u_yv_x\right) \\ \frac{1}{2\rho}(p_xu_x+p_yv_x) \\ \frac{1}{2\rho}(p_xu_y+p_yv_y) \\ \frac{1}{2}\rho c^2\left(u_x^2+v_y^2+2u_yv_x\right) \end{pmatrix} \tag{4.20}$$

Note that the terms in the first matrix in Equation (4.20) are simply the first-order terms of the acoustic equations evaluated at the linear characteristic origin of $\mathbf{x}^{lin} = -t\mathbf{v}$. To reduce computational cost, we prefer to use the nonlinear characteristic origin $\mathbf{x}^{**}$ defined in Equation (4.14) since it includes the appropriate first-order terms and is readily available.

We have previously noted the similarity of the Active Flux acoustics scheme to the Lax-Wendroff method in Section 2.1. Indeed, we can recover the solution to the acoustic problem in Equation (4.17) by using the method of spherical means and a sound speed of $c^2 = \partial p/\partial\rho$.

$$\mathbf{q}^{n+1} = \mathbf{q}^{**} + tM_R\{\partial_t\mathbf{q}^{**}\} + \frac{1}{c^2}\int_0^R rM_r\{\partial_{tt}\mathbf{q}^{**}\}\,\mathrm{d}r \tag{4.21}$$

The superscript ** refers to the intermediate solution to the advection problem evaluated at the characteristic origin $\mathbf{x}^{**}$. Note that the time-derivatives in Equation (4.21) use only the acoustic terms in Equations (4.17) and (4.19). After some simplifications, the evaluation of the acoustic-driven terms become

$$\begin{aligned}\rho^{n+1} = \rho^{**} &- t\rho^{**}M_R\{\nabla\cdot\mathbf{v}^{**}\} + \frac{1}{c^2}\int_0^R rM_r\{\nabla^2 p\}\,\mathrm{d}r \\ &+ \frac{t^2}{2}\rho\left(M_R\{\nabla\cdot\mathbf{v}\}\right)^2 - \frac{t^2}{2}\frac{1}{\rho}M_R\{\nabla\rho\}\cdot M_R\{\nabla p\}\end{aligned} \tag{4.22a}$$

$$\begin{aligned}\mathbf{v}^{n+1} = \mathbf{v}^{**} &- t\frac{1}{\rho^{**}}M_R\{\nabla p^{**}\} + \int_0^R rM_r\{\nabla^2\mathbf{v}\}\,\mathrm{d}r + \int_0^R rM_r\{\nabla\times\nabla\times\mathbf{v}\}\,\mathrm{d}r \\ &+ \frac{t^2}{2}M_R\{\nabla\cdot\mathbf{v}\}M_R\{\nabla(c^2)\} + \frac{t^2}{2}\frac{1}{\rho}M_R\{\nabla\cdot\mathbf{v}\}\left(c^2M_R\{\nabla\rho\} - M_R\{\nabla p\}\right)\end{aligned} \tag{4.22b}$$

$$p^{n+1} = p^{**} \ - t\rho^{**}c^{**2}M_R\{\nabla \cdot \mathbf{v}^{**}\} + \int\limits_0^R rM_r\{\nabla^2 p\}\,\mathrm{d}r$$
$$+ \frac{t^2}{2}\gamma\rho c^2\left(M_R\{\nabla \cdot \mathbf{v}\}\right)^2 - \frac{t^2}{2}\frac{c^2}{\rho}M_R\{\nabla\rho\} \cdot M_R\{\nabla p\} \tag{4.22c}$$

Using the same observations that were made to the nonlinear acoustic equations, the first lines in Equation (4.22) describe the exact solution to the linear acoustics problem, and the second lines contain the second-order nonlinear acoustic terms. The data used to evaluate the second-order terms need only be accurate to the leading (constant) term, which means that we can use either the data at the current timestep n or the intermediate solution after the advective operator has been applied. Another observation of interest is that the last two terms in Equation (4.22b) would cancel each other if the flow is fully isentropic, which could lend to further simplifications.

The remaining terms in the second matrix of Equation (4.20) describe the interaction between the advective and acoustic processes. Writing them in another form,

$$\begin{aligned}
\Delta\rho_{in} &= -\frac{1}{2}\rho\left((\nabla \cdot \mathbf{v})^2 + (u_y v_x - u_x v_y)\right) \\
\Delta u_{in} &= -\frac{1}{2\rho}\left(\nabla p \cdot \nabla u + p_y(v_x - u_y)\right) \\
\Delta v_{in} &= -\frac{1}{2\rho}\left(\nabla p \cdot \nabla v - p_x(v_x - u_y)\right) \\
\Delta p_{in} &= -\frac{1}{2}\rho c^2\left((\nabla \cdot \mathbf{v})^2 + (u_y v_x - u_x v_y)\right)
\end{aligned} \tag{4.23}$$

we can interpret these interaction terms to represent some physical behavior. For example, the density and pressure interaction terms describe the narrowing and twisting of the streamtube due to the velocity field. The process described by the velocity interaction terms is less clear, but it seems to involve some gyroscopic effect caused by the interaction between the pressure and velocity fields.

These interaction terms can theoretically be computed using either point values (advective method) or spherical means (acoustic method). Preferably, the spherical means value is used because the solution gradients are generally discontinuous at nodal points, so by computing the spherical means integral, we are essentially averaging the value of the gradients at each node.

4.1.3 Active Flux algorithm – Part 1

Let the nodal point update be a two-step method,

$$\mathbf{q}^{**} = \mathbf{q}^{n} + \Delta\mathbf{q}_{ad} \tag{4.24}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{**} + \Delta\mathbf{q}_{ac} + \Delta\mathbf{q}_{in} \tag{4.25}$$

which can be implemented using the following algorithm applied at the half and full timesteps $n+1/2$ and n for every vertex and edge node. Note that the equations have been discretized in such a way that if the Euler code were applied to the linear acoustic system, the method of spherical means would be recovered.

1. Apply nonlinear advection

$$\Delta\mathbf{q}_{ad} = \mathbf{q}^{n}(\mathbf{x}^{**}) - \mathbf{q}^{n}(\mathbf{0}) \tag{4.26}$$

where the modified characteristic with pressure gradient is

$$\mathbf{x}^{**} = \mathbf{x}^{*} + \frac{t^2}{2\rho}\nabla p = -t\mathbf{v} + t^2\left(\mathbf{v}\cdot\nabla\mathbf{v} + \frac{1}{2\rho}\nabla p\right)$$

2. Apply nonlinear acoustics with corrected first-order terms

$$\begin{aligned}\Delta\rho_{ac} = {}& -t\rho^{**}M_R\{\nabla\cdot\mathbf{v}^{**}\} + \frac{1}{c^2}\int\limits_0^R rM_r\{\nabla^2 p\}\,\mathrm{d}r \\ & + \frac{t^2}{2}\rho\left(M_R\{\nabla\cdot\mathbf{v}\}\right)^2 - \frac{t^2}{2}\frac{1}{\rho}M_R\{\nabla\rho\}\cdot M_R\{\nabla p\}\end{aligned} \tag{4.27}$$

$$\begin{aligned}\Delta\mathbf{v}_{ac} = {}& -t\frac{1}{\rho^{**}}M_R\{\nabla p^{**}\} + \int\limits_0^R rM_r\{\nabla^2\mathbf{v}\}\,\mathrm{d}r + \int\limits_0^R rM_r\{\nabla\times\nabla\times\mathbf{v}\}\,\mathrm{d}r \\ & + \frac{t^2}{2}M_R\{\nabla\cdot\mathbf{v}\}M_R\{\nabla(c^2)\} + \frac{t^2}{2}\frac{1}{\rho}M_R\{\nabla\cdot\mathbf{v}\}\left(c^2M_R\{\nabla\rho\} - M_R\{\nabla p\}\right)\end{aligned} \tag{4.28}$$

$$\begin{aligned}\Delta p_{ac} = {}& -t\rho^{**}c^{**2}M_R\{\nabla\cdot\mathbf{v}^{**}\} + \int\limits_0^R rM_r\{\nabla^2 p\}\,\mathrm{d}r \\ & + \frac{t^2}{2}\gamma\rho c^2\left(M_R\{\nabla\cdot\mathbf{v}\}\right)^2 - \frac{t^2}{2}\frac{c^2}{\rho}M_R\{\nabla\rho\}\cdot M_R\{\nabla p\}\end{aligned} \tag{4.29}$$

3. Include the remaining interaction terms using spherical means.

$$\Delta\mathbf{q}_{in} = -\frac{t^2}{2}\begin{pmatrix} \rho\left(M_R\{u_x\}^2 + M_R\{v_y\}^2 + 2M_R\{u_y\}M_R\{v_x\}\right) \\ \\ M_R\{p_x\}M_R\{u_x\} + M_R\{p_y\}M_R\{v_x\} \\ \\ M_R\{p_x\}M_R\{u_y\} + M_R\{p_y\}M_R\{v_y\} \\ \\ \rho c^2\left(M_R\{u_x\}^2 + M_R\{v_y\}^2 + 2M_R\{u_y\}M_R\{v_x\}\right) \end{pmatrix} \tag{4.30}$$

This concludes the second-order nodal update portion of the Active Flux method for the Euler equations. The next step in the method is to update the conservative cell averages to third-order accuracy by integrating the fluxes computed at each cell interface. The cell-average values have so far been implemented into the AF method in the form of a third-order bubble function. However, Maeng had encountered numerical difficulties with the bubble function that led to mesh alignment issues when solving multidimensional nonlinear advection problems (see [39] for details). Since we aim to only use the nonsymmetric operator splitting that applies the advective operator prior to the acoustic operator, the errors encountered with the mesh alignment issue is unfortunately large enough to prevent third-order convergence, especially on structured grids which favor certain directions more than others. Therefore, in the next section, we propose a new scheme to achieve a third-order Active Flux method without the use of a bubble function.

4.2 Obtaining third-order accuracy without bubble function

The idea behind adding a bubble function is to reconcile the third-order accurate cell-average conservative data with the second-order nodal primitive data in order to raise the overall convergence of the AF scheme to three. While the use of a bubble function is an elegant and compact approach to achieving this reconciliation, it is by no means the only way. In fact, we can also raise the order of accuracy of the nodal values by taking into account the differences between the cell-average values updated using fluxes, which is third-order accurate, and reconstructed using nodal values, which is only second-order accurate. These reconciliation differences are local to each element, which means that they will need to be averaged at each node depending on the number of elements that share the said node. This reconciliation method has the advantage that there is longer a need to evaluate any of the expensive third-order terms. However, some compactness is lost because we now need to share information about the local reconciliation differences across multiple nodes.

This section will discuss the new reconciliation method in detail, starting with the change of variables from primitive state to conservative and vice versa, and ending with the application of the method in the AF framework.

4.2.1 Change of variables

Let the conservative state vector be $\mathbf{u} = (\rho, \rho u, \rho v, \rho E)^T$ and the primitive state vector again be $\mathbf{q} = (\rho, u, v, p)^T$. The two sets of variables can be transformed into each other as follows:

$$\mathbf{u} = \boldsymbol{W_u}\mathbf{q} \tag{4.31a}$$

$$\mathbf{q} = \boldsymbol{W_q}\mathbf{u} \tag{4.31b}$$

where the transformation matrices are given by

$$\begin{gathered}\boldsymbol{W_u} = \frac{\partial \mathbf{u}}{\partial \mathbf{q}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \rho & 0 & 0 \\ 0 & 0 & \rho & 0 \\ 0 & \frac{1}{2}\rho u & \frac{1}{2}\rho v & \frac{1}{\gamma - 1} \end{pmatrix} \\ \boldsymbol{W_q} = \boldsymbol{W_u}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & -\frac{\gamma - 1}{2}u & -\frac{\gamma - 1}{2}v & \gamma - 1 \end{pmatrix}\end{gathered} \tag{4.32}$$

Another change of variables that may be required is the conversion of the changes in the cell-average conservative variables to the primitive state variables while maintaining third-order accuracy. As illustrated by the transformation matrix in Equation (4.32), with the exception of density, the conservative states are often a product of two or more primitive states. Hence, changes in the conservative cell average $\Delta\overline{\mathbf{u}} = (\Delta\overline{\rho}, \Delta\overline{\rho u}, \Delta\overline{\rho v}, \Delta\overline{\rho E})^T$ cannot be transformed directly to changes in the primitive cell average $\Delta\overline{\mathbf{q}} = (\Delta\overline{\rho}, \Delta\overline{u}, \Delta\overline{v}, \Delta\overline{p})^T$ using Equation (4.31b). Instead, given the evolution of the conservative cell-average values as

$$\overline{u}^{n+1} = \overline{u}^n + \Delta\overline{u}$$

the transformation must satisfy the following expression:

$$\Delta\overline{\mathbf{q}} = \overline{\boldsymbol{W}}_{\boldsymbol{q}}\,\Delta\overline{\mathbf{u}} \tag{4.33}$$

where the average values are taken from the initial data.

$$\overline{\boldsymbol{W}}_{\boldsymbol{q}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\dfrac{\overline{\rho u}}{\overline{\rho}^2} & \dfrac{1}{\overline{\rho}} & 0 & 0 \\ -\dfrac{\overline{\rho v}}{\overline{\rho}^2} & 0 & \dfrac{1}{\overline{\rho}} & 0 \\ \dfrac{\gamma-1}{2}\left(\dfrac{\overline{\rho u}+\overline{\rho v}}{\overline{\rho}}\right)^2 & -(\gamma-1)\dfrac{\overline{\rho u}}{\overline{\rho}} & -(\gamma-1)\dfrac{\overline{\rho v}}{\overline{\rho}} & \gamma-1 \end{pmatrix} \tag{4.34}$$

To obtain Equations (4.33) and (4.34), we have made the assumption that the changes of the cell-average values are small, which is generally true in a fully explicit method like Active Flux. We now have the necessary tools to reconcile the cell-average conservative data with the nodal data in the AF scheme.

4.2.2 A new approach to reconciliation

The naïve approach to increase the AF solution to third-order accuracy is to simply reconcile, using a constant correction term k_{corr}, the differences between the cell-average conservative data $\overline{\mathbf{u}} = (\overline{\rho}, \overline{\rho u}, \overline{\rho v}, \overline{\rho E})$ and the reconstructed cell-average value computed by integrating the point-wise primitive data $\mathbf{q} = (\rho, u, v, p)$,

$$\begin{aligned} \overline{\mathbf{u}} &= \mathbf{u}_{avg} + k_{corr} \\ &= \frac{1}{\Omega_j}\int_{\Omega_j} \boldsymbol{W}_{\boldsymbol{u}}\mathbf{q}_j(x,y)\,d\Omega + k_{corr} \end{aligned} \tag{4.35}$$

where Ω_j is the elemental area, and $\mathbf{q}_j$ is the quadratic reconstruction of the point-wise primitive solution within element j. As illustrated in Figure 4.1, this correction term would

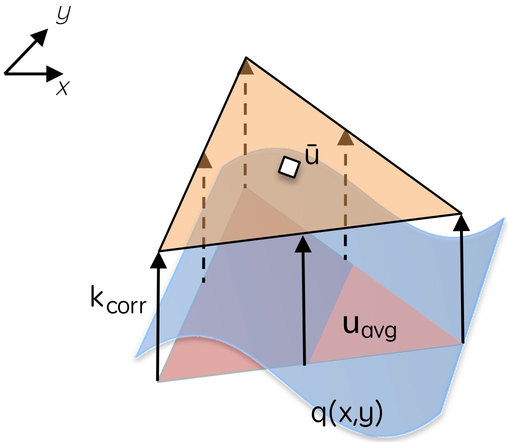


Figure 4.1: Reconciliation between the cell-average data and the reconstructed cell average values.

bring the reconstructed conservative cell average $\mathbf{u}_{avg}$ to the same value as $\overline{\mathbf{u}}$. Note that the integral in Equation (4.35) should be computed to at least third-order accuracy using Gauss-Legendre points. However, the simplest third-order accurate quadrature does not take contributions from data at the vertices [43]. This leaves us with some ambiguity about the update procedure at the vertex nodes, which we have resolved by choosing to update the vertex data by the same correction constant as for the edge data in each cell.

Unfortunately, we did not have much success in achieving clear third-order convergence with this approach, and have seen some instability issues at small Courant numbers as well. Rather, we propose an alternate approach to improve the order of accuracy of the nodal solution by looking at the residuals of the cell-average and point nodal data, not simply the solution data itself. The idea is the same as the reconciliation step described by Equation (4.35), but we will apply it to residuals instead of the conservative solutions.

For our purposes, the residual at timestep n is defined as the difference between the conservative solutions at timesteps n and $n+1$.

$$\mathbf{r}^n = \mathbf{u}^{n+1} - \mathbf{u}^n \tag{4.36}$$

For the cell-average conservative solution, the residual is third-order accurate due to the integration of interface fluxes around each element.

$$\begin{aligned}\overline{\mathbf{r}}^n &= \overline{\mathbf{u}}^{n+1} - \overline{\mathbf{u}}^n \\ &= \Delta t \int_{d\Omega_j} -(\mathbf{f}\hat{\imath} + \mathbf{g}\hat{\jmath})\, ds = \mathcal{O}(h^3)\end{aligned} \tag{4.37}$$

Here, $\mathbf{f}$ and $\mathbf{g}$ are the x- and y-directional flux vectors in Equation (4.2). By comparison, the residual of each nodal point is only second-order accurate,

$$\mathbf{r}_i^n = \mathbf{u}_i^{n+1} - \mathbf{u}_i^n = \mathcal{O}(h^2) \tag{4.38}$$

where the conservative solution is computed by applying a change of variables, $\mathbf{u}_i = \boldsymbol{W_u}\mathbf{q}_i$.

Now, we will reconcile the cell-average residual with the point-wise residual in each cell by adding a constant correction term $\mathbf{k} = (k_\rho, k_{\rho u}, k_{\rho v}, k_{\rho E})^T$ to the vertex and edge nodal values and integrating inside the element.

$$\begin{aligned}
\mathbf{k}_j &= \overline{\mathbf{r}}_j - \frac{1}{\Omega_j}\int_{\Omega_j} \mathbf{r}_j(\xi,\eta)\, d\Omega \\
&= \overline{\mathbf{r}}_j - \frac{1}{\Omega_j}\int_{\Omega_j} \left(\sum_{i=1}^{6} \mathbf{r}_{j,i}\phi_i(\xi,\eta)\right)\, d\Omega \\
&= \overline{\mathbf{r}}_j - \frac{1}{3}(\mathbf{r}_{j,2} + \mathbf{r}_{j,4} + \mathbf{r}_{j,6})
\end{aligned} \tag{4.39}$$

Here, we have assumed that the residual function inside the cell is quadratic, although we would like to note that it is most likely more accurate to reconstruct a higher-order function since the residual is in conservative form. However, in practice, we have not noticed much improvement in evaluating the integral at higher orders of accuracy, so we have opted for the minimum second-order reconstruction.

The primitive nodal solutions are then updated to third-order accuracy by again applying a change of variables in one of two ways. The first is to use the conservative point-wise data computed in Equation (4.38) at every node i of element j.

$$\mathbf{q}_{j,i} = \boldsymbol{W_q}(\mathbf{u}_{j,i} + \mathbf{k}_j) \tag{4.40}$$

The second choice is to recognize that $\mathbf{k}_j$ is relatively small in magnitude since it is the difference between two residual values that also tend to be small. Thus, we can use the transformation matrix $\overline{\boldsymbol{W}}_{\boldsymbol{q}}$ in Equation (4.34) to update the nodal point values directly.

$$\mathbf{q}_{j,i}^{n+1} = \mathbf{q}_{j,i} + \overline{\boldsymbol{W}}_{\boldsymbol{q}}\mathbf{k}_j \tag{4.41}$$

The second method may be preferred if the AF algorithm is looped over elements because there would be no need to store the conservative nodal point data, while the first method might be more efficient if the algorithm is looped over nodes.

Clearly, this method will produce different values of $\mathbf{k}$ for each element surrounding a vertex or edge node. This can be resolved by averaging all of the $\mathbf{k}_j$ values by the number of elements surrounding the node, which will keep the solution at third-order accuracy. As we will see later in Section (4.3), applying this reconciliation scheme to the AF method produces results that converge at third-order on both unstructured and structured grids, the latter showing no sign of the mesh alignment issues that we have seen with using a bubble function.

4.2.3 Active Flux algorithm – Part 2

Let the third-order update for the nodal point solution be dependent on the difference in residuals,

$$\mathbf{q}^{n+1} = \mathbf{q}_{2nd}^{n+1} + \overline{\boldsymbol{W}}_{\boldsymbol{q}} \Delta \overline{\mathbf{r}} \tag{4.42}$$

where $\mathbf{q}_{2nd}^{n+1}$ is the second-order accurate nodal solution from the first part of the AF algorithm in Section 4.1.3. This update can be implemented using the following algorithm applied at the end of a full timestep.

1. Compute residual of cell-average conservative state

   $$\overline{\mathbf{r}} = -\Delta t \int_{d\Omega_j} (\mathbf{f}\hat{\imath} + \mathbf{g}\hat{\jmath})\, ds \tag{4.43}$$

   by using the solutions at timesteps n, $n+1/2$, and $n+1$ to integrate the fluxes across each interface, as previously discussed in Section 1.3.3.

2. Compute cell-average residual from nodal solution

   $$\begin{aligned} \mathbf{r}_{avg} &= \frac{1}{\Omega_j} \int_{\Omega_j} \mathbf{r}_j(\xi, \eta)\, \mathrm{d}\Omega \\ &= \frac{1}{3}(\mathbf{r}_{j,2} + \mathbf{r}_{j,4} + \mathbf{r}_{j,6}) \end{aligned} \tag{4.44}$$

3. Determine the difference in residual values

   $$\Delta \overline{\mathbf{r}} = \frac{1}{ne} \sum_{i}^{ne} (\overline{\mathbf{r}}_i - \mathbf{r}_{i,avg}) \tag{4.45}$$

   where ne is the number of elements that share the given node.

The first two steps are computed over all elements and the last step is finished over all nodes. This algorithm is easy to implement and rather compact as it does not require extensive communication between neighboring elements. In conjunction with Section 4.1.3, the Active Flux scheme for solving the Euler equations is now complete.

4.3 Moving vortex problem

We will test the AF scheme for the Euler equations by initializing with the moving vortex data given by Shu [44],

$$\rho(\mathbf{x},0) = \left(\rho_\infty - \frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}\exp(1-r^2)\right)^{\frac{1}{\gamma-1}} \tag{4.46}$$

$$u(\mathbf{x},0) = u_\infty - \frac{\epsilon y}{2\pi}\exp(\frac{1}{2}(1-r^2)) \tag{4.47}$$

$$v(\mathbf{x},0) = \frac{\epsilon x}{2\pi}\exp(\frac{1}{2}(1-r^2)) \tag{4.48}$$

$$p(\mathbf{x},0) = \rho^\gamma \tag{4.49}$$

where $\epsilon = 5$, $\rho_\infty = 1.0$, $u_\infty = 1.0$, and the domain is square $(x, y) \in [-5, 5]$ with periodic boundaries. We chose this particular vortex because it represents a large range of Mach numbers, from 0.1 to 0.6. The problem is run for one period, $t = 10$, at which time the solution returns to its initial data.

The results from the AF method are compared with those from the Discontinuous Galerkin method with linear (DG1) and quadratic (DG2) reconstructions on structured and unstructured grids. In terms of error calculations, due to its continuous nature, the Active Flux error is computed based on nodes similar to finite difference errors, where nn is the number of total nodes,

$$L_2^{AF}(\mathbf{q}) = \left[\frac{1}{nn}\sum_i^{nn} |\mathbf{q}_i(t) - \mathbf{q}_i(0)|^2\right]^{1/2} \tag{4.50}$$

whereas the DG error is computed based on area integrals,

$$L_2^{DG}(\mathbf{q}) = \left[\frac{1}{|\Omega|}\int_\Omega |\mathbf{q}(t) - \mathbf{q}(0)|^2\,\mathrm{d}\Omega\right]^{1/2} \tag{4.51}$$

Note that both error measurements are computed in the L_2 sense.

Another point of comparison is the Courant condition used by both methods. The DG

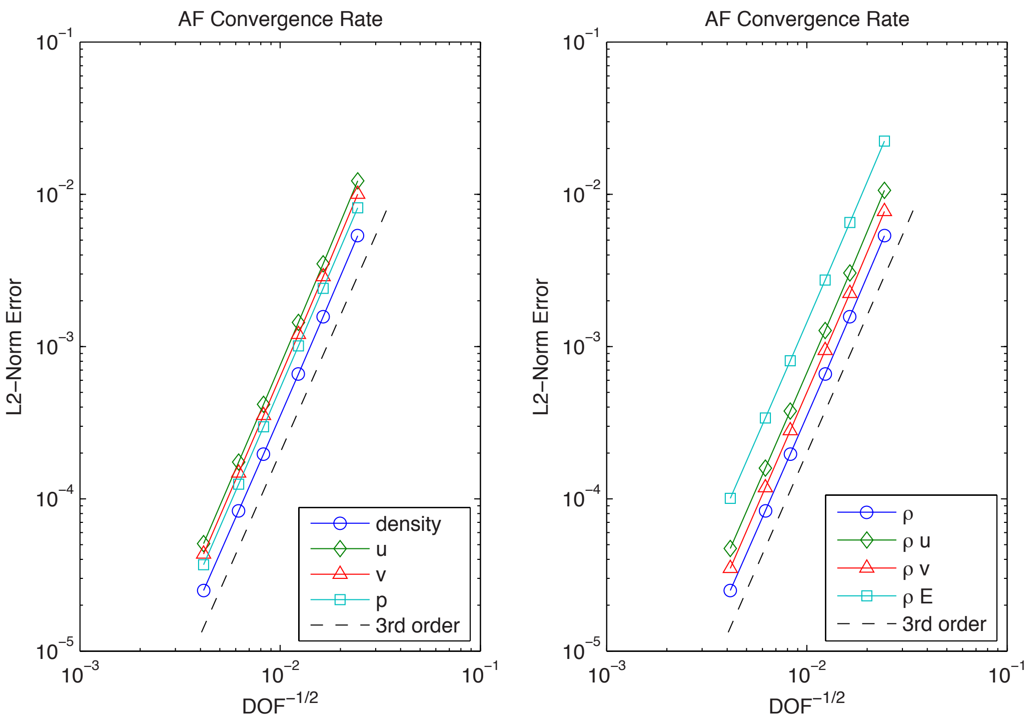


Figure 4.2: Convergence rate of AF results on structured grid after one period.

methods use the following equation to compute the time step,

$$\Delta t_{DG} = \frac{\mathrm{CFL}_{DG}}{\dfrac{1}{2}\max\limits_{i}\left(\dfrac{|\lambda_{\max}|_i}{S_i}\right)} \tag{4.52}$$

where i denotes the element number, $\mathrm{CFL}_{DG} = \mathrm{CFL}_0/2p + 1$, and p is the order of reconstruction. The maximum wave speed $|\lambda_{\max}| = |u| + c$ is calculated along each edge of element i and S is the elemental area. By comparison, the Courant condition used by the AF method is described as follows,

$$\Delta t_{AF} = \frac{\mathrm{CFL}_{AF}}{\max\limits_{i}(|\lambda_{\max}|_i)} \min\left(\frac{S_i}{l_{i,\text{edge}}}\right) \tag{4.53}$$

where $\mathrm{CFL}_{AF} = \mathrm{CFL}_0$ and $l_{i,\text{edge}}$ is one edge length of element i. The Courant number used in comparing the two methods is set to $\mathrm{CFL}_0 = 0.8$.

Starting with the structured grids, we used grid sizes of $n \times n$ where $n =$ (20, 30, 40, 60, 80, 120) for the AF and DG1 methods, and $n =$ (10, 20, 30, 40, 60, 80) for DG2 due to time constraints. Figure 4.2 clearly shows that the AF results converge at third-order for both primitive and conservative states. Comparing with the DG results in Figure 4.3, we observe that all state variables are converging at the expected rate of $p + 1$ for both DG methods. In fact, DG1 is converging at a slightly higher rate of 2.5. The solution errors using the AF method is always smaller than using DG1 and is approximately twice as large as using DG2 with the same total degrees of freedom.

In terms of computational cost, we recorded the runtime, in seconds, of all three methods and compared them against the solution error, as shown in Figure 4.4. Note that both the AF and DG codes are written in C++ by the same person, and though neither code has been optimized for performance, the runtimes should be comparable. We observe that the AF method once again outperforms DG1; for the same solution error, the computation time for DG1 is more than one magnitude larger than AF. The AF method also requires less computation time than DG2 by a factor of approximately 3 to 5 depending on the grid size. At the finer grids, the ratio of the difference in solution error to the computational time of the AF method seems to be about the same as DG2.

Similar observations can be made for the AF and DG solutions on unstructured grids, as shown from Figures 4.5 and 4.6. The unstructured grids were generated by perturbing the nodes on the structured grids. We can conclude this chapter by stating that the AF method for the Euler equations satisfies third-order convergence for all state variables. In terms of performance, it is more accurate than the DG1 method, and is more cost-efficient than both DG1 and DG2 schemes.

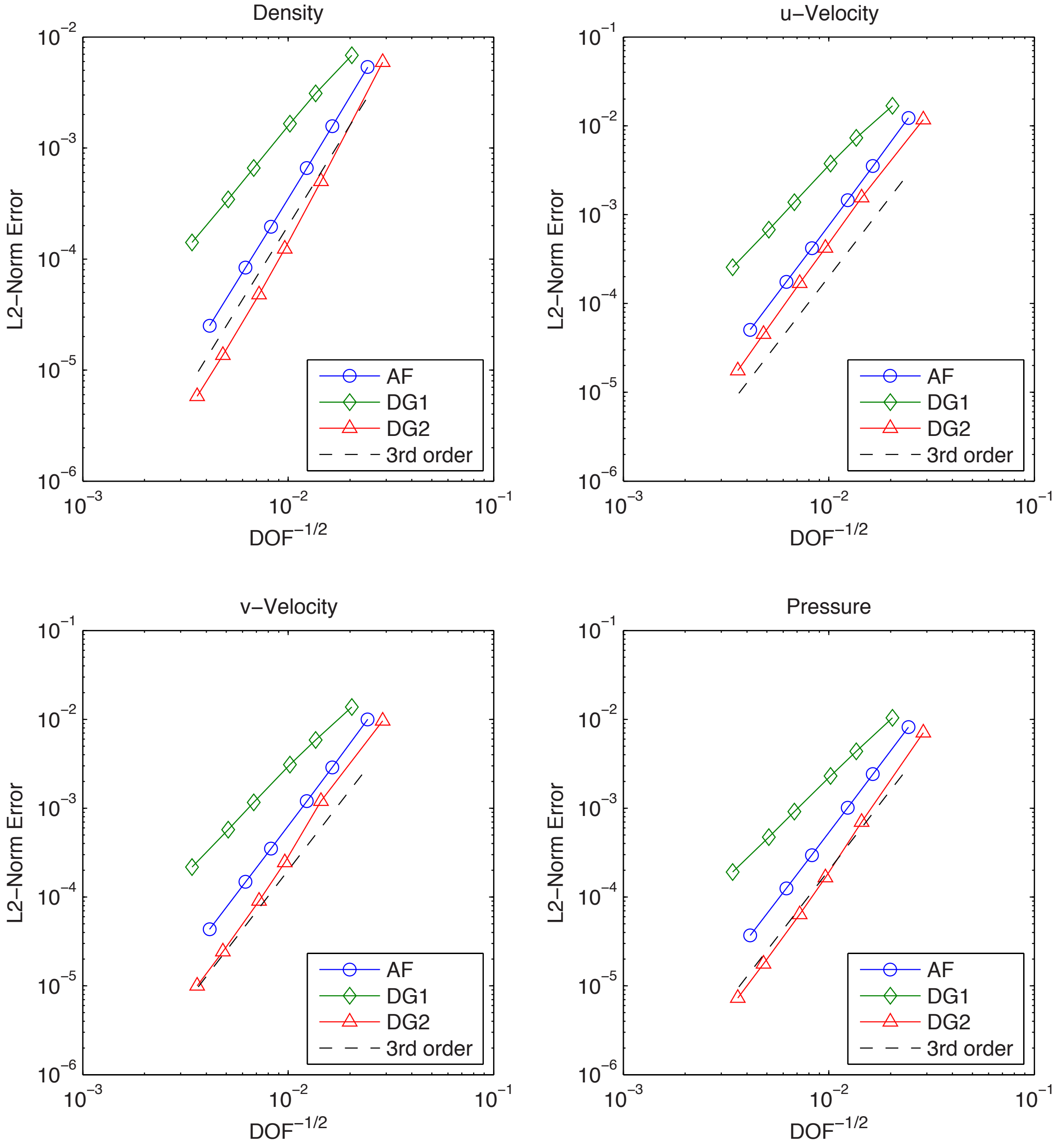


Figure 4.3: Convergence rate of AF and DG results on structured grid after one period.

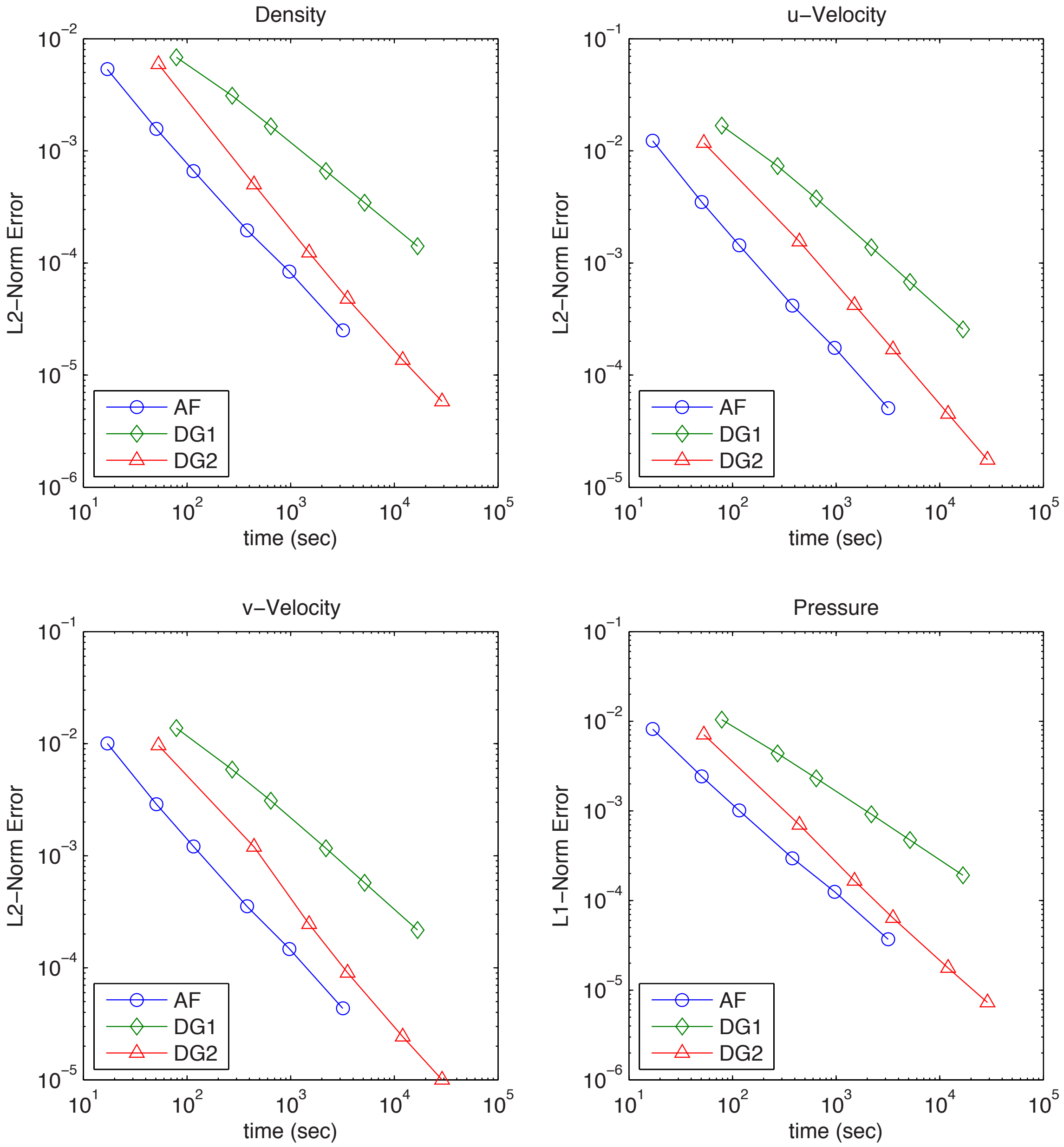


Figure 4.4: Computation time of AF and DG methods on structured grid.

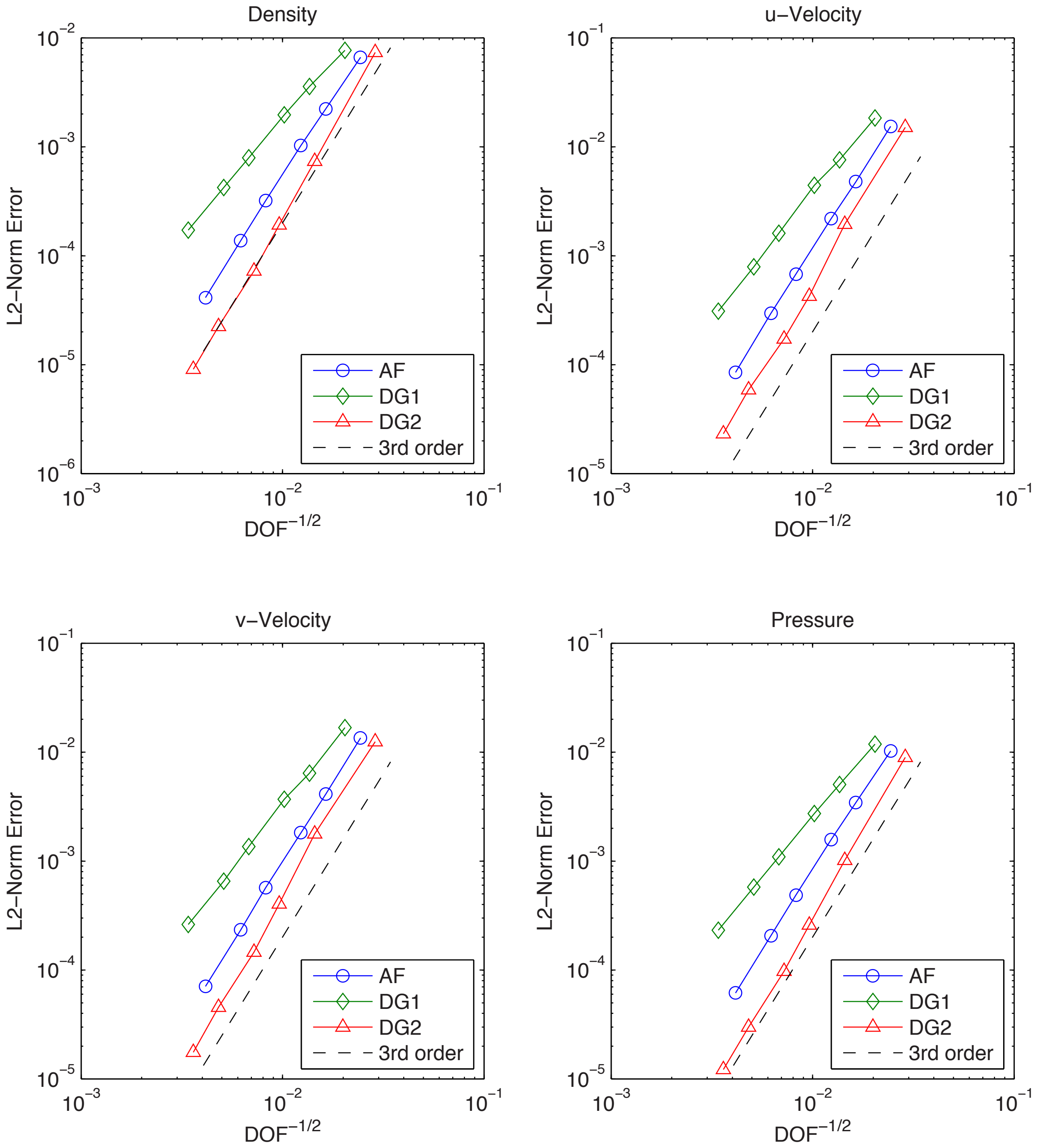


Figure 4.5: Convergence rate of AF and DG results on unstructured grid after one period.

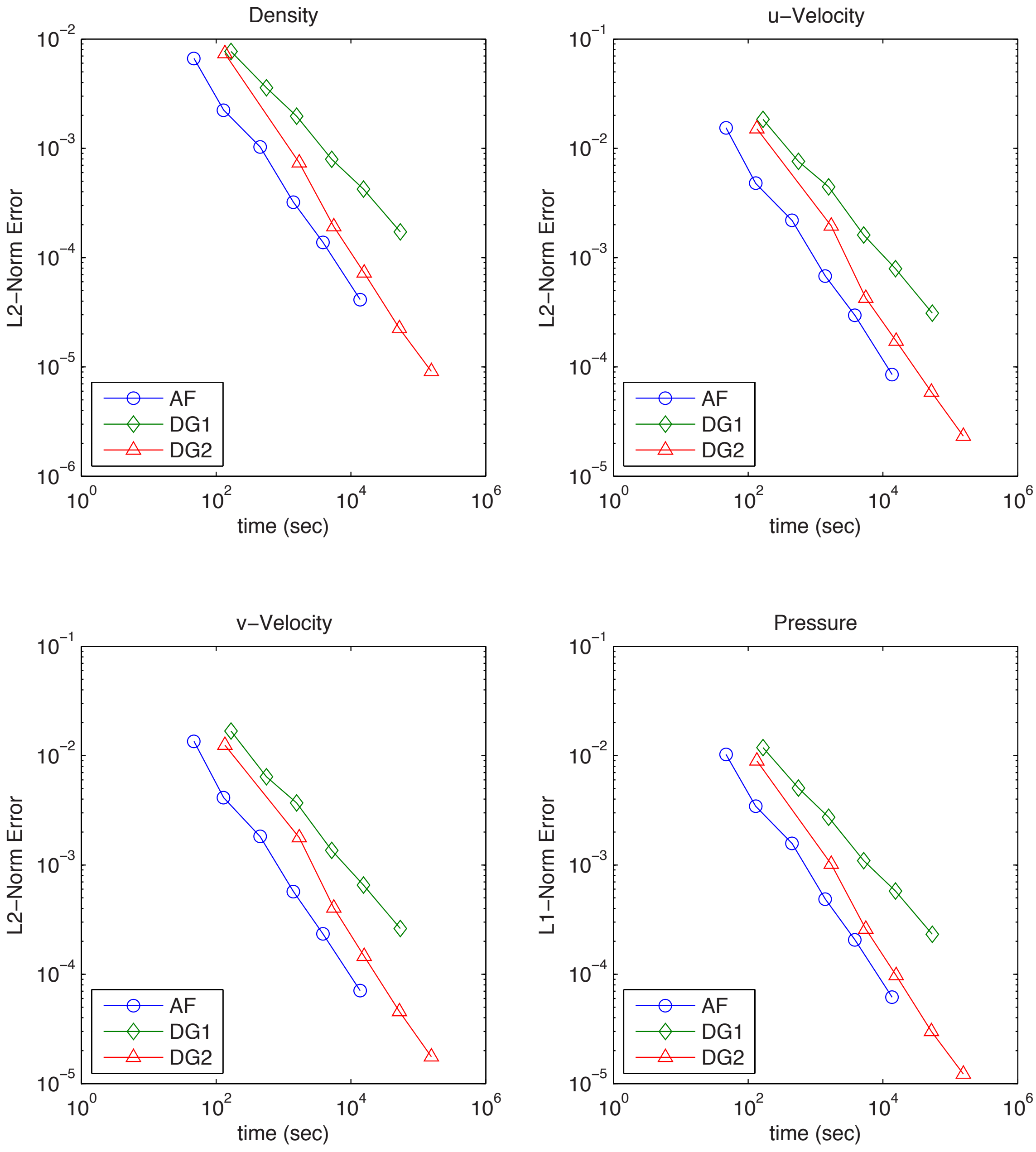


Figure 4.6: Computation time of AF and DG methods on unstructured grid.

CHAPTER 5

Boundary Conditions for Euler Equations

Thus far we have implemented and verified open and flat wall boundaries for linear and nonlinear acoustics systems. With the Euler equations, we now need to account for the appropriate advection conditions at the boundary. This chapter will introduce three types of boundaries that are necessary to the Active Flux method: inflow, open, and curved wall.

5.1 Inflow & outflow

By far the most widely used inflow and outflow boundary conditions for the Euler equations are based on the use of characteristics, usually applied in multidimensions by generalizing the one-dimensional case [36]. We have previously shown in Section 3.3 that a similar concept can be applied with great success to acoustic problems using partial spherical means integrals. In addition, characteristic tracing has already been built into the advective solution as part of the AF method. Hence, by combining the two different boundary considerations, both inflow and outflow boundaries can be solved in a very straightforward manner using the AF scheme.

The outflow, or open boundary condition for the Euler system consists of two steps, as illustrated in Figure 5.1. The first is to use characteristic tracing for the advective update only if the characteristic origin lies within the computational domain; otherwise, the boundary value is unchanged. The second step follows the same process as was performed for the nonlinear acoustics system: we only integrate the spherical means for the portion that is inside the domain. Note that we would compute this partial integration on the data after the advective update has been performed.

The inflow boundary condition for the advective step is essentially the reverse of the outflow. Instead of tracing the characteristic from within the domain, we now use the prescribed data from outside the domain, as illustrated in Figure 5.2. The acoustic step still

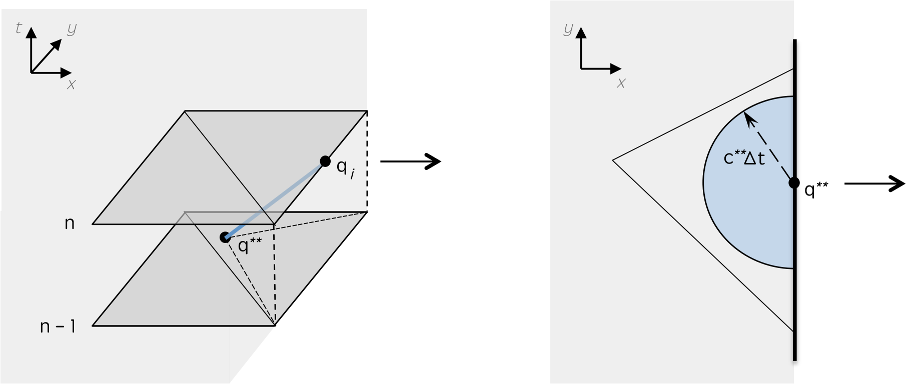


Figure 5.1: Regions where point values are taken to update advective (left) and acoustic (right) terms at an open (outflow) boundary. The interior domain is colored gray.

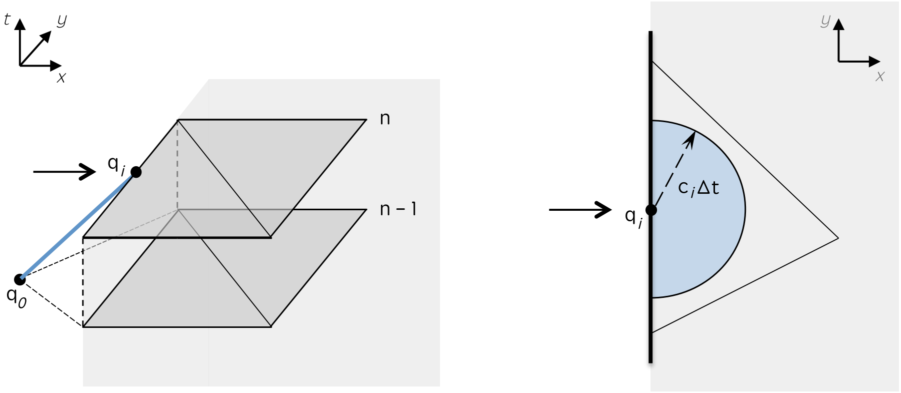


Figure 5.2: Regions where point values are taken to update advective (left) and acoustic (right) terms at an inflow boundary. The interior domain is colored gray.

employs the same concept as for the open boundary; we integrate only the part of the disc that lies within the domain as to allow outgoing waves to escape.

To verify these two boundary conditions in practice, we have used the moving vortex problem described by Equation (4.46) in Section 4.3. Instead of periodic boundaries, we will now use inflow on the left boundary and outflow on the right, and run it until $t = 10$. This should result in the vortex moving completely out of the computational domain, leaving behind the far-field state values of $\mathbf{q}_\infty = (1.0, 1.0, 0, 1.0)^T$. In order to see the effects of the new boundary conditions, we have computed the L_2-norm error and maximum relative error of the solution, as shown in Table 5.1. We observe that the behavior of the L_2-norm error seems to suggest that the AF results are converging to a numerical solution that is not described simply by the far-field solution. It is most likely that the solution is slightly different from the correct one, as if it might be the correct solution for the boundary conditions

	L_2-Norm Error			
Grid size	ρ	u	v	p
30×30	2.19873×10^{-4}	2.65243×10^{-3}	1.39665×10^{-3}	3.07816×10^{-3}
60×60	2.26268×10^{-4}	2.64294×10^{-3}	1.35370×10^{-3}	3.16822×10^{-3}
120×120	2.30599×10^{-4}	2.67363×10^{-3}	1.35163×10^{-3}	3.22973×10^{-3}

	Maximum Relative Error			
Grid size	ρ	u	v	p
30×30	0.0136691	0.0156261	0.0111425	0.0191905
60×60	0.0143315	0.0161398	0.0104494	0.0201212
120×120	0.0169222	0.0186734	0.0106683	0.0237708

Table 5.1: Solution errors from the moving vortex problem with inflow and outflow boundary conditions.

being imposed. This would result in some waves being reflected back into the the domain. From the computation of maximum errors, we can conclude that the solution errors caused by the boundaries are weakly dependent on the grid size and range from 1.3% to 2.3%, which are well within the widely accepted limit of 5%.

5.2 Curved wall

In order for a high-order method to achieve its high order of convergence at a wall boundary, additional information is required to describe the wall geometry in more detail. A common practice is to increase the number of spatial nodes so that curved boundaries can be reconstructed to the required order, as is done in DG methods. For the AF method, however, this technique is not quite desirable for a few reasons. The first is obvious; the memory storage required would be higher since the number of grid nodes would be doubled if we use quadratic reconstruction instead of linear, and this could increase the computational cost. In terms of the AF method itself, the use of curved elements adds difficulty in the execution of both advective and acoustic schemes. For the advective solver, determining which element the characteristic origin lies in becomes more computationally expensive because the Jacobian is no longer a constant, while for the acoustic solver, it becomes much more complicated to compute the spherical means integral to the required order of accuracy with the limits of integration set as a curved edge. It is for these reasons that we choose to use

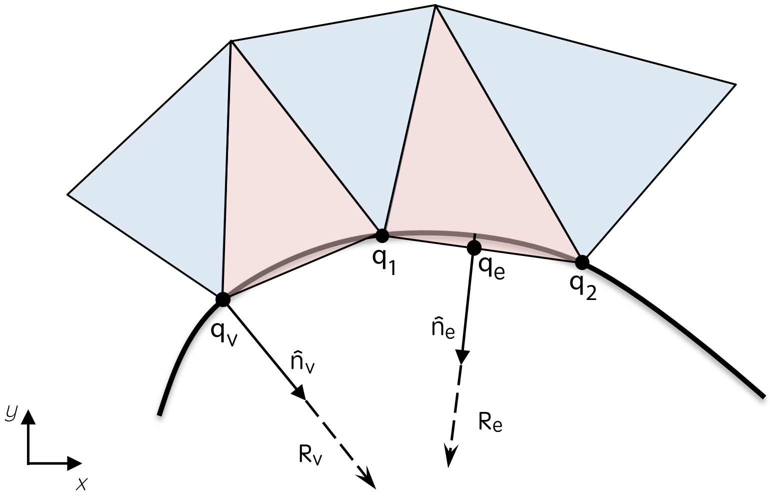


Figure 5.3: Nomenclature for curved wall boundary conditions at vertex and edge nodes. All of the blue cells are evaluated before red.

the radius of curvature instead to define a high-order geometry. This results in elements of low-order geometry with a constant Jacobian but varying radii of curvature defined at every vertex and edge nodes.

Four distinct conditions are needed in order to solve for solutions at the wall boundary. The first is simple; because the curved wall is a streamline, our advective update using characteristic tracing naturally satisfies the wall boundary conditions for an advection process. We also know that entropy is advected along a streamline, hence from the advection stage, we can determine the first boundary condition.

$$s = \log\left(\frac{p^{**}}{\rho^{**\gamma}}\right) = \log p^{**} - \gamma \log \rho^{**} \tag{5.1}$$

Since this entropy value is not affected by an acoustic process, we can use Equation (5.1) to relate the final pressure and density solutions. The three remaining conditions can be directly imposed after the acoustics stage at vertex node v as the following,

$$\mathbf{v}_v \cdot \hat{\mathbf{n}}_v = 0 \tag{5.2}$$

$$\begin{aligned} \frac{\partial p_v}{\partial \mathbf{n}_v} &= \frac{\rho q_v^2}{R_v} \\ &= \frac{q_v^2}{R_v}\left(\frac{p_v}{\exp s_v}\right)^{1/\gamma} \end{aligned} \tag{5.3}$$

$$\omega_v = \nabla \times \mathbf{v}_v$$

$$= \frac{\partial v_v}{\partial x} - \frac{\partial u_v}{\partial y} = 0 \tag{5.4}$$

where R is the radius of curvature and $q = \sqrt{u^2 + v^2}$ is the velocity magnitude. Note that Equation (5.3) evaluated at either the vertex or edge node will be of the form $A + Bp = Cp^{1/\gamma}$, which can be easily solved using an iterative scheme such as Newton's method. In addition, an alternative constraint will need to be considered to replace Equation (5.4) for rotational flows.

Now consider an edge node on the curved boundary shown in Figure 5.3. We can assume that all of the cells with vertex boundary nodes have already been updated correctly using Equations (5.1) to (5.4). Since the edge 1-2 will not in general form part of the boundary, we must enforce the boundary condition described by Equation (5.2) in a weak sense by insisting that there is no flow through the arc 1-2. This is equivalent to zero mass flow through the edge 1-e-2, which we can integrate numerically using Simpsons Rule.

$$\int_1^2 \rho v_n \, \mathrm{d}s \approx \frac{1}{6}(\rho_1 v_{1,n} + 4\rho_e v_{e,n} + \rho_2 v_{2,n}) = 0 \tag{5.5}$$

The resulting normal velocity will most likely be a very small but nonzero value. The other three conditions still hold at the edge nodes.

5.2.1 Gaussian bump

This problem is aimed at testing the AF method for the computation of internal isentropic flow with a high-order curved boundary consideration. The constant initial conditions are

$$\begin{aligned} \rho_\infty &= 1.4 \\ c_\infty &= 1.0 \\ M_\infty &= 0.5 \\ \alpha &= 0 \end{aligned} \tag{5.6}$$

and $\gamma = 1.4$. Note that the entropy should remain a constant in the flow field. We initialize the data on a computational domain proposed by Wang et. al [1] where the bump along the lower wall is defined as

$$y = 0.0625 \exp(-25x^2) \tag{5.7}$$

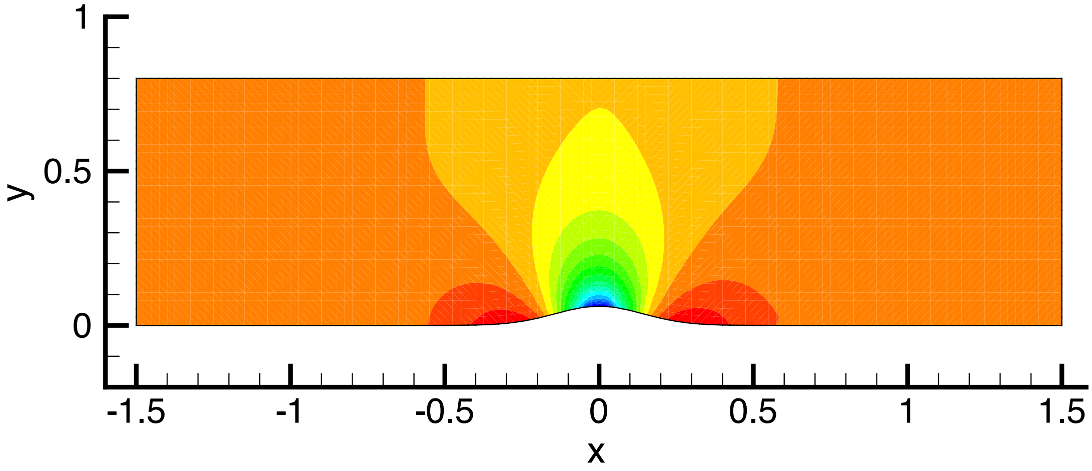


Figure 5.4: Density solution of Gaussian bump problem on coarse grid of 1800 elements.

We can explicitly compute the radius of curvature at any point along the bump using Equation (5.7).

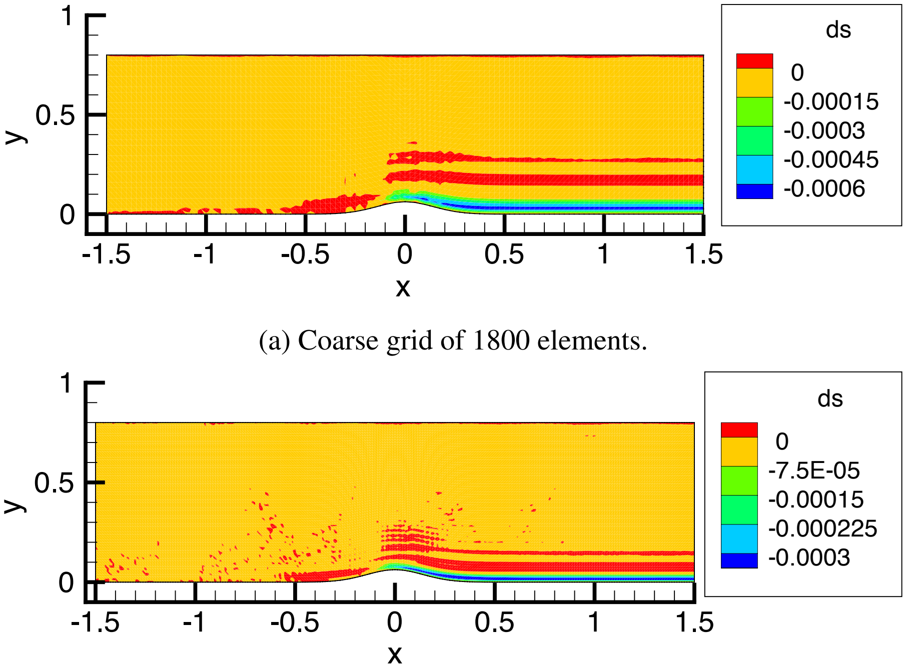
$$R(x) = \frac{\left[1 + y'^2(x)\right]^{3/2}}{|y''(x)|} \tag{5.8}$$

The problem should be run until steady state. However, because the AF method is fully explicit, its solutions are time-dependent and we would need to run the code for a very long time in order to reach steady state. Due to time constraints, we ran the problem until the solution residual reached 10^{-3} in hopes of making some qualitative and possibly quantitative observations about AF method with curved wall boundary. Figure 5.4 shows that the density solution at the final time is close to being symmetric about the bump as expected.

Since the complete analytical solution is unknown for this particular problem, the entropy error is usually used as the indicator of solution accuracy. Looking at the relative entropy error profile of the AF solution in Figure 5.5, we first observe that the entropy produced is very small; it is only 0.015% to 0.06% of the initial value on the coarse grid. In addition, the maximum entropy error decreases by a factor of two when the grid is refined once. We can also compute the total entropy error and the entropy produced along the bump to determine whether they converge at the correct order.

$$Err_{\text{total}}(s) = \frac{1}{nn} \sum_{i}^{nn} |s(t) - s(0)| \tag{5.9}$$

$$Err_{\text{bump}}(s) = \frac{1}{\partial\Omega} \int_{\partial\Omega} |s(t) - s(0)| \, \mathrm{d}S \tag{5.10}$$



(a) Coarse grid of 1800 elements.

(b) Fine grid of 7200 elements.

Figure 5.5: Entropy errors of Gaussian bump problem.

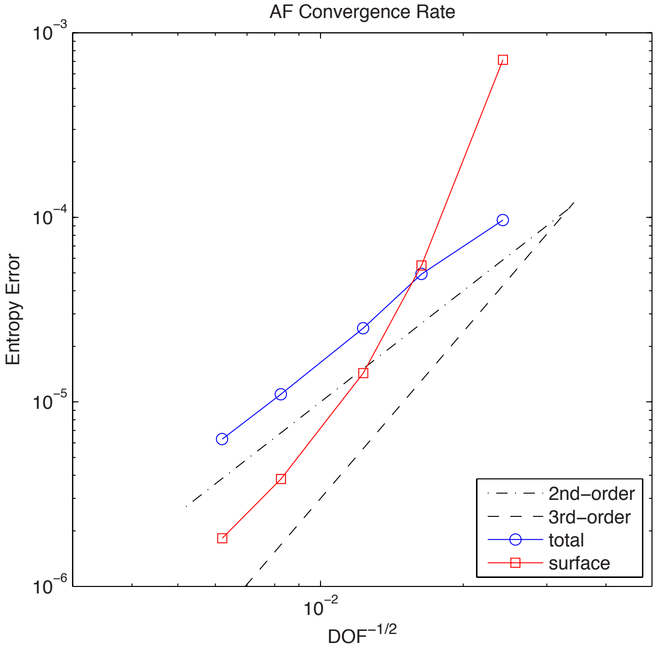


Figure 5.6: AF convergence rate of entropy errors for Gaussian bump problem; $s = \log(p/\rho^{\gamma})$.

We observe from Figure 5.6 that the total entropy error is converging at only second-order while the the error at the surface seems to converge at third-order. This could be explained by the fact that the steady state solution has not yet been reached, thus the intermediate solution is resolved with some error in the total entropy. However, along the bump, the solutions are indeed being solved with third-order accuracy with the boundary conditions previously specified. Based on these results and given ample time, the potential for a steady state solution to converge at third-order is high.

In terms of comparing quantitative values of the total entropy error produced, the AF solutions can be tentatively compared to DG1 steady state solutions by using the results from [1]. Figure 5.7 shows that while the AF method only converges at second-order at a solution residual of 10^{-3}, its total entropy errors are smaller than that of DG1 on coarse grids. We should also note that superconvergence is usually observed for the DG method for this problem, but in certain situations where different wall boundary conditions have been specified, as was most likely the case with the DG1 solver from University of Tennessee, the order of convergence may be less than $2p + 1$.

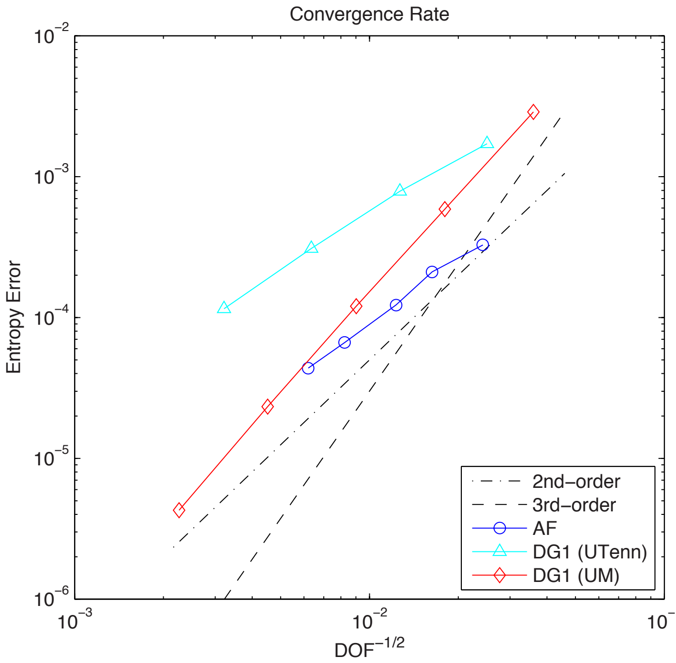


Figure 5.7: Convergence rates of AF and DG1 methods from University of Tennessee (UTenn) and University of Michigan (UM) [1] for Gaussian bump problem;
$s = p/\rho^{\gamma}$.

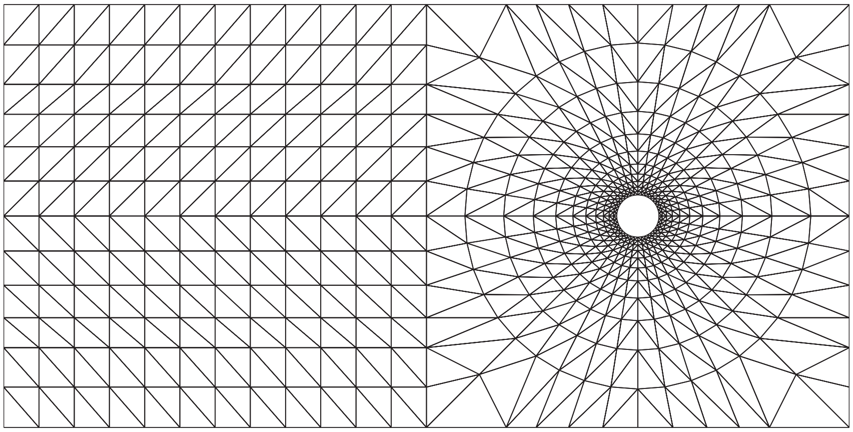


Figure 5.8: Rectangular grid with structured triangular elements in the extended front half of the domain, and symmetric elements around the cylinder.

5.2.2 Flow past a cylinder

Next, we will explore the solution to the inviscid flow past a cylinder using the AF method. The freestream variables are described based on the freestream Mach number M_∞, velocity in the x-direction u_∞, and pressure p_∞.

$$\begin{aligned} c_\infty &= u_\infty / M_\infty \\ \rho_\infty &= \gamma p_\infty / c_\infty^2 \\ K &= p_\infty / \rho_\infty^\gamma \end{aligned} \tag{5.11}$$

where K is the isentropic constant. The initial conditions are the well-known potential flow solutions of incompressible flow past a cylinder of radius R_c, which are described as

$$p(x,y,0) = p_\infty + \frac{1}{2}\rho_\infty u_\infty^2 R_c^2 \frac{2x^2 - 2y^2 - R_c^2}{(x^2+y^2)^2} \tag{5.12}$$

$$u(x,y,0) = u_\infty + u_\infty R_c^2 \frac{y^2 - x^2}{(x^2+y^2)^2} \tag{5.13}$$

$$v(x,y,0) = -2u_\infty R_c^2 \frac{xy}{(x^2+y^2)^2} \tag{5.14}$$

$$\rho(x,y,0) = (p(x,y,0)/K)^{1/\gamma} \tag{5.15}$$

For this particular test, we have set $M_\infty = 0.2$, $u_\infty = 0.2$, $p_\infty = 2$, and $R_c = 0.2$.

The left boundary is an inlet with freestream conditions, and the top, bottom, and right boundaries are left open. The size of the mesh domain is generally set to ten times the radius of the cylinder ($L = 10R_c$) in order to reduce the impact of the inflow and outflow

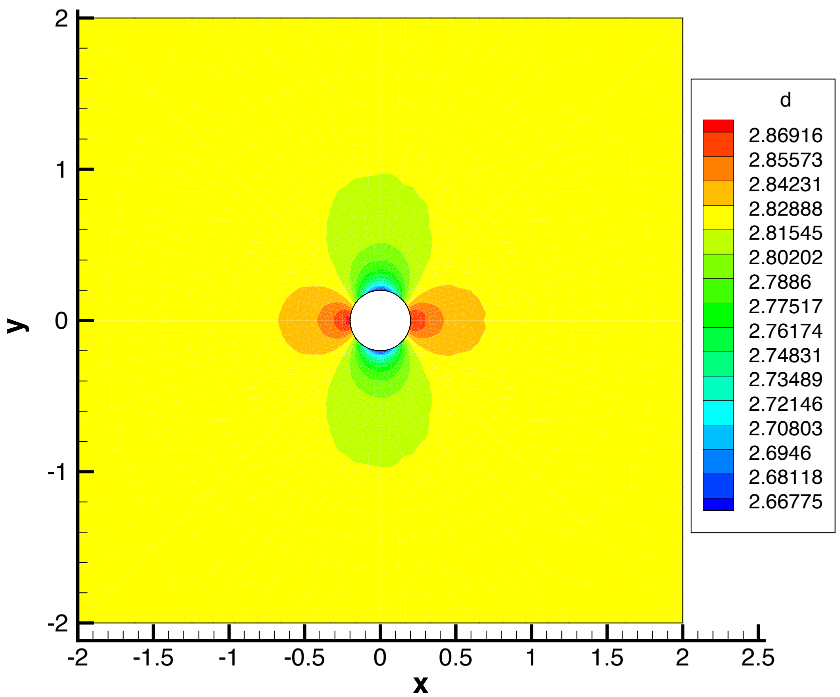


Figure 5.9: Density distribution for flow past cylinder problem with 40 vertex nodes around cylinder; $t = 50$.

boundary conditions. In this case, we would like to increase the domain in front of the cylinder to $30R_c$ in order to ensure the streamlines are parallel as they approach the cylinder. Hence, the problem will be solved on the extended rectangular domain of $x \in [-3L, L]$ and $y \in [-L, L]$ as illustrated in Figure 5.8.

The inviscid flow problem uses a grid with 40 vertex nodes distributed evenly around the cylinder and run until $t = 50$, which is when the solution residual is around 10^{-3}. While this again does not result in the steady-state solution, we can certainly make useful observations about the qualitative accuracy of the curved wall boundary conditions. The density distribution is shown in Figure 5.9 and displays convincing top-down and fore-aft symmetry around the cylinder. Looking at the Mach number contours in Figure 5.10, however, we observe that while the solution is symmetric about the horizontal axis, the fore-aft symmetry is a bit lacking. Regardless, the AF scheme is still a huge improvement over methods such as the matrix-distribution Lax-Wendroff Positive Streamwise Invariant (PSI) scheme that produces a clear artificial wake behind the cylinder, and the second-order upwind FV method that destroys the fore-aft symmetry of the solution [19]. We can also use results of this problem from Carette et al. [45], which is shown in Figure 5.11, to compare

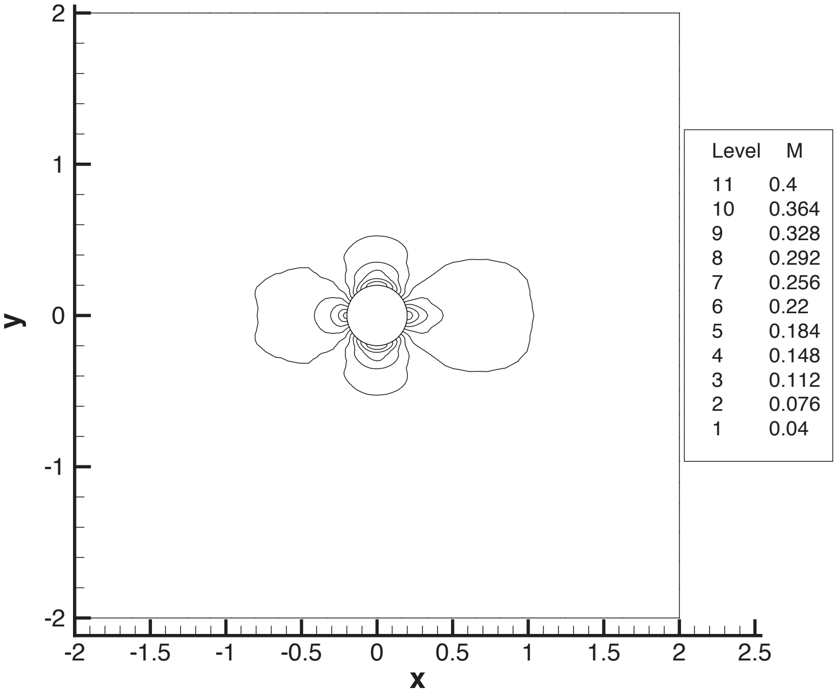


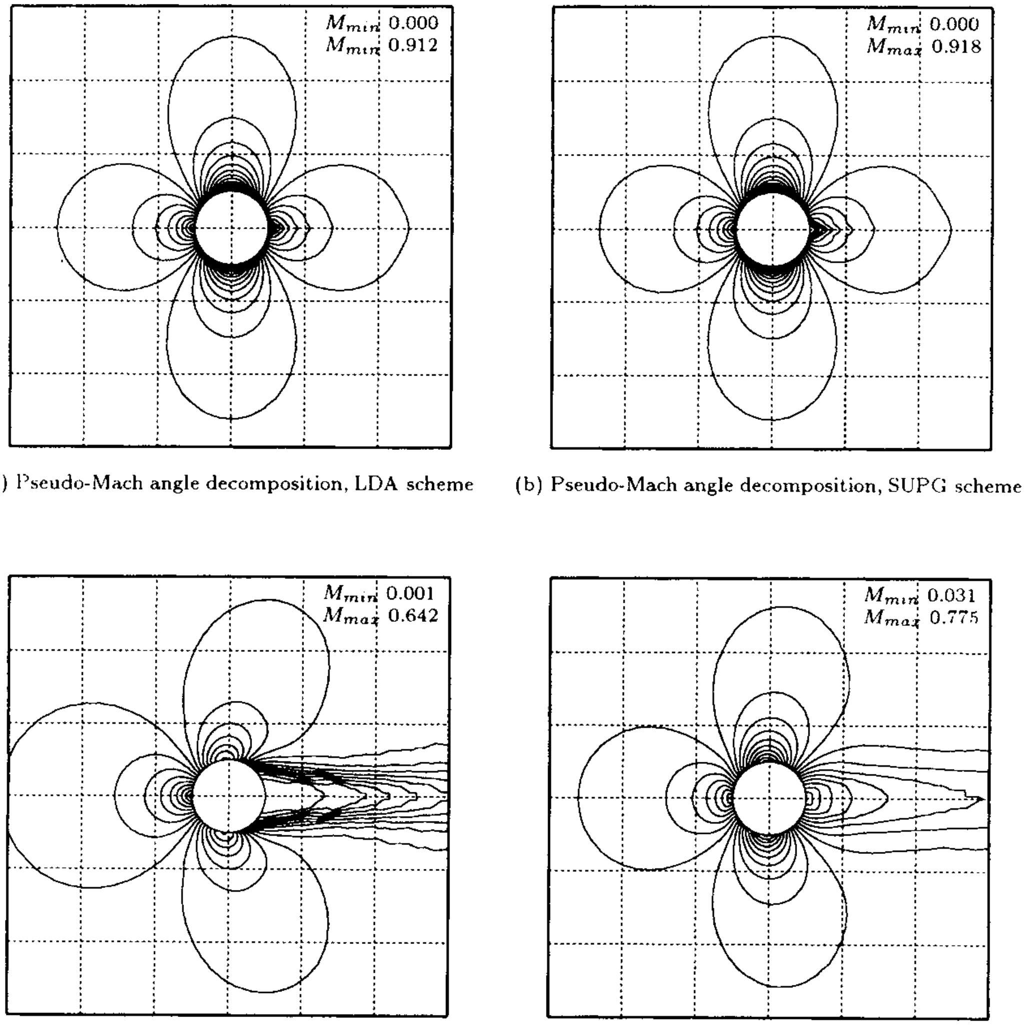
Figure 5.10: Mach contours for flow past cylinder problem with 40 vertex nodes around cylinder; $t = 50$.

to our AF results [1]. Note that the results from Carette et al. uses an initial freestream Mach number of 0.38 and a fine grid of 128 points around the cylinder. Considering the AF solution is computed on a rather coarse grid of only 40 points around the cylinder, Figure 5.10 reinforces the fact that the AF method is capable of producing results with good fore-aft symmetry and low numerical dissipation when compared to methods such as the simple wave model D or schemes that use artificial dissipation.

Two additional solution properties that are commonly analyzed are the lift and drag forces produced on the surface of the cylinder due to the pressure distribution. The lift and drag coefficients are computed by normalizing the respective forces with respect to the freestream density, speed, and cylinder diameter D_c.

$$C_L = \frac{1}{\frac{1}{2}\rho_\infty q_\infty^2 D_c} \int_{-\pi}^{\pi} p \sin\theta \, \mathrm{d}\theta \tag{5.16}$$

[1] The best results from Carette et al. make use of the “pseudo-Mach number decomposition”, which is a technique that carefully distinguishes subsonic and supersonic flows. It therefore has something in common with the AF method, but it was not found possible to extend to three dimensions.



(a) Pseudo-Mach angle decomposition, LDA scheme

(b) Pseudo-Mach angle decomposition, SUPG scheme

(c) Model D, LDA scheme

(d) Pseudo-Mach angle decomp., SUPG + AV scheme

Figure 5.11: Mach contours for flow past a cylinder with $M_\infty = 0.38$ using various other methods [45].

$$C_D = \frac{1}{\frac{1}{2}\rho_\infty q_\infty^2 D_c} \int_{-\pi}^{\pi} p \cos\theta \, \mathrm{d}\theta \tag{5.17}$$

The same inviscid problem is rerun on grids with 40, 80, and 160 vertex nodes distributed evenly around the cylinder, and the resulting lift and drag coefficients on the cylinder plotted against runtime are shown in Figure 5.12. As expected, the total lift is zero for an inviscid flow over a top-down symmetric object such as a cylinder, and this is true for all of the grids tested. For the drag, we observe that it has convincingly converged to a constant value as the AF solution approaches steady state. In addition, as the grid is refined, the

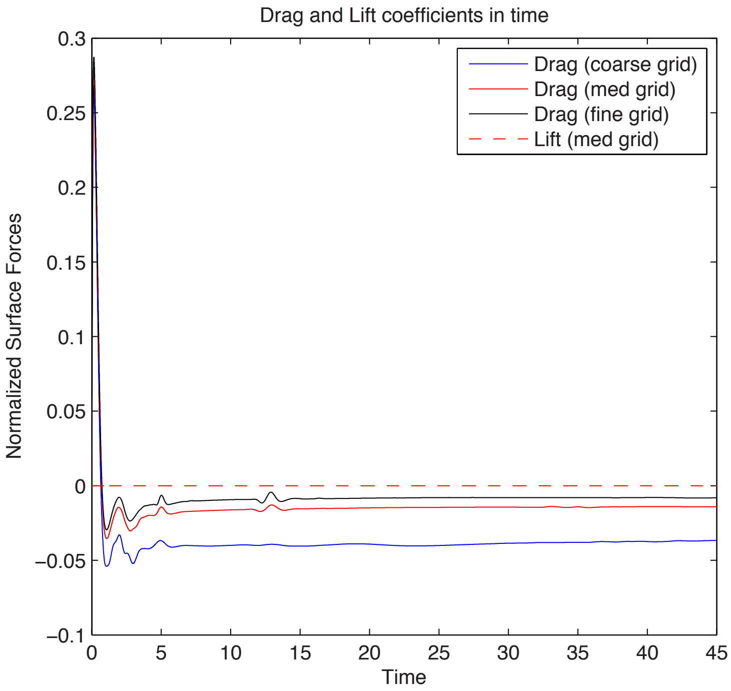


Figure 5.12: Drag and lift coefficients on the cylinder in time. Grids used contain 40 (coarse), 80 (medium), and 160 (fine) nodes distributed around the cylinder.

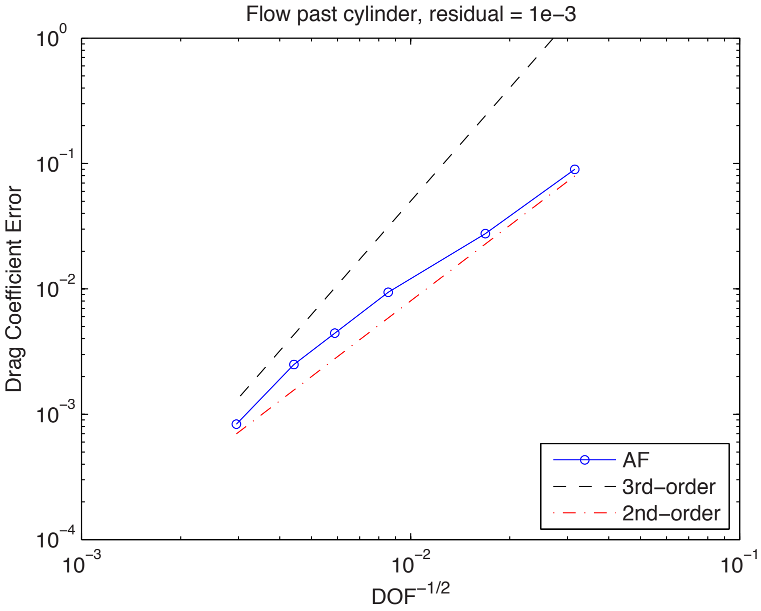


Figure 5.13: Convergence plot of drag coefficient on cylinder.

Model	Scheme	C_L	C_D
PMA	LDA	-0.010689	0.000526
PMA	SUPG	0.006814	-0.001232
D	LDA	0.000924	0.676016
PMA	SUPG + AV	0.002084	0.324879

Table 5.2: Table reproduced from Carette et al. [45]. Steady-state solution of inviscid flow past a cylinder with $M_\infty = 0.38$.

numerical drag coefficient approaches the theoretical limit of zero, which indicates that the solutions become more fore-aft symmetric around the cylinder. Comparing the drag coefficient of -0.008158 on the fine grid to the values produced by Carette et al. [45], which is shown in Table 5.2, it is clear that the AF solutions result a drag coefficient that is comparable to the pseudo-Mach angle (PMA) decomposition with either LDA or SUPG scheme, which outperforms the other two methods by an order of magnitude of three. In addition, the lift coefficient is only -6.0787×10^{-5} from the AF solutions on the fine grid, which is much smaller than any of the lift values computed from the other methods shown in Table 5.2. This suggests that the AF method results in superior top-bottom symmetry.

Furthermore, while we do not have the exact solution for this problem to compare to, we can still determine the order of convergence of the AF solution along the wall boundary by using the drag coefficient error. Note that the exact numerical solution used is determined by solving the problem on a very fine grid with 320 vertex nodes around the cylinder. Figure 5.13 clearly shows that as the grid is refined, the convergence rate approaches three. These drag results are promising in establishing third-order convergence when solving more complex flow problems using the Active Flux method.

Recall from Chapter 4 that the current AF method adds a correction term to the non-conservative nodal solutions in order to raise the overall order of accuracy to three. The solutions shown thus far did *not* have any of the curved wall boundary nodes changed using this correction constant. Instead, only the nodes inside the domain have been updated to third-order accuracy. Now, we would like to also update the wall boundary nodes with the correction terms, but with an additional constraint that the velocity normal to the wall stays zero. Figure 5.14 shows that with this new boundary update, the density profile shows that a very small wake has developed behind the cylinder and the solution no longer displays fore-aft symmetry. Similarly, the pressure profile also shows less fore-aft symmetry as illustrated in Figure 5.15 but the pressure wake is much smaller and less noticeable than the density. One possible explanation for this drawback is that with the additional corrections

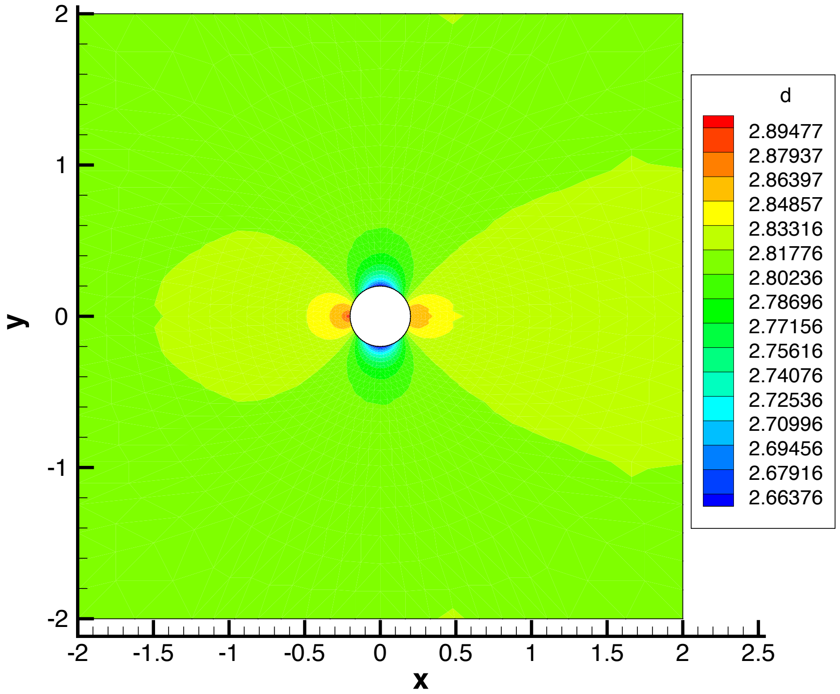


Figure 5.14: Density profile with conservative correction terms applied at the wall boundary with 40 vertex nodes around cylinder; $t = 50$.

to density and pressure, there is no guarantee that the conditions specified by Equations (5.1) and (5.3) are always satisfied at the boundary. On the other hand, if we plot the pressure coefficient in as shown in Figure 5.16, we observe that the pressure curve at its minimum point ($x = 0$) is smoother with this new wall boundary update.

In terms of the effect that updating the boundary nodes has on the drag coefficient, Figure 5.17 shows that the drag coefficient is reduced drastically with this new boundary update on a grid with 80 nodes around the cylinder. In fact, the new value of C_D is -0.00425, which is more than three times less than the previous value of -0.01407 shown in Figure 5.12. Note that the lift coefficient still remains at zero.

Finally, although we were only able to tentatively identify the order of convergence for the AF method with curved boundaries by using drag errors, we were able to conclude that its qualitative results are very promising. The symmetry of the AF solutions across both horizontal and vertical axes was particularly encouraging, especially when compared to the solutions from other numerical methods. We hope to eventually develop a steady-state AF solver in order to reduce computational time for such problems and to show stronger third-order results.

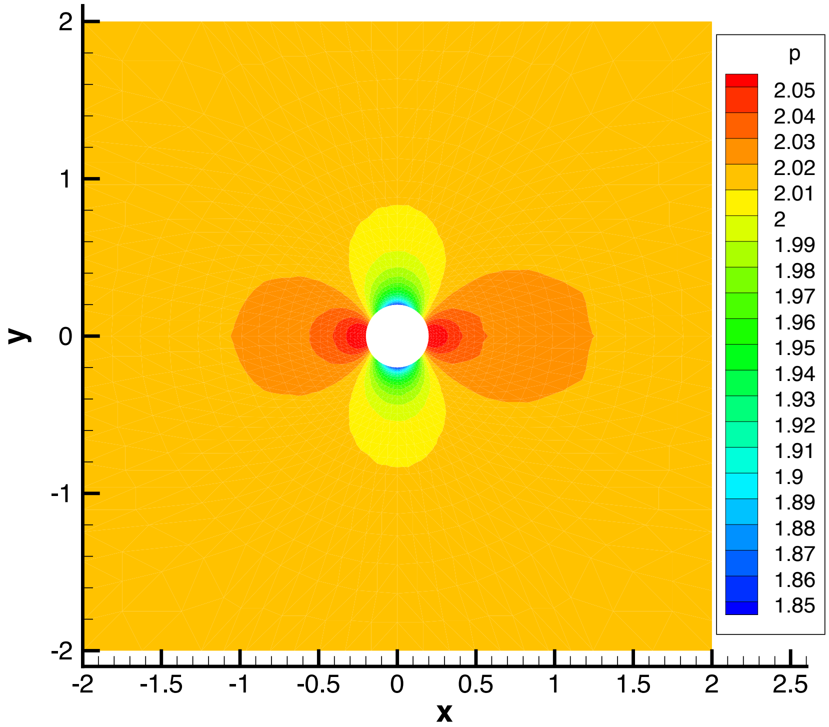


Figure 5.15: Pressure profile with conservative correction terms applied at the wall boundary with 40 vertex nodes around cylinder; $t = 50$.

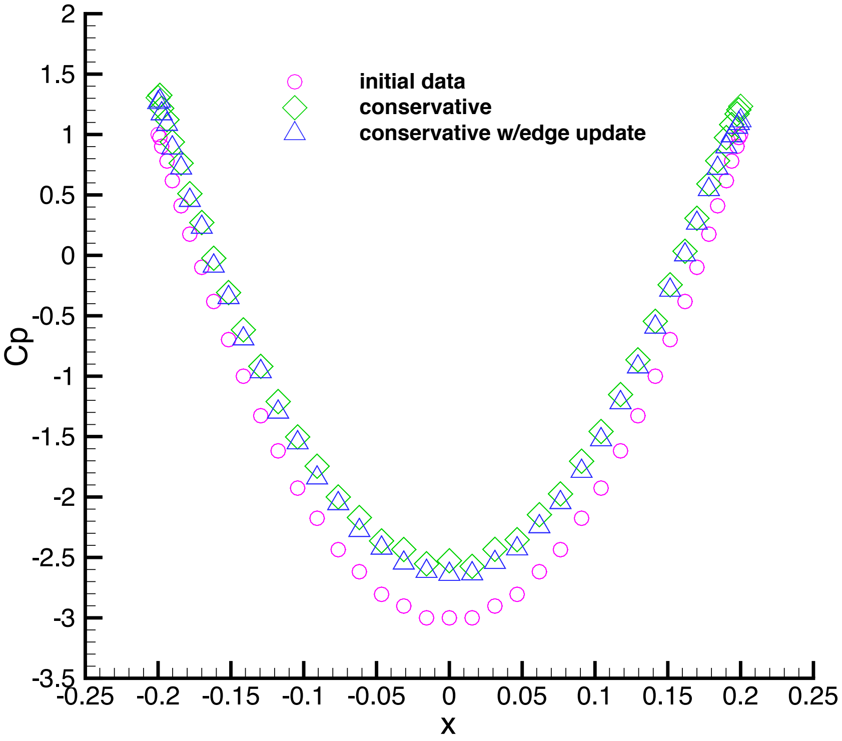


Figure 5.16: Pressure coefficient of various solutions on the cylinder.

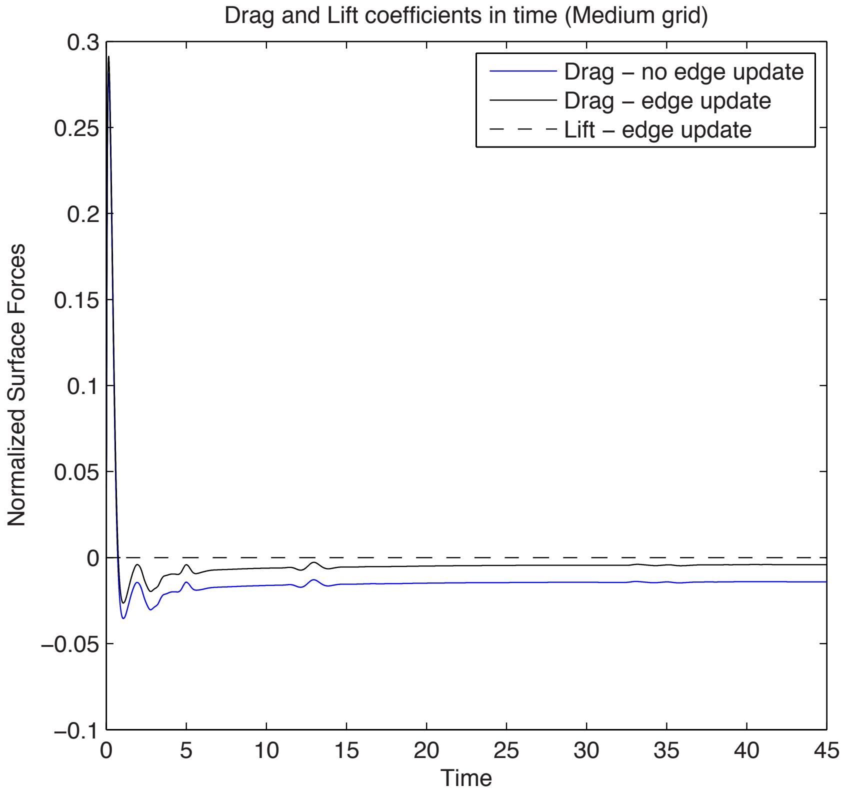


Figure 5.17: Drag and lift coefficients on the cylinder in time with and without conservative boundary node updates; grid with 80 vertex nodes around cylinder.

CHAPTER 6

Conclusions

6.1 Summary

This dissertation details the development of the Active Flux scheme pertaining to acoustic systems and subsystems, as in the case with the Euler equations. The main form of the nonconservative update in the scheme is the evaluation of spherical means integral at the nodes, which possesses fully multidimensional properties that naturally devolves in two-dimensions to integration over a disc, and in one-dimension to the method of characteristics. The scheme is demonstrated for a linear acoustics system, the p-system of hyperbolic conservation laws, and the acoustic portion of the full Euler equations. In addition, a new method to reconcile the nonconservative data with the conservative cell-average data without the use of a third-order bubble function is introduced in Chapter 4.

The numerical properties of the method are examined in detail in Chapter 2, including vorticity preservation, robustness, stability, and convergence. Chapter 3 extends the linear acoustics system to nonlinear by reducing the exact evolution formula using spherical means to second-order accuracy. From the formulation of the linear and nonlinear solutions, we notice a striking similarity to the Lax-Wendroff method. Thus, in Chapter 4, we use the second-order Lax-Wendroff method to evaluate the Euler equations. The AF scheme is then applied differently to the advective and acoustic terms in the Euler formulation and shown to converge at third-order. The end of Chapter 4 provides a direct comparison between the AF method and DG methods using linear and quadratic reconstructions.

Finally, numerous boundary conditions for the AF scheme have been developed throughout this dissertation. We start with the simpler flat wall and open boundaries applied to the linear and nonlinear acoustic equations in Chapters 2 and 3, respectively, and then move onto the development of inflow, outflow, and curved wall boundaries for the Euler equations in Chapter 5.

6.2 Research contributions

The major contributions of this dissertation are:

- Extension of Eymann's AF method for the scalar wave equation into a general framework applicable to nonlinear acoustic systems.

- Development and implementation of the AF method for the full Euler equations using nonlinear operator splitting of primitive variables.

- Use of a constant correction term instead of bubble function in the reconciliation stage of the AF scheme to obtain third-order accuracy.

- Development of various boundary conditions for the Active Flux method including open, flat wall, inflow, and curved wall boundaries.

6.3 Future work

The current Active Flux method for the full Euler system is capable of solving various types of problems, which should be verified in the near future to confirm third-order convergence. In addition, the following topics could be explored further.

- **Steady state:** The AF method is fully discrete, which means that its solutions are time-dependent. As seen in Chapter 5, it is very time consuming for a problem to reach steady state with the current AF schemes. Techniques could be developed to somehow remove the time dependence from the steady-state solution.

- **Computational efficiency:** Although we have shown in Chapter 4 that the AF method is currently more efficient than both DG1 and DG2 methods, we believe that there are still ways to improve. One area of possible improvement is to compute the solutions at the full timestep by using some of the data from the half timestep.

- **Viscous terms:** The eventual goal is to obtain an AF method that is capable of solving the full Navier-Stokes equations in order to accurately compute near-body flows. An advective-diffusion scheme has already been implemented by Nishikawa that rewrites the one-dimensional Navier-Stokes equations as a first-order order system of hyperbolic equations [46] and could possibly be extended into multidimensions. A drawback of his formulation is that it assumes steady-state calculation, so modifications would be required for unsteady problems.

For the acoustics solver, there are a couple of important topics still left to cover that would be beneficial to the development of the AF method for the Euler equations.

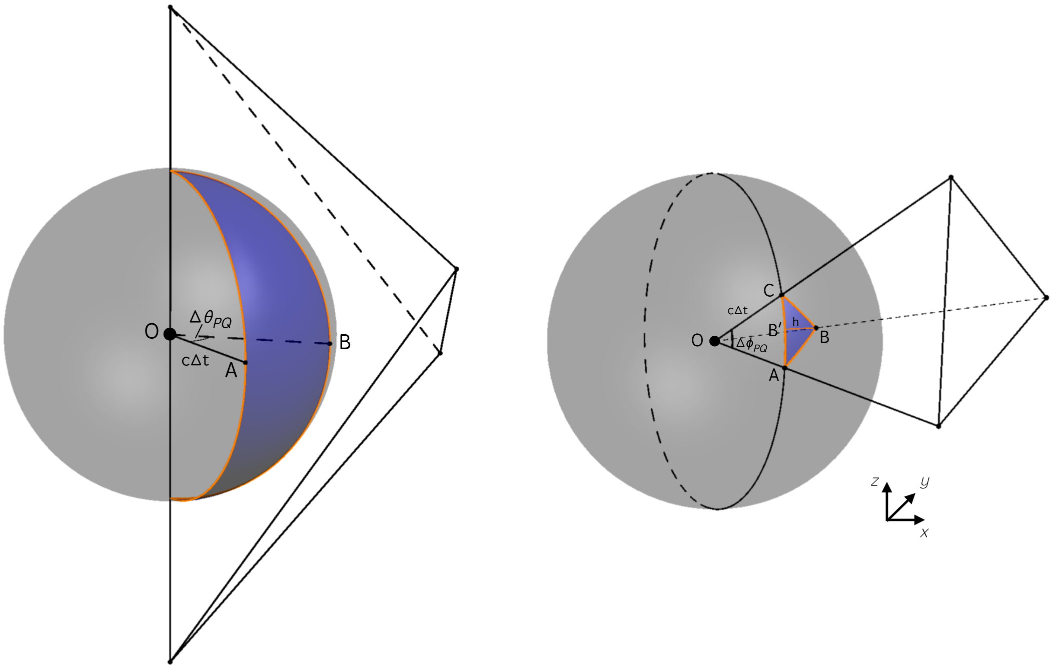
- **Multidimensional nonlinear limiting:** As with any numerical method with higher than first order accuracy, some form of nonlinear limiting is necessary for numerical accuracy. In general, when bounds on the solution values are available, the limiter can be designed to respect them. However, in the case of the wave equations, waves in multidimensions may focus and produce pressure much greater than those initially present. A promising novel limiting approach has been developed by Roe, Lung, and Maeng [47] that takes a tentative step forward in time and reevaluates the time derivative. This has been applied successfully to the Lax-Wendroff method and could potentially be modified for the AF scheme.

- **Three-dimensional solver for wave equations:** The use of spherical means integral to solve acoustic systems of equations has the advantage of being physically valid when extended to three-dimensions. Some simplifications could be applied if we rotate the three-dimensional element such that one edge lies along the z-axis. For an edge node that lies along this direction, the limits of integration for the partial spherical means formula is quite simple, as illustrated in Figure 6.1(a), where $R = c\Delta t$ is the radius of the sphere and the constraint $\Delta\theta_{PQ} = \theta_Q - \theta_P > 0$ always holds true.

$$M^{3D}_{R,\text{edge}}\{f\}(x_0, y_0, z_0) = \frac{1}{4\pi R^2} \int\limits_0^\pi \int\limits_{\theta_P}^{\theta_Q} f(x_0 + R\sin\phi\cos\theta, y_0 + R\sin\phi\sin\theta, \\ z_0 + R\cos\phi) R^2 \sin\phi \, \mathrm{d}\theta \, \mathrm{d}\phi \tag{6.1}$$

For a vertex node, evaluating the partial integrals is a bit more complex. Referring to Figure 6.1(b), the limits of integration in the polar direction is simply from ϕ_P to ϕ_Q, whereas the limits of integration in the azimuthal direction varies linearly with ϕ. Let $h = R\theta_{\max}$ be the maximum arc length B-B' from the vertical face of the tetrahedral element and ϕ_h be the polar angle at point B', the partial spherical means integral at a vertex node can be evaluated as

$$M^{3D}_{R,\text{vertex}}\{f\}(x_0, y_0, z_0) = \frac{1}{4\pi R^2} \int\limits_{\phi_P}^{\phi_Q} \int\limits_0^{\theta(\phi)} f(x_0 + R\sin\phi\cos\theta, y_0 + R\sin\phi\sin\theta, \\ z_0 + R\cos\phi) R^2 \sin\phi \, \mathrm{d}\theta \, \mathrm{d}\phi \tag{6.2}$$



(a) Edge node.

(b) Vertex node.

Figure 6.1: Partial spherical means integral in 3-D.

where

$$\theta(\phi) = \begin{cases} \dfrac{\theta_{\max}}{\phi_h - \phi_P}(\phi - \phi_P) & \text{if } \phi_P \leq \phi < \phi_h \\ \dfrac{\theta_{\max}}{\phi_Q - \phi_h}(\phi_Q - \phi) & \text{if } \phi_h < \phi \leq \phi_Q \end{cases} \tag{6.3}$$

At the moment, this integral seems to be more complicated to compute than its two-dimensional counterpart, but simplifications are most likely possible once we systematically analyze the quadratic functions required in the three-dimensional AF method.

APPENDIX A

Calculating spherical means using Laplacian point evaluations

Recall that the spherical mean formula can be written as the sum of point evaluations of increasing powers of the Laplacian,

$$M_R = \sum_{p=0}^{\infty} \frac{R^{2p}}{(2p+1)!} \nabla^{2p} \tag{A.1}$$

Since the data is piecewise quadratic in the AF method, all powers of Laplacian beyond the first power should vanish (i.e. $p \geq 3$ terms vanish). The spherical means can be approximated by

$$M_R\{f\} \approx \frac{\theta_{PQ}}{2\pi} \left(f|_p + \frac{R^2}{6} \nabla^2 f|_p + \frac{R^4}{120} \nabla^4 f|_p + \cdots \right) \tag{A.2}$$

where f is any arbitrary function within the domain, $f|_p$ is f evaluated at point p, and $\theta_{PQ} = \theta_Q - \theta_P$. We can then compare the above approximation with the general spherical means integral in two-dimensions,

$$M_R\{f\} = \frac{1}{2\pi R} \int_{\theta_P}^{\theta_Q} \int_0^R f \frac{r}{\sqrt{R^2 - r^2}} dr d\theta \tag{A.3}$$

A.1 Correction to the Laplacian approximation

We would like to determine the differences between the exact spherical means integral and its Laplacian approximation. Let us first consider the spherical means of quadratic functions in Cartesian coordinates:

$$f(x, y) = A + Bx + Cy + Dx^2 + Exy + Fy^2 \tag{A.4}$$

Term	spherical means	
	$M_{R,\text{exact}}$	$M_{R,\text{Laplacian}}$
1	$\theta_{PQ}/2\pi$	$\theta_{PQ}/2\pi$
x	$\frac{R}{8}(\sin\theta_Q - \sin\theta_P)$	0
y	$\frac{R}{8}(\cos\theta_P - \cos\theta_Q)$	0
x^2	$\frac{R^2}{12\pi}(2\theta_{PQ} + \sin 2\theta_Q - \sin 2\theta_P)$	$R^2\theta_{PQ}/6\pi$
xy	$\frac{R^2}{6\pi}(\cos^2\theta_P - \sin^2\theta_Q)$	0
y^2	$\frac{R^2}{12\pi}(2\theta_{PQ} - \sin 2\theta_Q + \sin 2\theta_P)$	$R^2\theta_{PQ}/6\pi$

Term	correction term	
	$\theta_{PQ} = \pi$ (edges)	$\theta_{PQ} \neq \pi$ (vertices)
1	0 (exact)	0 (exact)
x	$-\frac{R}{4}\sin\theta_P$	$\frac{R}{8}(\sin\theta_Q - \sin\theta_P)$
y	$\frac{R}{4}\cos\theta_P$	$\frac{R}{8}(\cos\theta_P - \cos\theta_Q)$
x^2	0 (exact)	$\frac{R^2}{12\pi}(\sin 2\theta_Q - \sin 2\theta_P)$
xy	0 (exact)	$\frac{R^2}{6\pi}(\cos^2\theta_P - \sin^2\theta_Q)$
y^2	0 (exact)	$\frac{R^2}{12\pi}(\sin 2\theta_P - \sin 2\theta_Q)$

Table A.1: Comparison between exact integration and Laplacian approximation of spherical means. The correction term is $M_{R,\text{exact}} - M_{R,\text{Laplacian}}$.

In order to solve for the spherical means exactly, we must convert to polar coordinates assuming the Laplacian is evaluated at point $(x_0, y_0) = (0, 0)$, i.e. $x = r\cos\theta$ and $y = r\sin\theta$. Then, using Equations (A.2) and (A.3), the resulting spherical means are summarized in Table 1. Note that the Laplacian approximation gives the exact solution for both the constant and quadratic terms when considering points along each edge of an element.

A.2 Application to basis functions

The general practice is to work in a reference space of $\xi \in [0, 1]$ and $\eta \in [0, 1]$ in which every element is transformed to a canonical triangle with node 1 starting at the origin and each subsequent node ordered in a counterclockwise direction. The basis functions can easily be determined in the reference space as $\phi_j(\xi, \eta)$, where j corresponds to the local

Term	spherical means	
	$M_{R,\text{exact}}$	$M_{R,\text{Laplacian}}$
x^3	$\frac{R^3}{128}(9\sin\theta_Q + \sin 3\theta_Q - 9\sin\theta_P - \sin 3\theta_P)$	0
x^2y	$\frac{R^3}{32}(\cos^3\theta_P - \cos^3\theta_Q)$	0
xy^2	$\frac{R^3}{32}(\sin^3\theta_Q - \sin^3\theta_P)$	0
y^3	$\frac{R^3}{128}(\cos 3\theta_Q - 9\cos\theta_Q - \cos 3\theta_P + 9\cos\theta_P)$	0
x^4	$\frac{R^4}{120\pi}(12\theta_{PQ} + 8\sin 2\theta_Q + \sin 4\theta_Q - 8\sin 2\theta_P - \sin 4\theta_P)$	$R^4\theta_{PQ}/10\pi$
x^3y	$\frac{R^4}{15\pi}(\cos^4\theta_P - \cos^4\theta_Q)$	0
x^2y^2	$\frac{R^4}{120\pi}(4\theta_{PQ} - \sin 4\theta_Q + \sin 4\theta_P)$	0
xy^3	$\frac{R^4}{15\pi}(\sin^4\theta_Q - \sin^4\theta_P)$	0
y^4	$\frac{R^4}{120\pi}(12\theta_{PQ} - 8\sin 2\theta_Q + \sin 4\theta_Q + 8\sin 2\theta_P - \sin 4\theta_P)$	$R^4\theta_{PQ}/10\pi$

Table A.2: Comparison between exact integration and Laplacian approximation of spherical means for higher order terms.

node number. In addition, we assume the mapping Jacobian, $\boldsymbol{J}$, is linear and constant, and is defined as

$$\boldsymbol{J}(\vec{\xi}) = \frac{\partial \vec{x}}{\partial \vec{\xi}} = \begin{bmatrix} x_3 - x_1 & x_5 - x_1 \\ y_3 - y_1 & y_5 - y_1 \end{bmatrix} \tag{A.5}$$

where $\vec{x} = [x, y]$, $\vec{\xi} = [\xi, \eta]$, and the subscripts $1, 3, 5$ are the vertex nodes of each element.

When determining the spherical mean integral of each basis function, we simply need to remap from the reference space to Cartesian using $\vec{\xi} = \boldsymbol{J}^{-1}\vec{x}$:

$$M_R\{\phi_j\} = \frac{1}{2\pi R}\iint \phi_j(\xi,\eta)\frac{r}{\sqrt{R^2 - r^2}}drd\theta = \frac{1}{2\pi R}\iint \hat{\phi}_j(x,y)\frac{r}{\sqrt{R^2 - r^2}}drd\theta \tag{A.6}$$

A.3 Higher order terms

When dealing with higher-order methods, we may need to consider up to quartic functions (i.e. $p > 4$ terms vanish). The spherical means integrals are again calculated using Equations (A.2) and (A.3), and shown in Table 2.

APPENDIX B

Determining modified coefficients of Euler Equations

Given the Euler equations containing advective terms only, we are left with a nonlinear advection system.

$$\begin{aligned} \rho_t + \mathbf{v} \cdot \nabla \rho = 0 \\ u_t + \mathbf{v} \cdot \nabla u = 0 \\ v_t + \mathbf{v} \cdot \nabla v = 0 \\ p_t + \mathbf{v} \cdot \nabla p = 0 \end{aligned} \tag{B.1}$$

The second-order accurate solution to Equation (B.1) is given by Maeng [39] as

$$\mathbf{q}(\mathbf{0}, t) = \mathbf{q}(\mathbf{x}^*, 0) \tag{B.2}$$

with the characteristic origin

$$\mathbf{x}^* = -t(\mathbf{v}\mathbf{I} - t\mathbf{v} \cdot \nabla \mathbf{v}) \tag{B.3}$$

Taking these advective terms into account, we can simplify the modified coefficients in Equations (4.10) and (4.11) to

$$\mathbf{A}^* = \mathbf{A} - t(\mathbf{v} \cdot \nabla u)\mathbf{I} - t\mathbf{A}^{**} \tag{B.4}$$

$$\mathbf{B}^* = \mathbf{B} - t(\mathbf{v} \cdot \nabla v)\mathbf{I} - t\mathbf{B}^{**} \tag{B.5}$$

where the twice-modified coefficient matrices are

$$\mathbf{A}^{**} = \begin{pmatrix} \frac{1}{2\rho}p_x & -\frac{1}{2}\rho_t + \frac{1}{2}\mathbf{v}\cdot\nabla\rho + \frac{1}{2}\rho u_x & \frac{1}{2}\rho u_y & -\frac{1}{2\rho}\rho_x \\ 0 & \frac{1}{2\rho}p_x + \frac{1}{2\rho}(\rho c^2)_x & 0 & -\frac{1}{\rho^2}\mathbf{v}\cdot\nabla\rho - \frac{1}{2\rho}v_y \\ 0 & \frac{1}{2\rho}(\rho c^2)_y & \frac{1}{2\rho}p_x & \frac{1}{2\rho}u_y \\ 0 & -\frac{1}{2}(\rho c^2)_t + \frac{1}{2}\mathbf{v}\cdot\nabla(\rho c^2) + \frac{1}{2}\rho c^2 u_x & \frac{1}{2}\rho c^2 u_y & \frac{1}{2\rho}p_x - \frac{1}{2\rho}c^2\rho_x \end{pmatrix} \tag{B.6}$$

$$\mathbf{B}^{**} = \begin{pmatrix} \frac{1}{2\rho}p_y & \frac{1}{2}\rho v_x & -\frac{1}{2}\rho_t + \frac{1}{2}\mathbf{v}\cdot\nabla\rho + \frac{1}{2}\rho v_y & -\frac{1}{2\rho}\rho_y \\ 0 & \frac{1}{2\rho}p_y & \frac{1}{2\rho}(\rho c^2)_x & \frac{1}{2\rho}v_x \\ 0 & 0 & \frac{1}{2\rho}p_y + \frac{1}{2\rho}(\rho c^2)_y & -\frac{1}{\rho^2}\mathbf{v}\cdot\nabla\rho - \frac{1}{2\rho}u_x \\ 0 & \frac{1}{2}\rho c^2 v_x & -\frac{1}{2}(\rho c^2)_t + \frac{1}{2}\mathbf{v}\cdot\nabla(\rho c^2) + \frac{1}{2}\rho c^2 v_y & \frac{1}{2\rho}p_y - \frac{1}{2\rho}c^2\rho_y \end{pmatrix} \tag{B.7}$$

We observe from Equations (B.6) and (B.7) that there is a repeated pressure gradient term along the matrix diagonals. These pressure gradients can be interpreted as forces with an acceleration a that displace a fluid particle from its original inertial path by an amount $\frac{1}{2}at$. A modified characteristic origin is then defined as

$$\mathbf{x}^{**} = \mathbf{x}^{*} + t^2\frac{1}{2\rho}\nabla p \tag{B.8}$$

and the advective solution is determined at

$$\mathbf{q}(\mathbf{0}, t) = \mathbf{q}(\mathbf{x}^{**}, 0) \tag{B.9}$$

The modified coefficient matrix can then be expressed as

$$\mathbf{A}^{*} = \mathbf{A} - t\left(\mathbf{v}\cdot\nabla u + \frac{1}{2\rho}p_x\right)\mathbf{I} - t\mathbf{A}^{***} \tag{B.10}$$

$$\mathbf{B}^{*} = \mathbf{B} - t\left(\mathbf{v}\cdot\nabla v + \frac{1}{2\rho}p_y\right)\mathbf{I} - t\mathbf{B}^{***} \tag{B.11}$$

where, after some simplifications, the thrice-modified coefficient matrices become

$$\mathbf{A}^{***} = \begin{pmatrix} 0 & -\frac{1}{2}\rho_t + \frac{1}{2}\mathbf{v}\cdot\nabla\rho + \frac{1}{2}\rho u_x & \frac{1}{2}\rho u_y & -\frac{1}{2\rho}\rho_x \\ 0 & \frac{1}{2\rho}(\rho c^2)_x & 0 & \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right) - \frac{1}{2\rho}v_y \\ 0 & \frac{1}{2\rho}(\rho c^2)_y & 0 & \frac{1}{2\rho}u_y \\ 0 & -\frac{1}{2}(\rho c^2)_t + \frac{1}{2}\mathbf{v}\cdot\nabla(\rho c^2) + \frac{1}{2}\rho c^2 u_x & \frac{1}{2}\rho c^2 u_y & -\frac{1}{2\rho}c^2\rho_x \end{pmatrix} \tag{B.12}$$

$$\mathbf{B}^{***} = \begin{pmatrix} 0 & \frac{1}{2}\rho v_x & -\frac{1}{2}\rho_t + \frac{1}{2}\mathbf{v}\cdot\nabla\rho + \frac{1}{2}\rho v_y & -\frac{1}{2\rho}\rho_y \\ 0 & 0 & \frac{1}{2\rho}(\rho c^2)_x & \frac{1}{2\rho}v_x \\ 0 & 0 & \frac{1}{2\rho}(\rho c^2)_y & \mathbf{v}\cdot\nabla\left(\frac{1}{\rho}\right) - \frac{1}{2\rho}u_x \\ 0 & \frac{1}{2}\rho c^2 v_x & -\frac{1}{2}(\rho c^2)_t + \frac{1}{2}\mathbf{v}\cdot\nabla(\rho c^2) + \frac{1}{2}\rho c^2 v_y & -\frac{1}{2\rho}c^2\rho_y \end{pmatrix} \tag{B.13}$$

Finally, the Lax-Wendroff approximation of the Euler equations can now be written as

$$\begin{aligned} \mathbf{q}^{n+1} = \mathbf{q}^n - t\left\{\left(\mathbf{A} - t(\mathbf{v}\cdot\nabla u + \tfrac{1}{2\rho}p_x)\mathbf{I}\right)\partial_x\mathbf{q} + \left(\mathbf{B} - t(\mathbf{v}\cdot\nabla v + \tfrac{1}{2\rho}p_y)\mathbf{I}\right)\partial_y\mathbf{q}\right\} \\ + \frac{1}{2}t^2\left\{\mathbf{A}^2\partial_{xx}\mathbf{q} + (\mathbf{AB} + \mathbf{BA})\partial_{xy}\mathbf{q} + \mathbf{B}^2\partial_{yy}\mathbf{q}\right\} \\ + t^2(\mathbf{A}^{***}\partial_x\mathbf{q} + \mathbf{B}^{***}\partial_y\mathbf{q}) \end{aligned} \tag{B.14}$$

BIBLIOGRAPHY

[1] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *Int. J. Numer. Meth. Fluids*, Vol. 72, 2013, pp. 811–845.

[2] Wendt(Ed.), J. F., *Computational fluid dynamics: an introduction*, Springer-Verlag, Berlin, 2009.

[3] Hirsch, C., *Numerical computation of internal and external flows: fundamentals of fluid dynamics*, Butterworth-Heinemann, Oxford, 2nd ed., 2007.

[4] Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Wigton, L. B., and Young, D. P., "Algorithm issues and challenges associated with the development of robust CFD codes," *Variational Analysis and Aerospace Engineering*, Springer, 2009, pp. 1–19.

[5] Roe, P. L., "A Simple Explanation of Superconvergence for Discontinuous Galerkin Solutions to u t+ u x=0," *Comm. Comp. Phys.*, Vol. 21, No. 4, 2017, pp. 905–912.

[6] Godunov, S. K., "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Mat. Sb.*, Vol. 89, No. 3, 1959, pp. 271–306.

[7] Meier, D. L., "Constrained transport algorithms for numerical relativity. i. development of a finite-difference scheme," *Astrophys. J.*, Vol. 595, No. 2, 2003, pp. 980–991.

[8] Mishra, S. and Tadmor, E., "Constraint preserving schemes using potential-based fluxes. I. Multidimensional transport equations," *Comm. Comp. Phys.*, Vol. 9, 2010, pp. 688–710.

[9] Tóth, G., "The $\nabla \cdot B$=0 constraint in shock-capturing magnetohydrodynamics codes," *J. Comp. Phys.*, Vol. 161, No. 2, 2000, pp. 605–652.

[10] Jiang, G.-S. and Shu, C.-W., "Efficient implementation of weighted eno schemes," *J. Comp. Phys.*, Vol. 126, No. 1, 1996, pp. 202–228.

[11] Park, S. H. and Kwon, J., "An improved HLLE method for hypersonic viscous flows," Paper 2001-2633, AIAA, June 2001.

[12] Colella, P., "Multidimensional upwind methods for hyperbolic conservation laws," *J. Comp. Phys.*, Vol. 87, No. 1, 1990, pp. 171–200.

[13] Davis, S. F., "A rotationally biased upwind difference scheme for the Euler equations," *J. Comp. Phys.*, Vol. 56, 1984, pp. 56–92.

[14] Levy, D. W., Powell, K. G., and van Leer, B., "Use of a rotated Riemann solver for the two-dimensional Euler equations," *J. Comp. Phys.*, Vol. 106, 1993, pp. 201–214.

[15] LeVeque, R. J., "High-resolution conservative algorithms for advection in incompressible flow," *SIAM J. Numer. Anal.*, Vol. 33, No. 2, 1996, pp. 627–665.

[16] Roe, P. L., "Did numerical methods for hyperbolic problems take a wrong turning?" In print.

[17] Roe, P. L., "Is discontinuous reconstruction really a good idea?" In print.

[18] Roe, P. L., "Fluctuations and signals, a framework for numerical advection problems," *Numerical Methods for Fluid Dynamics*, Academic Press, 1982, pp. 219–257.

[19] Rad, M., *A residual distribution approach to the Euler equations that preserves potential flow*, Ph.D. thesis, University of Michigan, 2001.

[20] Nishikawa, H., Rad, M., and Roe, P. L., "A third-order fluctuation splitting scheme that preserves potential flow," Paper 2001-2595, AIAA, June 2001.

[21] Abgrall, R. and Mezine, M., "Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems," *J. Comp. Phys.*, Vol. 188, 2003, pp. 16–55.

[22] Reddy, A. S., Tikekar, V. G., and Prasad, P., "Numerical solution of hyperbolic equations by method of bicharacteristics," *J. Math. Phys. Sci.*, Vol. 16, No. 6, 1982, pp. 575–603.

[23] Alpert, B., Greengard, L., and Hagstrom, T., "An Integral Evolution Formula for the Wave Equation," *J. Comput. Phys.*, Vol. 162, 2000, pp. 536–543.

[24] Li, J.-R. and Greengard, L., "Strongly consistent marching schemes for the wave equation," *J. Comp. Phys.*, Vol. 188, No. 1, 2003, pp. 194–208.

[25] Hagstrom, T., "High-resolution difference methods with exact evolution for multidimensional waves," *Appl. Numer. Math.*, Vol. 93, 2015, pp. 114–122.

[26] Strang, G., "On the construction and comparison of difference schemes," *SIAM J. Numer. Anal.*, Vol. 5, No. 3, 1968, pp. 506–517.

[27] Morton, K. W. and Roe, P. L., "Vorticity-preserving Lax-Wendroff-type schemes for the system wave equation," *SIAM J. Sci. Comput.*, Vol. 23, No. 1, 2001, pp. 170–192.

[28] Lung, T. B. and Roe, P. L., "Toward a reduction of mesh imprinting," *Int. J. Numer. Meth. Fluids*, Vol. 76, No. 7, 2014, pp. 450–470.

[29] van Leer, B., "Toward the ultimate conservative differencing scheme, IV: a new approach to linear convection," *J. Comp. Phys.*, Vol. 23, 1977, pp. 276–299.

[30] Eymann, T. A., *Active Flux Schemes*, Ph.D. thesis, University of Michigan, 2013.

[31] Courant, R. and Hilbert, D., *Methods of mathematical physics, volume II*, Interscience, New York, 1962.

[32] Whitham, G. B., *Linear and nonlinear waves*, Wiley-Interscience, New York, 1974.

[33] Lukáčová-Medivid'ová, M., Morton, K. W., and Warnecke, G., "Evolution Galerkin methods for hyperbolic systems in two space dimensions," *Math. Comput.*, Vol. 69, No. 232, 2000, pp. 1355–1384.

[34] Hu, F. Q., "A Stable, Perfectly Matched Layer for Linearized Euler Equations in Unsplit Physical Variables," *J. Comp. Phys.*, Vol. 173, 2001, pp. 455–480.

[35] Higdon, R. L., "Numerical absorbing boundary conditions for the wave equation," *Math. Comput.*, Vol. 49, No. 179, 1987, pp. 65–90.

[36] Giles, M. B., "Non-reflecting boundary conditions for Euler equation calculations," *AIAA J.*, Vol. 28, No. 12, 1990, pp. 2050–2058.

[37] Thompson, K., "Time dependent boundary conditions for hyperbolic systems," *J. Comp. Phys.*, Vol. 68, 1987, pp. 1–24.

[38] Smoller, J., *Shock waves and reaction-diffusion equations*, Springer-Verlag, Berlin, 2nd ed., 1994.

[39] Maeng, J. B., *On the advective component of Active Flux schemes for nonlinear hyperbolic conservation laws*, Ph.D. thesis, University of Michigan, 2017.

[40] Zha, G.-C. and Bilgen, E., "Numerical solutions of Euler equations by using a new flux vector splitting scheme," *Int. J. Numer. Meth. Fluids*, Vol. 17, No. 2, 1993, pp. 115–144.

[41] Xiao, F., Akoh, R., and Ii, S., "Unified formulation for compressible and incompressible flows by using multi-integrated moments II: Multi-dimensional version for compressible and incompressible flows," *J. Comp. Phys.*, Vol. 213, No. 1, 2006, pp. 31–56.

[42] Kwatra, N., Su, J., Grétarsson, J. T., and Fedkiw, R., "A method for avoiding the acoustic time step restriction in compressible flow," *J. Comp. Phys.*, Vol. 228, No. 11, 2009, pp. 4146–4161.

[43] Lyness, J. and Jespersen, D., "Moderate degree symmetric quadrature rules for the triangle," *J. Inst. Maths Applics*, Vol. 15, 1995, pp. 19–32.

[44] Shu, C.-W., “High Order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD,” Report 2001-210865, NASA, May 2001.

[45] Carette, J. C., Deconinck, H., Paillere, H., and Roe, P. L., “Mulitdimensional upwinding: Its relation to finite elements,” *Int. J. Numer. Meth. Fluids*, Vol. 20, 2013, pp. 935–955.

[46] Nishikawa, H., “Active Flux schemes for advection diffusion,” Paper 2015-2450, AIAA, June 2015.

[47] Roe, P. L., Lung, T. B., and Maeng, J., “New approaches to limiting,” Paper 2015-2913, AIAA, June 2015.