Residual-based Discretization Error Estimation for Computational Fluid Dynamics

Tyrone S. Phillips

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Christopher J. Roy
Eugene M. Cliff
William H. Mason
Jeffrey T. Borggaard
Danesh K. Tafti

September 24, 2014
Blacksburg, Virginia

Keywords: CFD, solution reconstruction, discretization error, truncation error, defect correction, error transport equations

Residual-based Discretization Error Estimation for Computatioanl Fluid Dynamics

Tyrone S. Phillips

(ABSTRACT)

The largest and most difficult numerical approximation error to estimate is discretization error. Residual-based discretization error estimation methods are a category of error estimators that use an estimate of the source of discretization error and information about the specific application to estimate the discretization error using only one grid level. The higher-order terms are truncated from the discretized equations and are the local source of discretization error. The accuracy of the resulting discretization error estimate depends solely on the accuracy of the estimated truncation error. Residual-based methods require only one grid level compared to the more commonly used Richardson extrapolation which requires at least two. Reducing the required number of grid levels reduces computational expense and, since only one grid level is required, can be applied to unstructured grids where multiple quality grid levels are difficult to produce. The two residual-based discretization error estimators of interest are defect correction and error transport equations. The focus of this work is the development, improvement, and evaluation of various truncation error estimation methods considering the accuracy of the truncation error estimate and the resulting discretization error estimates. The minimum requirements for accurate truncation error estimation is specified along with proper treatment for several boundary conditions. The methods are evaluated using various Euler and Navier-Stokes applications. The discretization error estimates are compared to Richardson extrapolation. The most accurate truncation error estimation method was found to be the k-exact method where the fine grid with a correction factor was considerably reliable. The single grid methods including the k-exact require that the continuous operator be modified at the boundary to be consistent with the implemented boundary conditions. Defect correction showed to be more accurate for areas of larger discretization error; however, the cost was substantial (although cheaper than the primal problem) compared to the cost of solving the ETEs which was essential free due to the linearization. Both methods showed significantly more accurate estimates compared to Richardson extrapolation especially for smooth problems. Reduced accuracy was apparent with the presence of stronger shocks and some possible modifications to adapt to singularies are proposed for future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fluid dynamics is described by a set of Partial Differential Equations (PDEs) known as the Navier-Stokes equations. These equations are derived from basic conservation principles (conservation of mass, momentum, and energy) resulting in a set of five equations plus an equation of state. The Navier-Stokes equations, in theory, can be used to solve any fluid flow problem. Analytic solutions to the Navier-Stokes equations have been found only using assumptions that reduce the complexity of the equations considerably which removes general applicability. Computational Fluid Dynamics (CFD) is a tool used to compute approximate solutions to the Navier-Stokes equations with varying degrees of limiting assumptions and sub-models and has enormous potential to impact the analysis, design, and optimization of engineering systems. The numerical solution to the Navier-Stokes equations requires that the fluid domain be discretized into a computational mesh. Due to limitations in computational resources, the error due to the spatial resolution of the mesh is often the largest and most difficult to estimate numerical error (i.e. discretization error).

In general, the discrete solution to differential equations can be written using the Generalized Truncation Error Expression (GTEE)[1,2]

$$L_h(\cdot) = L(\cdot) + \tau_h(\cdot). \tag{1.1}$$

The original continuous PDEs are represented by $L(\cdot)$ where $\tilde{u}$ is an exact solution which exactly satisfies the PDEs so that $L(\tilde{u}) = 0$. The discrete equations are represented by $L_h(\cdot)$. The discrete equations are solved on a mesh with cell size h so that $L_h(u_h) = 0$ where u_h is the exact solution to the discrete equations. The difference between the exact solution and the numerical solution is the discretization error

$$\varepsilon_h = u_h - \tilde{u}. \tag{1.2}$$

The error due to the approximations used in the derivation of the discrete equations $L_h(\cdot)$ is the truncation error $\tau_h(\cdot)$, and is a function of solution derivatives, grid related derivatives, and cell size. For example, truncation error can be written to take the general form $\tau_h(u) = c_1(x,y,z,t)h^{p_f} + c_2(x,y,z,t)h^{p_f+1} + O(h^{p_f+2})$. For sufficiently small cell size, the leading term dominates the error

such that $c_1(x,y,z,t)h^{p_f} >> c_2(x,y,z,t)h^{p_f+1} + O(h^{p_f+2})$ and the error asymptotically approaches zero at the rate p_f (i.e. formal order of accuracy) which is chosen by design during the derivation of the discrete equations. By manipulating the GTEE, the truncation error can be shown to be the local source of discretization error

$$L_h(\varepsilon_h) = -\tau_h(\tilde{u}). \tag{1.3}$$

The discretization error is convected and diffused through the domain in the same manner as the solution. The resulting discretization error is the net total of locally generated error and error transported from the rest of the flow domain. From Equation 1.3, the discretization error is directly related to the truncation error. In generally, the discretization error is expected to decrease at the same rate as the leading truncation error term when the solution is in the asymptotic range and for smooth grids.

The basis of defect correction discretization error estimation relies on adding the truncation error as a source term which removes the local source of error related to the discretization of the domain and governing equations. If the exact truncation error is added as a source term then the resulting numerical solution is the exact solution to the governing equations

$$L_h(\tilde{u}) = \tau_h(\tilde{u}). \tag{1.4}$$

The exact truncation error can only be computed if the exact solution is known. Instead the truncation error is estimated using the original numerical solution u_h and results in an approximatation of the exact solution to the governing equations

$$L_h(\bar{u}_h) = \tau_h(u_h) \tag{1.5}$$

where $\tau_h(u_h) \approx \tau_h(\tilde{u})$ and $\bar{u}_h \approx \tilde{u}$. The discretization error in the numerical solution is estimated as $\varepsilon_h \approx u_h - \bar{u}_h$. The accurate estimation of truncation error is the single most important aspect of residual-based discretization error estimation methods.

1.1 Discretization Schemes

Three different methods of discretizing PDEs are commonly used and include finite element, finite difference, and finite volume methods. Finite difference and finite volume methods are commonly used for fluid flow where finite element methods are more commonly associated with structural mechanics. The discretization scheme of interest in the current research is the finite volume method because flow discontinuities are more easily handled making the discretization scheme more appropriate for compressible fluid dynamics.

Using fundamental physical principles, the governing equations are derived with an assumption of element size. Using an infinitesimal control volume (e.g. $\lim_{\Delta x \to 0} \Delta f/\Delta x = df/dx$) leads to the differentiable form (strong form) of the governing equations which, assuming differential fluxes,

takes the general form

$$\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \bar{\bar{F}} = \overline{Q} \tag{1.6}$$

and is naturally solved using a finite difference method. Using a finite control volume leads to the integral form (weak form) of the equations which takes the general form

$$\frac{\partial}{\partial t} \iiint_V \vec{U} dV + \iint_A \bar{\bar{F}} \cdot d\vec{A} = \iiint_V \vec{S} dV \tag{1.7}$$

In Equations 1.6 and 1.7, $\bar{\bar{F}}$ is a tensor of fluxes, $\vec{U}$ is a vector of conserved variables, and $\vec{Q}$ is a vector of source terms. Both methods result in a discrete solution; however, the fundamental differences in the governing equations result in fundamentally different representations in the numerical solution. A finite difference solution represents the solution at the i-th grid node location with grid spacing h as

$$f_{i,h} = f(x_i), \tag{1.8}$$

and a finite volume solution represents the average solution over the i-th cell with cell size h as

$$f_{i,h} = \frac{1}{V_i} \int_{V_i} f(x) dV. \tag{1.9}$$

An example comparing the different solution representations is shown in Figure 1.1. This fundamental difference plays an important role in the methods used for solution reconstruction. The finite volume numerical solution can be treated as a solution existing at the cell-center which is accurate in the limit as cell volume approaches zero (i.e. grid refinement). If the solution is assumed to exist at the geometric average of the cell nodes, solution reconstruction is limited to second-order accuracy. For arbitrary higher order accuracy, solution reconstruction should be consistent with the finite volume representation of the solution in Equation 1.9, that is, the average of the reconstruction over the cell should reproduce the finite volume solution.

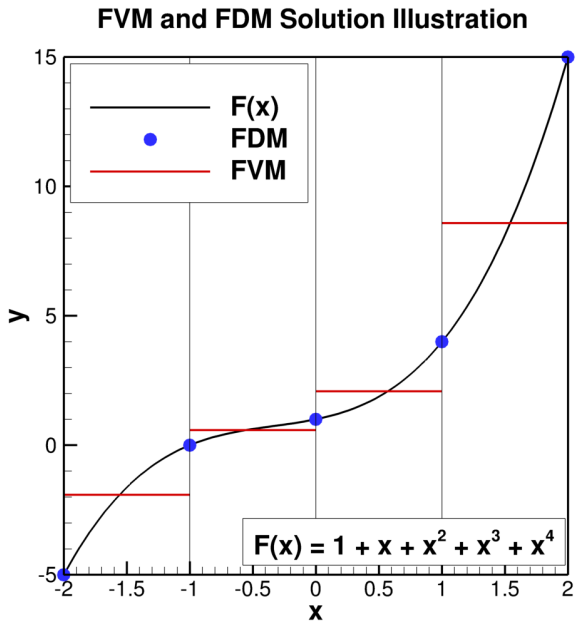


Figure 1.1: Finite volume and finite difference solution representations

Steady state Burgers' equation can be used to illustrate the finite-volume discretization scheme. Burgers' equation is a differential equation with a nonlinear convective term and a linear diffusive term which is considered analogous to the Navier-Stokes equations. The weak form of one-dimensional Burgers' equation is

$$L(u) = \frac{\partial u}{\partial t} + \iint_S F(x) dS = 0 \tag{1.10}$$

where the flux is

$$F(x) = \frac{u^2}{2} - \nu \frac{\partial u}{\partial x}.$$

The discrete equation is

$$L_h(u) = \frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{F_{i+1/2}^n - F_{i-1/2}^n}{\Delta x} = 0 \tag{1.11}$$

where a discrete second-order accurate central flux is

$$F_{i+1/2} = F_h(u_{i+1}, u_i) = \frac{u_{i+1}^2 - u_i^2}{4} - \nu \frac{u_{i+1} - u_i}{\Delta x}.$$

An exact solution to Burgers' equation is

$$\tilde{u}(x) = -u_{ref} tanh\left(x\frac{Re}{2L}\right) \tag{1.12}$$

where $Re = u_{ref}L/\nu$, $L = 8\,m$, and $u_{ref} = 2\,m/s$. The leading truncation error terms for this second-order discretization scheme on a uniformly spaced grid are

$$\tau_h(u) = \Delta x^2\left(\frac{3}{8}u_x u_{xx} + \frac{1}{8}uu_{xxx} - \frac{1}{24}\nu u_{xxxx}\right) + O(\Delta x^4). \tag{1.13}$$

For simple discretization schemes, the leading truncation error terms can be derived as shown in Equation 1.13; however, as the schemes and governing equations become more complex the more difficult the expression is to derive. Taking into account truncation error on a non-uniform grid, the number of terms for Burgers' equation increases significantly.

1.2 Discretization Error Estimation

1.2.1 Richardson Extrapolation

The most widely used discretization error estimator is Richardson extrapolation which uses a series of solutions on increasingly finer grid resolution to construct a higher order accurate solution. Richardson extrapolation is simple to apply as a post processing step and does not require code modification. The disadvantage of Richardson extrapolation is the requirement of multiple grid levels, since grid generation is one of the most time consuming tasks for a numerical simulation. In addition to the expense of generating multiple grid levels, each grid level used in Richardson extrapolation must also be in the asymptotic range in order to compute an accurate error estimate.

For smooth solutions with no discontinuities, the exact solution can be written in terms of a power series expansion, $\tilde{u} = u_h - \sum_{p=p_f}^{\infty}\alpha_p h^p$, which can be used to rewrite Eq. 1.2 as

$$\varepsilon_h = \sum_{p=p_f}^{\infty}\alpha_p h^p = \alpha_{p_f}h^{p_f} + HOT. \tag{1.14}$$

To reasonably approximate the discretization error, the Higher Order Terms (HOT) are dropped reducing Eq. 1.14 to

$$\varepsilon_h \approx \bar{\varepsilon}_h = \alpha_{p_f}h^{p_f}. \tag{1.15}$$

The exclusion of the HOT introduces the assumption that numeric solutions are asymptotic (i.e., $HOT \ll \alpha_{p_f}h^{p_f}$) for the estimated discretization error, $\bar{\varepsilon}_h$, to accurately approximate the true discretization error, ε_h. Richardson extrapolation is formulated to solve for the unknowns in Eq. 1.15,

α_{p_f} and $\bar{\varepsilon}_h$ resulting in

$$\bar{\varepsilon}_h = \frac{u_{rh} - u_h}{r^{p_f} - 1} \tag{1.16}$$

where r is the refinement factor and is a constant greater than one and is the relative size of the grid spacing on the coarse grid and the fine grid. The most common and easist to implement refinement factor is two which requires removing every other grid node. The expensive of Richardson extrapolation increases rapidly with more complex problems. For example, there is a factor of eight difference in grid nodes between the coarse and fine mesh for a three dimensional simulation. Richardson extrapolation is easy to implement and is the basis for comparison for the residual-based methods because of its widespread use.

1.2.2 Residual-based Methods

Residual-based methods use a discrete solution combined with information about the PDE or discrete equations being solved to estimate the discretization error. Residual-based methods require only one grid level which reduces the grid generation requirements. The methods have the potential to be more accurate than Richardson extrapolation as the residual is an estimate of the truncation error which is directly related to the solution, the discretization scheme, and the grid. If the exact truncation error is used then the exact solution to the PDEs should be recovered.

The residual-based methods of interest are defect correction and the error transport equations (ETEs). Defect correction is fairly straight forward to implement. The truncation error is added as a source term (constant if the solution is steady state), and the cost is a second numerical solution (although the primal solution can be used to provide a very good starting solution). The error transport equations are derived by linearizing the original governing equations about a solution so that the discretization error is directly computed. In addition to estimating the truncation error, the error transport equations must be derived specifically for the discretization scheme.

Defect Correction

Defect correction is derived by inserting the exact solution into the GTEE (Equation 1.1). The exact solution exactly satisfies the continuous operator $L(\tilde{u}) = 0$ which reduces the GTEE to Equation 1.4. Since the exact truncation error is not know, the estimated truncation error is used instead

$$L_h(\bar{u}_h) = \tau_h(I_h u_h). \tag{1.17}$$

The operator I is an interpolation operator and was previously ommited for simplicity. The interpolation operator is used for both restriction to a discrete space and a prolongation to continuous space and is more generally written as I_a^b where a is the starting space and b is the target space. The discrete space h represents a solution on a mesh with spacing h and a blank sub/superscript represents a continuous space with infinite degrees of freedom. In Equation 1.17, the operator I_h

prolongs the discrete solution to a continuous space because the truncation error operator $\tau_h(\cdot)$ operates on a continuous function but has a discrete output because it is specific to the computational grid. In general, the truncation error operator is not know except for very simple discretization schemes and is used instead to represent a truncation error estimate for the numerical solution u_h.

As the estimated discretization error $\tau_h(I_h u_h)$ approaches the exact discretization error $\tau_u(\tilde{u})$, the defect correction solution $\bar{u}_h$ approaches the exact solution $\tilde{u}$. The discretization error in u_h is estimated by taking the difference between u_h and $\bar{u}_h$

$$\bar{\varepsilon}_h = u_h - \bar{u}_h.$$

Error Transport Equations

The error transport equations must be derived specifically for each discretization scheme. A general derivation can be shown using the GTEE. (The derivation of the ETEs also relates discretization error and truncation error which was stated without proof previously). Inserting the exact solution into the GTEE results in the same expression given for the defect correction in Equation 1.4

$$L_h(I^h\tilde{u}) = \tau_h(\tilde{u}).$$

Subtracting $L_h(u_h)$ from the left-hand side results in

$$L_h(u_h) - L_h(I^h\tilde{u}) = -\tau_h(\tilde{u}). \tag{1.18}$$

If the discrete operator is linear then $L_h(u_h) - L_h(I^h\tilde{u}) = L_h(\varepsilon_h)$, otherwise the discrete operator must be linearized. Using Taylor's series

$$L_h(I^h\tilde{u}) = L_h(u_h) - \bar{\varepsilon}_h\frac{\partial L_h(u_h)}{\partial u} + \frac{1}{2}\bar{\varepsilon}_h^T\frac{\partial^2 L_h(u_h)}{\partial u^2}\bar{\varepsilon}_h - O(||\bar{\varepsilon}_h^3||) \tag{1.19}$$

the final ETE reduces to

$$\frac{\partial L_h(u_h)}{\partial u}\bar{\varepsilon}_h = -\tau_h(I_h u_h) + O(||\bar{\varepsilon}_h^2||). \tag{1.20}$$

The derivative is the left-hand size of the linearized discrete operator which is required to implement an implicit time marching method. The left-hand side is commonly available in flow solvers; however, the left-hand side required to solve the ETEs must be the full higher order (e.g. second-order) left-hand side where only the first-order left-hand side is required for an implicit solve. The ETE is approximate so the exact discretization error may not result if the exact truncation error is used; however, the error in Equation 1.20 decreases at a much higher rate and would introduce non-negliglbe error only in the case of very large discretization errors.

Using Burgers' equation as an example, the ETE flux is

$$F_h(\varepsilon_{i+1}, \varepsilon_i, u_{i+1}, u_i) = \frac{u_{i+1}\varepsilon_{i+1} + u_i\varepsilon_i}{2} - \nu\frac{\varepsilon_{i+1} - \varepsilon_i}{\Delta x} + HOT \tag{1.21}$$

where the higher order error term is

$$HOT = -\frac{\varepsilon_{i+1}^2 + \varepsilon_i^2}{4}.$$

1.3 Analysis

In general, the focus of the discretization error estimation is local solution estimation. The local solution variables for the compressible Euler and Navier-Stokes equations are density, pressure, and velocity components. To evaluate the accuracy, contour plots are used; however, for a more quantitative analysis discrete L_1- and L_2-norms are used. The L_1-norm is used when shocks are present in the solution

$$||\varepsilon_h||_{L_1} = \frac{1}{N}\sum_{i=1}^{N}|\varepsilon_i| \tag{1.22}$$

and the L_2-norm is used otherwise

$$||\varepsilon_h||_{L_2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2}. \tag{1.23}$$

To compare the estimated errors to the exact errors, the effectivity index is used[3]

$$\theta_{L_2} = \frac{||\bar{\varepsilon}_h||_{L_2}}{||\varepsilon_h||_{L_2}} \tag{1.24}$$

where $\bar{\varepsilon}_h$ is an estimated error and ε_h is the exact error. If the estimated error is accurate $\theta_{L_2} \approx 1$. For asymptotically accurate error estimates the effectivity index should approach one as the computational grid is refined. The effectivity index is used to evaluate local truncation error and local discretization error estimates for L_1- and L_2-norms where appropriate.

1.4 General Outline

The accuracy of residual-based discretization error estimation methods depends almost solely on the accuracy of the truncation error estimate and is therefore the focus of this work. Chapter 3 focuses on truncation error estimation for the interior of the domain. Boundary conditions are not included and a survey of several previously used, modified, and new truncation error estimation methods are evaluated. The minimum requirements for asymptotic accuracy are determined for all methods (i.e., does the truncation error estimation method accurately estimate the exact truncation error as cell size approaches zero). Chapter 4 extends truncation error estimation to boundary conditions. The implementation of boundary conditions for truncation error accurate truncation error

estimation for inflow, outflow, farfield, and slip-wall boundary conditions are determined as well as the resulting effect on discretization error estimation. Chapter 5 develops numerical benchmark solutions (numerical solutions on significantly finer grids with relatively negligible numerical errors) for a subsonic airfoil, transonic airfoil, and supersonic inlet, and a supersonic cone. The previous development and evaluation of residual-based discretization error estimation methods included only simple, smooth solutions. To determine the efficacy of the newly developed and improved truncation error estimation methods, more realistic applications are required. The numerical errors in the benchmark solutions are estimated and the affects on the error evaluation grid levels are determined (i.e., coarser grids intented for truncation error and discretization error estimation). Chapter 6 uses the previously developed numerical benchmarks to evaluate various truncation error and discretization error methods. The residual-based methods include defect correction and the error transport equations compared to the more commonly used Richardson extrapolation for all cases. Chapter 7 concludes with a discussion of the results, recommendations for usage, and suggestions for future work.

Chapter 2

Literature Review

2.1 Defect Correction

Limited work has been done in the area of defect correction. Initial work was done by Pereyra.[4] The initial intent of the method was to iteratively correct the original numerical solution which in theory would converge to the exact solution. Peryra found that one to two iterations could be computed before the error in the solution began to diverge. Stetter[5] generalized Pereyra's work along with several other related concepts. The correction used a solution which was known and was related to the original numerical solution. If the nearby problem is close enough to the numerical solution the discretization errors should be similar and the discretization error can be estimated or a correction can be computed to correct the numerical solution. Limited work was done on defect correction (also refered to as iterated deferred correction or difference correction) for CFD until 2009. Kurzen et al.[6] used a global spline fit to generate one-dimensional and two-dimensional functions based on a numerical solution. The technique is referred to as the method of nearby problems. The global spline fit was treated as a known exact solution and a second discrete solution was computed with appropriate source terms (similar to the method of manufactured solutions[7]) to compute the error in the nearby problem.

A slightly different method of defect correction was used by Naumovich et al.[8] which was a discrete approach. Instead of finding or creating a nearby solution to estimate the discretization error, the numerical solution is passed into a higher order discretization scheme to compute a residual. The original problem is solved with the residual and results in a higher order accurate numerical solution. The work by Noumovich et al.[8] computed first-order numerical solutions and used defect correction to correct the solution to second-order accuracy. The motivation is the relatively quick first-order solutions due to better conditioned linear systems. The result is a quicker more stable solution which is second-order accurate.

2.2 Error Transport Equations

Significantly more work has been done recently developing methods for solving the error transport equations. Work by Zhang et al.[9] solved the continuous error transport equations with a residual that is estimated using truncation error terms that are derived from Taylor's series expansions. The results applied to one-dimensional linear and nonlinear problems resulted in qualitatively accurate error estimates. When compared to three other estimators by Zhang et al.,[10] the error transport equation was significantly outperformed by Richardson extrapolation. For a supersonic test problem, the error transport equations over-estimated the discretization error by a constant and for the subsonic test problem, under-estimated the discretization error by a constant. The constant offset is due to the accuracy of the truncation error estimate using the analytic truncation error expression referred to as the modified equation.

Celik and Hu[11] developed a general linear error transport equation where the residual is derived using Taylor's series expansions. The results were compared to exact solutions and the method performed very well for linear problems computed on evenly spaced grids but the discretization error was constantly underpredicted for nonlinear problems. The constant offset was later identified by Phillips and Roy[12] to be caused by improperly linearized error equations. Qin and Shih[13] corrected the linearization for nonlinear problems by subtracting a constant based on the linearized error transport equations; however, residual estimation accuracy resulted in poor discretization error estimates for nonlinear problems.

Qin et al.[14,15] develop the error transport equation for the two-dimensional Euler equations. They show that for their linearization method, using the exact residual will result in the exact discretization error. The error transport equations were solved for two-dimensional and three-dimensional solutions to the Euler equations by Cavallo and Sinha.[16,17] Applications include a subsonic two-dimensional airfoil, a UCAV 1303 configuration, and a circular jet. The discretization error estimation results were compared to experimental data when available with no comparisons to numerical benchmarks to assess the accuracy of the estimates. Cavallo et al.[18] extended the error transport equations to turbulent flows. The discretization error estimates were evaluated by comparisons to experimental data or Richardson extrapolation estimates.

All past work on the error transport equations solve error equations that have been directly linearized or an equation set that is quasi-linear. Banks et al.[19] showed that a nonlinear error transport equation may be required for solutions with discontinuities. Using Burgers' equation, it was shown that higher order terms that are dropped during the linearization are important to manage the transport error when shocks or other singularities are present.

2.3 Truncation Error Estimation

The most commonly used truncation error estimation method uses derived residual terms using Taylor's series expansions. This method is refered to as the modified equation. The modified

equation is easily derived for simple one-dimensional and two-dimensional scalar problems such as the wave equation, advection diffusion equation, and Burgers' equation. See for example Qin and Shih.[13] Qin and Shih[20] showed that the accuracy of modified equation degrades rapidly as the grid diverges from evenly spaced nodes. One of the most commonly used approaches to estimate the truncation error for the Euler and Navier-Stokes equations proposed by Zhang et al.[9] is similar and is based on the upwindng terms.

Shih and Williams[21] recognized the failure of the commonly used residual estimation methods and began comparing and developing different methods. They proposed a multiple grid method that requires two or three solutions and interpolates the coarse grid residual to a fine grid. It was noted that this method tended to over predict the discretization error but outperformed the modified equation approach used by Zhang et al.[9]

Other methods include a multiple grid method used by Venditti and Darmafal[22] which used the discrete residual of the numerical solution interpolated on to a finer grid. A similar approach used by Fulton[23] estimated truncation error by interpolating a finer grid solution onto a coarser grid for finite difference methods. The estimate was corrected based on the assumption of an order of accuracy for the truncation error. Fraysse et al.[24] extended Fulton's work to finite volume methods and included the effects of iterative convergence error for numerical solutions which are not fully converged.

Chapter 3

Truncation Error Estimation

Finite Volume Solution Reconstruction Methods For Truncation Error Estimation

Tyrone S. Phillips, Joseph M. Derlaga, Christopher J. Roy,
and Jeff Borggaard

Virginia Tech, Blacksburg, Virginia, 24061

The numerical solution to differential equations results in a discrete solution space for finite volume and finite difference discretization methods. For various reasons, it can be necessary to prolong the solution from a discrete space to a continuous space. The prolongation to a continuous space can be done using various curve-fitting methods which adds an additional level of approximation to the solution. The allowable error of a prolongation operation depends on the specific task required. In this paper we investigate various prolongation methods and identify the minimal requirements specifically for the purpose of truncation error estimation for finite volume methods, where the truncation error is the difference between the discrete and integral governing equations. The reconstruction methods investigated include k-exact and ENO methods. Truncation error estimation for 1D Burgers' equation using the k-exact method suggests that the minimum polynomial order is dependent on the highest derivatives in the truncation error expression and, therefore, the discretization scheme. The effect of different reconstruction methods on truncation error estimation is investigated and the minimum polynomial order for accurate truncation error estimation for the Euler and Navier-Stokes equations is found to be a second-order polynomial reconstruction for fully upwinded MUSCL extrapolation.

3.1 Introduction

Finite volume and finite difference methods are two fundamentally different discretization methods used in the numerical solution to partial differential equations. The two different discretization methods result in discrete, approximate solutions to the original equations. (Finite element methods are also commonly used; however, the resulting approximate numerical solution is continuous.) The differences between the finite volume and finite difference methods arise from fundamentally different formulations of the governing equations. Using fundamental physical principles, the governing equations are derived with an assumption of element size. Using an infinitesimal element (e.g. $\lim_{\Delta x \to 0} \Delta f/\Delta x = df/dx$) leads to the differentiable form (strong form) of the governing equations which, assuming differential fluxes, takes the general form

$$\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \bar{\bar{F}} = \overline{Q} \tag{3.1}$$

and is naturally solved using a finite difference method. A finite control volume formulation uses the integral form (weak form) of the equations which, after application of the divergence theorem, takes the general form

$$\frac{\partial}{\partial t}\int_V \vec{U}dV + \oint_A \bar{\bar{F}}\cdot d\vec{A} = \int_V \vec{S}dV \tag{3.2}$$

In Equations 3.1 and 3.2, $\bar{\bar{F}}$ is a tensor of fluxes, $\vec{U}$ is a vector of conserved variables, and $\vec{Q}$ is a vector of source terms (See Hirsch[1] for more details). Both methods result in a discrete solution; however, the differences in the governing equations result in fundamentally different representations of the numerical solution. A finite difference solution represents the solution at the i-th grid node location with grid spacing h as

$$f_{i,h} = f(x_i), \tag{3.3}$$

and a finite volume solution represents the average solution over the i-th cell with width h as

$$f_{i,h} = \frac{1}{V_i}\int_{V_i} f(x)dV. \tag{3.4}$$

This fundamental difference plays an important role in the form of the reconstruction. An example comparing the different solution representations is shown in Figure 3.1. The finite volume numerical solution can be treated as a solution existing at the cell-center which is accurate in the limit as cell volume approaches zero (i.e. grid refinement). If the solution is assumed to exist at the geometric average of the cell nodes, solution reconstruction is limited to second-order accuracy. For arbitrary higher order accuracy, solution reconstruction should be consistent with the finite volume representation of the solution in Equation 3.2, that is, the average of the reconstruction over the cell should reproduce the finite volume solution.

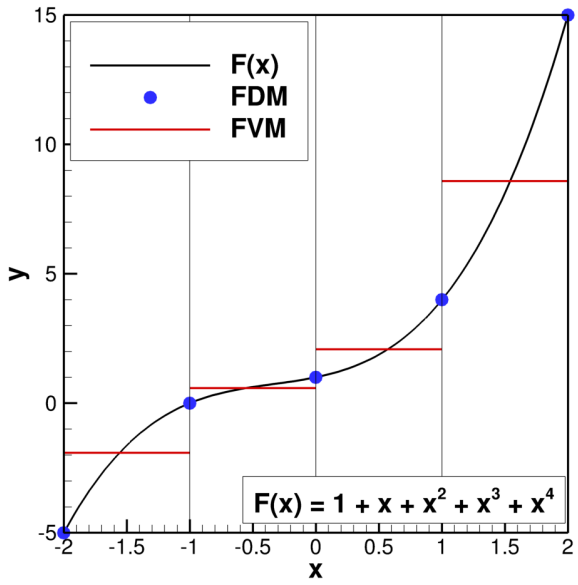


Figure 3.1: Finite volume and finite difference solution representations

Significant work has been done in the area of solution reconstruction for flux calculations on structured and unstructured grids. Most reconstruction procedures have a similar form that seek to represent the discrete solution as a polynomial over a given cell which can then be integrated using a quadrature to compute a higher-order flux at the cell face. One of the more basic reconstruction methods which was developed by Barth[2,3] is the k-exact method where k represents the order of the polynomial used in the reconstruction. The k-exact method uses the smallest stencil possible to reconstruct the solution and has been successfully applied to the Euler and Navier-Stokes equations.[2,3,4] An essentially non-oscillatory (ENO) method developed by Harten and Osher[5] uses the same principles as the k-exact method except the reconstruction stencil is adapted so that the solution is non-oscillatory. The adaptation of the reconstruction stencil from iteration to iteration causes convergence issues with steady-state problems and several modifications have been developed to improve convergence and improve computational efficiency (see for example Shu[6] and Godfrey et al.[7]). The weighted ENO (WENO) method was developed by Liu et al.[8] to address convergence issues of the ENO method. Instead of selecting the best stencil for the solution reconstruction, all possible stencils are chosen with specific weights applied to each. The WENO method has been successfully applied to up to seventh-order polynomial reconstructions.[9] Hybrid methods have also been developed, for example, Li and Ren[10] developed a hybrid method that combines the k-exact and WENO methods.

The development of the k-exact and ENO schemes are driven by the need for stable, computationally inexpensive, higher-order accurate solutions. Solution reconstruction for the purposes of truncation error estimation is driven by a slightly different set of requirements. Truncation error is the result of truncated terms from the infinite series (commonly Taylor's series) used to ap-

proximate differential or integral governing equations and is a function of solution derivatives and grid spacing. Several different methods of estimating truncation error are available. The focus of this paper is the development of a single method for estimating truncation error from an accurate, continuous representation of the numerical solution. The requirements for good numerical convergence properties is no longer necessary, and since truncation error is computed only once, computational efficiency is significantly less important. Instead, an emphasis is placed on the accuracy of the truncation error estimate. The non-oscillatory properties of the ENO scheme are useful for solution reconstruction near singularities and one-sided stencils which are ultimately required near the edge of the computational domain.

3.2 TE Framework

A set of operators are used to represent the governing equations and prolongation/restriction operations. Prolongation is an operation that transforms a coarser-spaced solution to a finer-spaced solution (e.g. solution reconstruction), and restriction is an operation that transforms a finer-spaced solution to a coarser-spaced solution. The numerical solution to a given set of PDEs can be represented by the General Truncation Error Expression (GTEE)[11,12]

$$L_h(\cdot) = L(\cdot) + \tau_h(\cdot) \tag{3.5}$$

where $L_h(\cdot)$ represents the discretization operator, $L(\cdot)$ represents the continuous governing equations, and $\tau_h(\cdot)$ represents the truncation error. Note that different truncation error equations can be derived by evaluating the GTEE on either continuous or discretized solutions and apply the needed prolongation or restriction operators. For example, if f is a continuous function then

$$L_h(I^h f) = I^h L(f) + \tau_h(f) \tag{3.6}$$

where I^h is an interpolation operator and is used to move from a continuous space to a discrete space with mesh size h. In general, we consider interpolation operators I_a^b where a is the starting space and b is the target space. The discrete space h represents a solution on a mesh with spacing h and a blank sub/superscript represents a continuous (infinite dimensional) space with infinite degrees of freedom. The restriction of a continuous space to a discrete space is relatively straightforward. For a finite difference method the restriction could simply be considered as the evaluation of the continuous function at the i-th x location

$$f_{i,h} = I^h f = f(x_i). \tag{3.7}$$

For a finite volume method the natural restriction to a discrete space is the average value of the function over the i-th cell with volume V_i

$$f_{i,h} = I^h f = \frac{1}{V_i} \int_{V_i} f(x) dV. \tag{3.8}$$

If we consider a discretized function, then the GTEE can be written as

$$L_h(f_h) = I^h L(I_h f_h) + \tau_h(I_h f_h). \tag{3.9}$$

The prolongation from discrete space to continuous space is an approximation that is better written as

$$I_h = I_h^q + O(h^{q+1}) \tag{3.10}$$

to account for the error in the prolongation, where q is the order of the reconstructed polynomial for which the k-exact and ENO schemes are $q+1$ order accurate. Equation 3.9 is then written as

$$L_h(f_h) = I^h L(I_h^q f_h + O(h^{q+1})) + \tau_h(I_h^q f_h + O(h^{q+1}))$$

or

$$L_h(f_h) = I^h L(I_h^q f_h) + \tau_h(I_h^q f_h) + O(h^{\bar{q}}) \tag{3.11}$$

where $\bar{q}$ is an order of accuracy which is related to q may be modified by nonlinearities in the $L(.)$ and $\tau_h(.)$ operators. Due to the error in the prolongation, it cannot be generally assumed that $f_h = I^h I_h f_h$.

As is evident from the GTEE, the truncation error is the difference between the discrete and continuous governing equations and has been shown to be the local source of discretization error in a numerical solution[12] where the discrete form of the discretization error is defined as

$$\varepsilon_h = u_h - I^h \tilde{u}. \tag{3.12}$$

Here, u_h is the exact solution to the discrete equations such that $L_h(u_h) = 0$, and $\tilde{u}$ is the exact solution to the original governing equations such that $L(\tilde{u}) = 0$. The accurate estimation of truncation error is a key aspect to several residual-based discretization error estimation methods such as the error transport equations,[13,14] defect correction,[15,16,17] and adjoint methods.[18] Also, as the local source of discretization error, truncation error has been shown to be an effective mesh adaption indicator,[12] possibly with adjoint weighting.[19]

The focus of this paper is a truncation error estimation method which requires a prolongation of the numerical solution to a continuous space. Inserting the exact solution to the discrete equations u_h into Equation 3.5, and noting that $L_h(u_h) = 0$, results in

$$\tau_h(I_h^q u_h) = -I^h L(I_h^q u_h) + O(h^{\bar{q}}). \tag{3.13}$$

The accuracy of the truncation error estimate is determined by comparing the estimated truncation error to the exact truncation error which requires the exact solution to the governing equations. It can be found by inserting the exact solution into the GTEE and, noting that $L(\tilde{u}) = 0$, gives

$$\tau_h(\tilde{u}) = L_h(I^h \tilde{u}) \tag{3.14}$$

where the exact and approximate truncation errors are related by

$$\tau_h(\tilde{u}) = \tau_h(I_h u_h) \approx \tau_h(I_h^q u_h). \tag{3.15}$$

Various method of estimating truncation error have been developed and tend to be specific to the application. For example, adjoint methods typically use an embedded grid approach developed by Venditti and Darmofal.[19] This method inserts a coarse grid solution into a finer grid using a reconstruction method

$$\tau_h(I_h u_h) = -I_{h/r}^h L_{h/r}\left(I_q^{h/r} I_h^q u_h\right) + O(h^{\bar{q}}) \tag{3.16}$$

where h represents the coarse grid and h/r represents the fine grid (refined by factor r). This method is modified by Phillips et al.[20] to improve accuracy since the truncation error on the fine grid solution is still significant for typical refinement factors of two. Equation 3.16 is modified to include the effects the fine grid truncation error assuming that the truncation error is asymptotic so that $\tau_h = \frac{\tau_{h/r}}{r^{p_f}}$ where p_f is the formal order of accuracy of the truncation error

$$\tau_h(I_h u_h) = -I_{h/r}^h L_{h/r}\left(I_q^{h/r} I_h^q u_h\right)\left(\frac{r^{p_f}}{r^{p_f}-1}\right) + O(h^{\bar{q}}). \tag{3.17}$$

A detailed derivation is given in the Appendix. The formal order of truncation error is related to the formal order of accuracy of the discretization error; however, for unstructured grids or grids with randomly perturbed nodes, the formal order of truncation error can be lower than the formal order of the discretization error. Shih and Williams[21] estimated truncation error for the error transport equations by inserting a finer grid solution into the course grid discrete operator

$$\tau_h(I_h u_h) = L_h\left(I_{h/r}^h u_{h/r}\right). \tag{3.18}$$

Fulton[22] used a similar embedded grid approach for finite difference discretization schemes, but adjusted the truncation error estimate to take into account the relative magnitudes of the truncation error between the two grids

$$\tau_h(I_h u_h) = L_h\left(I_{h/r}^h u_{h/r}\right)\left(\frac{r^{p_f}}{r^{p_f}-1}\right). \tag{3.19}$$

Fraysse et al.[23] extended Fulton's work to finite volume methods and included the effects of iterative convergence error for numerical solutions which are not fully converged. Phillips et al.[20] followed a similar approach but reinterpolaed the estimated truncation error back onto the computational grid

$$\tau_h(I_h u_h) = I_q^h I_{rh}^q L_{rh}\left(I_h^{rh} u_h\right)\left(\frac{1}{r^{p_f}-1}\right) + O(h^{q+1}). \tag{3.20}$$

A detailed derivation is given in the Appendix.

3.3 Solution Reconstruction

Solution Integration

The cell average of the reconstructed solution is computed using a Curtis-Clenshaw[24] quadrature which is extended to multiple dimensions using Smolyak's[25] sparse grid construction. We chose Curtis-Clenshaw over the more common Gauss quadrature because the integration ponts are nested (i.e., higher-order reconstructions use all of the lower order quadrature points) thus allowing adaptive reconstruction and inexpensive quadrature error estimates. The Curtis-Clenshaw quadrature exactly integates a q-th order polynomial using $q+1$ function evaluations in one dimension. The quadrature domain $\vec{\xi}$ is a hypercube over the range $0 \leq \vec{\xi} \leq 1$ so for integration of arbitrary cells a 1st-order polynomial is used to map the arbitrary cell to the quadrature domain where the highest polynomial order for each dimension is 1. For example, in two dimensions the polynomial takes the form

$$x_1\left(\vec{\xi}\right) = a_{x_1} + b_{x_1}\xi_1 + c_{x_1}\xi_2 + d_{x_1}\xi_1\xi_2.$$

This polynomial is used to exactly constrain the transformation. Given the Curtis-Clenshaw function evaluation points $\vec{\xi}_{cc}$ and weights $\vec{w}$ the integral of a function over an arbitrarily dimensioned hypercube is

$$\frac{1}{V_i}\int_{V_i} f(\vec{x})\,dV = \frac{1}{V_i}\sum_{j=1}^{N_{cc}} f\left(\vec{x}\left(\vec{\xi}_{j,cc}\right)\right) w_{j,cc}\left|det(J(\vec{\xi}_{j,cc}))\right| \tag{3.21}$$

where J is the Jacobian of the coordinate transformation, and the weights can be found from Burkardt.[26] Because of the non-linear term in the transformation, the Jacobian is not necessarily constant.

The linear system for solution reconstruction is assembled from a stencil of N_{st} cells. To mitigate the effects of round-off error and prevent ill-conditioned systems, the reconstruction is computed on the domain $\xi_{fit} = [0,1]$ and requires an additional linear transformation to go from the fit domain to the Curtis-Clenshaw domain

$$\xi_{fit}^{(n)} = \frac{\xi_{cc}}{N_{st}} + \frac{n-1}{N_{st}} \tag{3.22}$$

where n is the cell number in the reconstruction stencil.

Equation 3.21 is written more appropriately using matrix notation

$$V_i u_{i,h} = \vec{w}_{cc}^{T} P\left(\vec{\xi}_{fit}^{(n)}\left(\xi_{cc}\right)\right)\vec{C} \tag{3.23}$$

where for the i-th cell in the reconstruction stencil, $\vec{w}$ is a column vector of weights returned from the Curtis-Clenshaw quadrature, $P(\vec{x}_i)$ is the coefficient matrix for a polynomial fit of size $\left(N_{cc} \times (k+1)^d\right)$, and $\vec{C}$ is a column vector of the unknown polynomial coefficients.

The complete linear system is written as

$$\vec{W}C = \vec{V}^T \vec{u}_h \tag{3.24}$$

where $\vec{V}$ is a column vector of cell volumes and for the n-th row $\vec{W}_n = \vec{w}_{cc}^T P\left(\vec{x}_n\left(\vec{\xi}_{cc}\right)\right)$. If the stencil does not change, then Equation 3.24 can be solved prior to computing the reconstruction for computational efficiency. For all solution reconstructions in this work, polynomial orders from one to four are considered.

K-Exact Method

The k-exact method developed by Barth[2,3] was designed to conserve the mean value of the cell, reconstruct polynomials of degree k or less exactly, be compact, and be computationally efficient. For a polynomial of order k there are $k+1$ unknowns and $N_{st} = k+1$. For higher dimensions, the polynomial is a tensor product of one-dimensional polynomials with $(k+1)^d$ unknowns where d is the dimension and $N_{st} = (k+1)^d$. A centered stencil is used for the interior of the domain and a shifted stencil near the computational boundary. The reconstruction guarantees that the average value over each cell used in the reconstruction reproduces the numerical solution

$$u_h = I_k^h I_h^k u_h \tag{3.25}$$

The k-exact is solved in a least squares method by increasing the size of the stencil so that Equation 3.25 is valid only for the cell of interest. The method implemented here results in the smallest possible stencil and is equally sized in each coordinate dimension. The conservation of the mean is important in the context of truncation error estimation so that

$$L_h(I_k^h I_h^k u_h) = 0. \tag{3.26}$$

ENO Method

The ENO method developed by Harten[5] was extended to arbitrarily dimensions by Godfrey et al.[7] The reconstruction-via-primitive ENO (RP-ENO) can achieve arbitrary high order accuracy, but is computationally expensive. Godfrey et al.[7] also introduced a less computationally expensive dimensionally split ENO (DS-ENO) method but does not include cross-derivative terms.

The basis of the ENO method is the selection of the stencil to reduce the variation of the solution used to compute the reconstruction. Starting at cell i, the difference between the solutions to the left $(i-1)$ and right $(i+1)$ are compared and the minimum is added to the reconstruction stencil. The divided difference is repeated until the stencil has N_{st} points.

RP-ENO Method

The RP-ENO scheme computes the left and right states of a cell using progressive one dimensional curve-fits with the adaptive stencils. For a two-dimensional reconstruction the line-averaged solution in the η-direction is first computed from a one dimensional solution reconstruction where the stencil extends in the ξ direction

$$p_i(\xi) = \frac{1}{\Delta\eta}\int_{j-1/2}^{j+1/2} u_{i,h}(\xi,\eta)d\eta. \tag{3.27}$$

The reconstructed solution is evaluated at the cell boundaries $p([\xi^+_{i-1/2},\xi^-_{i+1/2}])$ for all cells in the domain. Next, the solution is reconstructed at the left and right faces using a one dimensional reconstruction of the left and right face line averaged values with a stencil that extends in the η-direction. The reconstructed solution is then used to compute the flux at the left and right cell faces. The process is repeated for the top and bottom faces of the cell by computing the line averaged solution in the ξ-direction first. A similar process is followed for three dimensional reconstructions.

DS-ENO Method

The dimensionally split ENO scheme follows the same procedure except only one reconstruction is computed for each face to compute the line-averaged solution in the η-direction and the left and right faces and the line-averaged solution in the ξ-direction at the top and bottom faces. The average solution is used to compute the flux instead of the reconstructed local solution which does not capture the cross-derivative terms. The computational cost of the DS-ENO method is an order of magnitude cheaper than the RP-ENO and k-exact reconstruction methods; however, in our case, computational expense is less of a concern since the truncation error is generally a one time estimate.

3.4 Governing Equations

Burgers' Equation

The scope of applications investigated in this paper are related to fluid dynamic problems, more specifically, solutions to the Euler and Navier-Stokes equations. Burgers' equation is a useful scalar example problem to illustrate and initially test solution reconstruction methods. Burgers' equation is chosen because it is analogous to the Navier-Stokes equations including a non-linear convective term and a linear diffusive term. The conservative form of Burgers' equation is

$$\frac{\partial u}{\partial t} + \int_S \left(\frac{u^2}{2} - \nu\frac{\partial u}{\partial x}\right) dS = 0. \tag{3.28}$$

An exact solution for a viscous shock is

$$\tilde{u}(x) = -u_{ref} \tanh\left(x\frac{Re}{2L}\right) \tag{3.29}$$

where Re is the Reynolds number

$$Re = \frac{u_{ref}L}{\nu}$$

L is a reference length (here $L = 8$), and u_{ref} is chosen as two. The exact solution with $Re = 16$ is shown in Figure 3.2. This equation is solved using an explicit, cell-centered finite volume scheme where

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{F_{i+1/2}^n - F_{i-1/2}^n}{\Delta x} = 0 \tag{3.30}$$

and

$$F_{i+1/2} = \frac{u_{i+1}^2 + u_i^2}{4} - \nu\frac{u_{i+1} - u_i}{(\Delta x_{i+1} + \Delta x_i)/2}. \tag{3.31}$$

Green's theorem is used to compute the gradient $\partial u/\partial x$ in Equation 3.31 which is equivalent to a finite difference on a Cartesian grid.

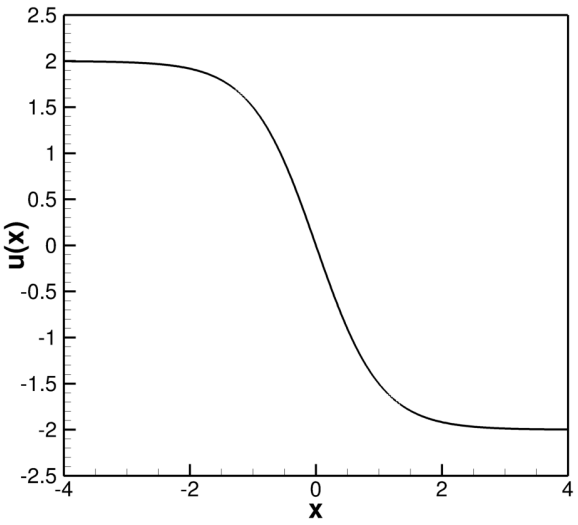


Figure 3.2: Burgers' Equation exact solution for $Re = 16$ and 513 grid nodes

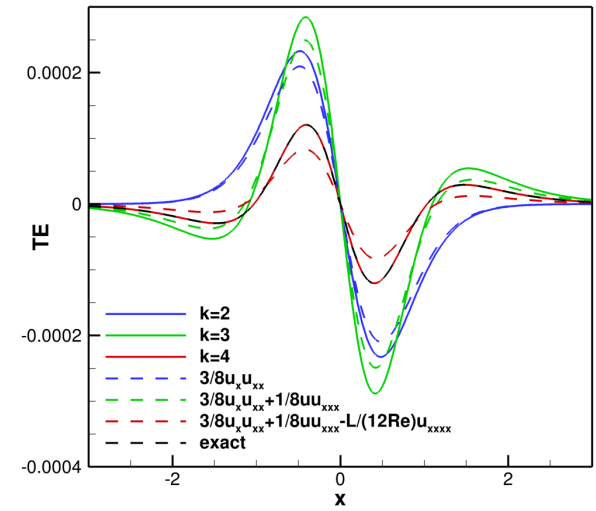


Figure 3.3: Burgers' Equation truncation error estimate for $Re = 16$ and 513 grid nodes

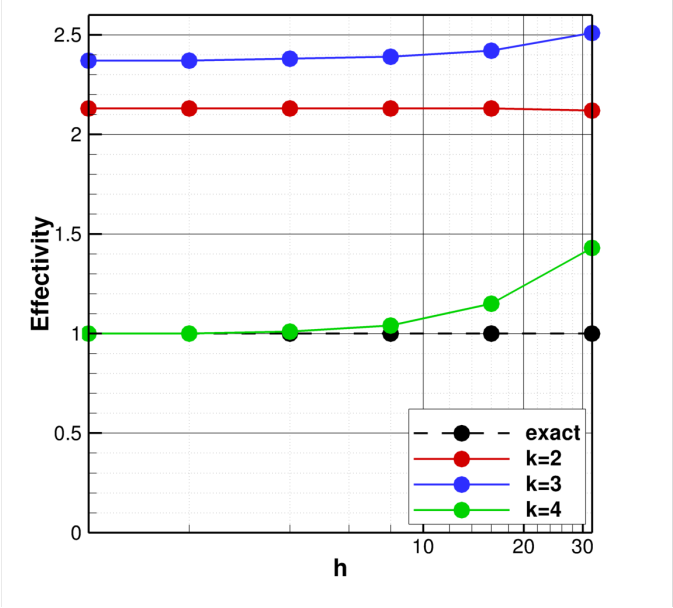


Figure 3.4: Burgers' equation normalized maximum truncation error estimate vs. grid refinement for $Re = 16$

The full truncation error expression consists of an infinite series and can only be exactly calculated using Equation 3.14 which requires the exact solution; however, for sufficiently fine grids, only the leading truncation error terms need to be accurately estimated. In general, the leading terms are not derived because of the complexity of discretization schemes, but Burgers' equation is simple enough that this is relatively easy. On a grid with equal spacing the leading truncation error terms

are

$$\tau_h(u) = \Delta x^2 \left(\frac{3}{8} u_x u_{xx} + \frac{1}{8} u u_{xxx} - \frac{1}{24} \nu u_{xxxx} \right) + O(\Delta x^4). \tag{3.32}$$

From Equation 3.32 it is clear that this is a second-order accurate scheme and that the next highest truncation error terms are fourth-order accurate. For an accurate estimate of the leading terms the error between the estimated truncation error and the exact truncation should decrease at at least a fourth-order rate.

The discrete solution with $Re = 16$ computed using 513 grid nodes is reconstructed using the k-exact reconstruction method with $k = 2, 3$, and 4. The minimum value for k is two because the highest derivative in the governing equations is two and any value of k less than this would result in a derivative term reducing to zero. The truncation error is estimated from the solution reconstructions using Equation 3.13 and the results are shown in Figure 3.3. The estimated truncation error is compared to the exact truncation error computed using Equation 3.14. The estimated truncation error for $k = 4$ is indistiguishable from the exact truncation error and the error between the exact and estimated truncation error decreases at a fourth-order rate. The other two reconstruction methods, $k = 2$ and 3, do not qualitatively match well, and the error between the exact and estimated truncation error decreases at only a second-order rate. Figure 3.4 compares the maximum truncation error estimate normalized by the maximum exact truncation error for a series of grids. The abscissa plots the grid spacing normalized by the finest grid spacing for 513 nodes. With grid refinement, it is clear that the estimated truncation error for $k = 2$ and 3 will never accurately represent the exact truncation error since they asymptote to a constant error with mesh refinement.

If Equation 3.32 is substituted into Equation 3.15, then

$$\Delta x^2 \left(\frac{3}{8} \tilde{u}_x \tilde{u}_{xx} + \frac{1}{8} \tilde{u} \tilde{u}_{xxx} - \frac{1}{24} \nu \tilde{u}_{xxxx} \right) -$$

$$\Delta x^2 \left(\frac{3}{8} (I_h^k u_h)_x (I_h^k u_h)_{xx} + \frac{1}{8} (I_h^k u_h)(I_h^k u_h)_{xxx} - \frac{1}{24} \nu (I_h^k u_h)_{xxxx} \right)$$

$$= 0 + O(\Delta x^{4,\bar{q}}).$$

and as long as k is sufficiently large so that the derivatives of the reconstruction do not go to zero because of an insufficient order polynomial then the error in the truncation error estimate should be fourth-order accurate. If $k = 3$, the fourth derivative of the reconstruction is zero which results in

$$\Delta x^2 \left(\frac{3}{8} \tilde{u}_x \tilde{u}_{xx} + \frac{1}{8} \tilde{u} \tilde{u}_{xxx} \right) - \Delta x^2 \left(\frac{3}{8} (I_h^k u_h)_x (I_h^k u_h)_{xx} + \frac{1}{8} (I_h^k u_h)(I_h^k u_h)_{xxx} \right)$$

$$= \frac{\Delta x^2}{24} \nu \tilde{u}_{xxxx} + O(\Delta x^{4,\bar{q}}).$$

This means that a polynomial of insufficient order cannot capture all of the leading terms in the truncation error resulting in a persistent error term that decreases at the same rate as the truncation error. Figure 3.3 compares the truncation error estimates computed using Equation 3.13 with $k = 2$

to the first truncation error term in Equation 3.32, with $k = 3$ to the first and second truncation error terms, and with $k = 4$ to all three of the truncation error terms. The selected combination of terms from Equation 3.32 and estimated truncation errors compare well. The implications of this observation would mean a minimum polynomial order k is required to accurately estimate truncation error specific to each discretization scheme, and is equal to the highest derivative found in the analytic truncation error.

Euler and Navier-Stokes Equations

The CFD code SENSEI,[27] a structured, cell-centered, finite volume solver, is a research code used to implement the reconstruction methods and truncation error estimation methods. SENSEI currently solves the Euler and laminar Navier-Stokes equations

$$L_h(\vec{U}) = \frac{\partial}{\partial t}\int_V \vec{U}dV + \oint_A \left(\vec{F}_i - \vec{F}_v\right)d\vec{A} = \int_V \vec{S}dV \tag{3.33}$$

where the inviscid flux is

$$\vec{F}_i = \begin{bmatrix} \rho\vec{V}\cdot\hat{n} \\ \rho u\vec{V}\cdot\hat{n} + n_x p \\ \rho v\vec{V}\cdot\hat{n} + n_y p \\ \rho w\vec{V}\cdot\hat{n} + n_z p \\ \rho h_t\vec{V}\cdot\hat{n} \end{bmatrix} \tag{3.34}$$

and $\vec{V} = [u, v, w]$ is the velocity vector, $\hat{n} = [n_x, n_y, n_z]$ is the face normal vector, and h_t is the total enthalpy computed using the calorically perfect gas assumption $h_t = \frac{p\gamma}{\rho(\gamma-1)} + \frac{|\vec{V}|^2}{2}$. The viscous flux is

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\theta_x + n_y\theta_y + n_z\theta_z \end{bmatrix} \tag{3.35}$$

where

$$\tau_{xx} = 2\mu\left(\frac{\partial u}{\partial x} - \frac{1}{3}\nabla\cdot\vec{V}\right),\ \tau_{yy} = 2\mu\left(\frac{\partial v}{\partial y} - \frac{1}{3}\nabla\cdot\vec{V}\right),$$

$$\tau_{zz} = 2\mu\left(\frac{\partial w}{\partial z} - \frac{1}{3}\nabla\cdot\vec{V}\right), \tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right),$$

$$\tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right),\ \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right),$$

$$\theta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x},$$

$$\theta_y = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y},$$

$$\theta_z = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z}.$$

The fluid is assumed to be a calorically perfect gas where the equation of state is $P = \rho RT$. The Navier-Stokes equations are reduced to the Euler equations by setting $\vec{F}_v = 0$ For the finite volume method the continuous operator is

$$L(\vec{U}) = \frac{\partial}{\partial t}\int_V \vec{U}dV + \oint_A \left(\vec{F}_i - \vec{F}_v\right)\cdot d\vec{A} - \int_V \vec{S}dV = 0 \tag{3.36}$$

The discrete operator is

$$L_h\left(\vec{U}_h\right) = \frac{\partial \vec{U}_i}{\partial t} + \frac{1}{V_i}\left[\sum_{k=1}^{k_{max}} \vec{F}_k\left(\vec{U}_L, \vec{U}_R\right) + \vec{F}_{v,h}\left(\vec{V}\right)\right]\cdot \hat{n}_k A_k - S_i = 0 \tag{3.37}$$

where $\vec{U}_L$ and $\vec{U}_R$ are the left and right approximations at face k, $\vec{F}_k$ is a Riemann solver, and $F_{v,h}$ is the viscous flux and Green's theorem is used to compute the derivatives centered about the cell face.[28] The Riemanns solver used for all simulations is the flux vector splitting method developed by van Leer.[29]

3.5 Applications

Manufactured solutions are a common method used for code verification purposes and allow for the arbitrary specification of an exact solution to almost any governing equation.[30,31] Different manufactured solutions are used to test the different reconstruction methods for truncation error estimation. The manufactured solutions are chosen to empirically identify various possible truncation error terms that result from the solution and grid choice and identify the minimum order reconstruction required to accurately estimate truncation error for the Euler and Navier-Stokes equations. Furthermore, the manufactured solutions are meant to highlight the differences between each of the reconstruction methods and identify possible weaknesses and advantages of each method. The manufactured solution source term is calculated by operating the governing equations on a general manufactured solution f

$$I^h L(f) = S_h. \tag{3.38}$$

The exact truncation error is then computed using

$$L_h(I^h f) - S_h = \tau_h(f). \tag{3.39}$$

The general form of the two-dimensional manufactured solution is

Table 3.1: Coefficients for manufactured solution

	a_0	a_x	b_x	c_x	a_y	b_y	c_y	a_z	b_z	c_z
ρ	1.0	0.15	2.0	0.33	-0.1	1.0	-0.2	0.2	0.5	0.1667
u	800.0	50.0	0.66	0.5	-30.0	1.5	-0.125	40.0	1.66	0.33
v	800.0	-75.0	1.66	-0.5	4.0	3.0	0.0	60.0	4.0	0.0
w	900.0	-30.0	0.5	0.0	80.0	1.5	-0.33	45.0	0.75	-0.25
p	1.e5	-2.e4	0.5	0.5	5.e4	1.0	-0.5	3.e4	0.33	0.0
		a_{xy}	b_{xy}	c_{xy}	a_{yz}	b_{yz}	c_{yz}	a_{xz}	b_{xz}	c_{xz}
ρ		-0.05	1.0	0.25	0.25	0.5	-0.33	0.15	2.0	0.125
u		-60.0	1.5	0.125	25.0	0.625	-0.25	50.0	0.5	0.25
v		60.0	1.5	0.125	-25.0	0.625	-0.25	-60.0	0.37	5 0.25
w		30.0	1.0	0.25	-50.0	0.5	-0.33	40.0	1.5	0.33
p		-1.e4	0.75	-0.25	2.e4	0.5	-0.25	-3.e4	1.5	0.125

$$f(x,y) = a_0 + a_x \sin\left(\frac{b_x \pi x}{L} + c_x \pi\right) + a_y \sin\left(\frac{b_y \pi y}{L} + c_y \pi\right) + a_{xy} \sin\left(\frac{b_{xy} \pi xy}{L^2} + c_{xy} \pi\right). \quad (3.40)$$

The coefficients for the supersonic manufactured solution are shown in Table 3.1 and the two-dimensional manufactured solution is shown in Figure 3.5. A subsonic manufactured solution is created by dividing the velocity coefficients by ten (e.g. a_0, a_x, a_y, etc.). The dynamic viscosity for the Navier-Stokes equations is set to a constant value of $1\,Pa \cdot s$. The viscosity was set so that the convective and diffusive source terms were on the same order of magnitude.

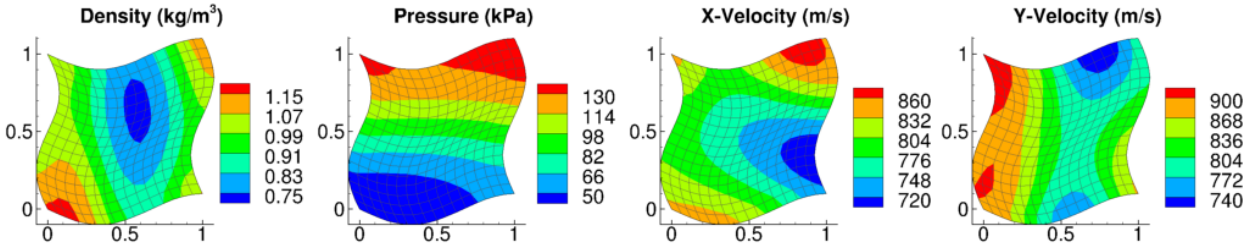


Figure 3.5: Curvilinear manufactured solution

3.6 Results

The same process is applied for the Euler and Navier-Stokes equations as was done for Burgers' equation. The initial goal of this research is to determine the minimum order polynomial reconstruction required to accurately estimate the truncation error for the Euler and Navier-Stokes equations. Solutions and truncation error estimates are computed using three different grid families with the coarsest grid being 17×17 nodes and the finest grid being 257×257 nodes. An example

of each is shown in Figure 3.6. The Cartesian grid is the simplest and is expected to be the easiest to estimate truncation error. The complexity is increased on the skewed grid; however, the linear grid transformation can exactly represent the grid distribution. The final grid is a curvilinear grid with a sinusoidal distribution which cannot be exactly represented by the linear grid transformation.

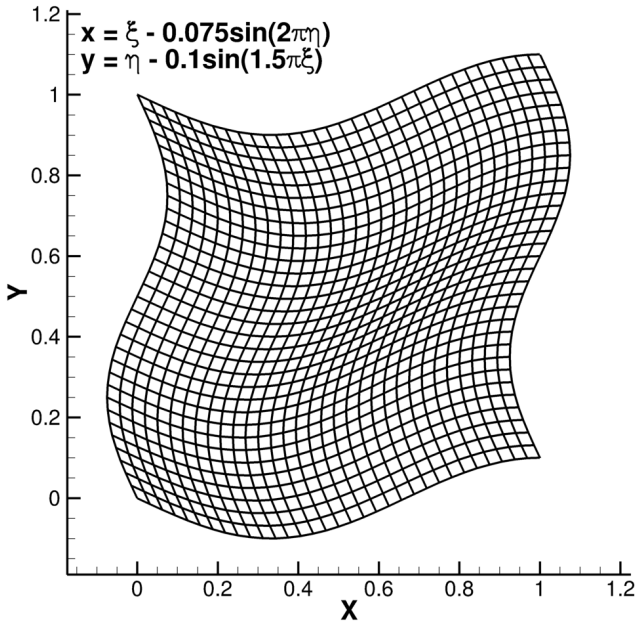


Figure 3.6: Curvilinear grid, 33×33 grid nodes

Truncation error estimates for Euler and Navier-Stokes solutions are computed using the curvilinear grid. The different methods include k-exact, RP-ENO, DS-ENO, the coarse grid method used by Fulton,[22] Equation 3.20, with and without the correction term, and the fine grid method used by Venditti and Darmafal,[19] with and without the correction term, for reconstructions with $k = [1,2,3,4]$. The effectivity index[32] is used to evaluate the truncation error estimates which is the truncation error estimate normalized by the exact truncation error, $\theta_{L_2} = ||\tau_h(I_h u_h)||_{L_2} / ||\tau_h(\tilde{u})||_{L_2}$. Figure 3.7 shows the effectivity index for the energy truncation error for the supersonic Euler solution, and Figure 3.8 shows the effectivity index for the energy truncation error for the subsonic Navier-Stokes solution. The subsonic Euler and Supersonic Navier-Stokes results are very similar to the the subsonic Navier-Stokes and supersonic Euler results, respectively. The energy equation was shown because it is the most complex truncation error term; however, the truncation error for the other equations behave in a similar manner.

For both the Euler and Navier-Stokes equations, the k-exact reconstruction method is generally the best performing method. For $k = 1$, the truncation error estimates decrease at a first-order rate where the exact truncation error decreases at a second-order rate. The lower rate results in the estimated truncation error diverging from the exact truncation error with grid refinement. For

$k = 2$ and greater, the k-exact truncation error estimates are very accurate and outperform the other methods with only a few exceptions. Based on our previous results with Burgers' equation, it was expected that a higher-order reconstruction would be required for the Navier-Stokes equations; however, all results show that $k = 2$ is sufficient for both the Euler and Navier-Stokes equations. A possible explanation for this result is due to the discretization scheme for the diffusion terms. The derivatives are computed using Green's theorem which reduces to a second-order central difference for one dimensional solutions on an evenly spaced grid. The truncated terms for the second-order central difference is $O(\Delta x^2)$ and $O(\Delta x^4)$. The convective terms have higher order terms which are $O(\Delta x^2)$ and $O(\Delta x^3)$. This would mean that the convective truncation error terms will dominate the error in the truncation error estimate. A higher viscosity Navier-Stokes solution was computed to try to increase the dominance of the diffusive terms. The manufactured solution source terms were compared, and the diffusive terms were on the same order of magnitude as the convective terms. Further evidence that $k = 2$ is sufficient is shown by the divergence of the ENO methods for the supersonic solutions shown in Figures 3.9 and 3.10 which are missing derivatives due to the lower dimension reconstruction. If the diffusive terms were insignificant, the ENO methods would be accurate as shown in Figure 3.9. It might be possible that these terms could dominate for highly diffusive areas of a flow requiring a minimum of $k = 3$; however, there is no evidence to suggest that $k = 2$ is not sufficient.

For the Euler equations and $k = 1$, the ENO truncation error estimation methods decrease at a first-order rate similar to the k-exact method and results in an effectivity index that diverges with grid refinement (not shown in the figure). For $k = 2$, the ENO truncation error estimation methods decrease at a third-order rate resulting in a substantial under prediction of the truncation error. The exact reason for this behavior is not yet understood; however, the ENO schemes are used for solution reconstruction where $k = 2$ is third order accurate. The truncation error estimate for $k = 2$ could be an estimate of the truncation error not for the current MUSCL extrapolation upwind method but an approximation of the third-order accurate ENO scheme; however, the truncation error estimates for $k = 3$ and $k = 4$ results in accurate truncation error estimates with only slightly higher error than the k-exact method. For the Navier-Stokes equations, the subsonic solution follows a similar trend as the Euler equations; however, for the supersonic solution the ENO methods do not accurately estimate the truncation error. The estimate is zeroth order accurate because the ENO methods cannot represent the cross-derivative terms. The difference between the subsonic and supersonic Navier-Stokes solution (shown in Figure 3.9) is thought to be due to the relative magnitude of the convective and diffusive terms. The supersonic solution has much larger convective terms due to the larger magnitude velocity and velocity gradient and the zeroth order terms are more apparent. The ENO schemes can be used for truncation error estimation; however, accuracy would suffer in diffusive dominated flow regimes such as boundary layers or shear flows. The effects of a more dominant diffusive term is shown in Figure 3.10 compared to Figure 3.9. (Due to stability issues with the higher viscosity, only three grid levels converged iteratively.) The results are similar, except the truncation error estimated from the solution computed with a viscosity of $50\,Pa \cdot s$ diverges more quickly than the truncation error estimated from the solution computed with a viscosity of $1\,Pa \cdot s$.

For the coarse grid truncation error estimation methods, solution reconstruction is used to interpolate the coarse grid truncation error estimate back to the computational mesh. The uncorrected coarse grid truncation error estimation method is off by a factor of $1/(r^{p_f}-1)$; however, the corrected method accurately estimates the truncation error. The most accurate reconstruction scheme for the corrected method uses a k-exact reconstruction method with $k=1$ for the Euler equations. For the Navier-Stokes equations, $k=1$ underestimates the error for the subsonic solution and the more diffusive supersonic solution; therefore, for the Navier-Stokes equations $k=2$ is recommended.

For the fine grid truncation error estimation methods, solution reconstruction is used to interpolate the computational solution onto a finer mesh that is refined by a factor of two in all coordinate directions. The uncorrected fine grid truncation error estimation method underestimates the truncation error. Corrected by a factor of $r^{p_f}/(r^{p_f}-1)$ results in an accurate truncation error for $k=2$ or higher for both the Euler equations and the Navier-Stokes equations. The fine grid method with the correction is one of the most accurate truncation error estimators evaluated. The method is more accurate for $k=2$ than the ENO and coarse grid methods and indistiguishable from the k-exact method for all solutions except the highly diffusive Navier-Stokes solutions in which the k-exact truncation error estimate is more accurate. It is important to note that the test problems are very smooth. The correction term assumes that the truncation error decreases at the formal order of accuracy which may not be the case for highly non-asymptotic solutions, solutions with poor grid quality, or solutions with singularities/discontinuities.

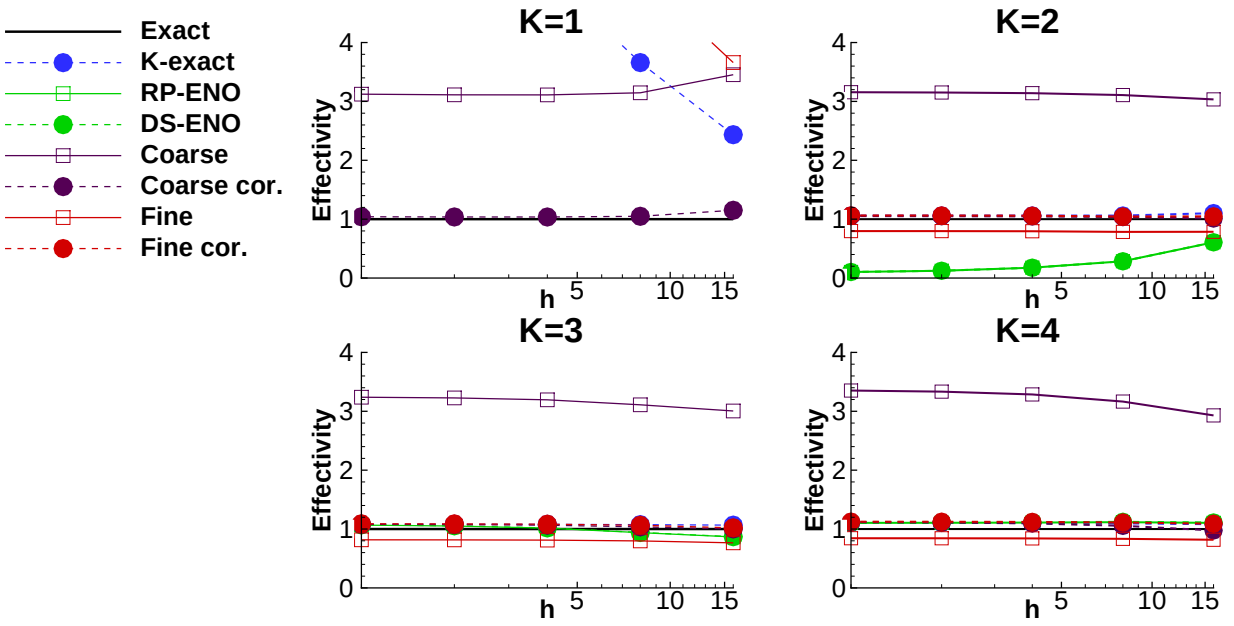


Figure 3.7: Euler energy truncation error estimation comparisons on the curvilinear mesh for the supersonic solution with a viscosity of $\mu = 1\,Pa\cdot s$

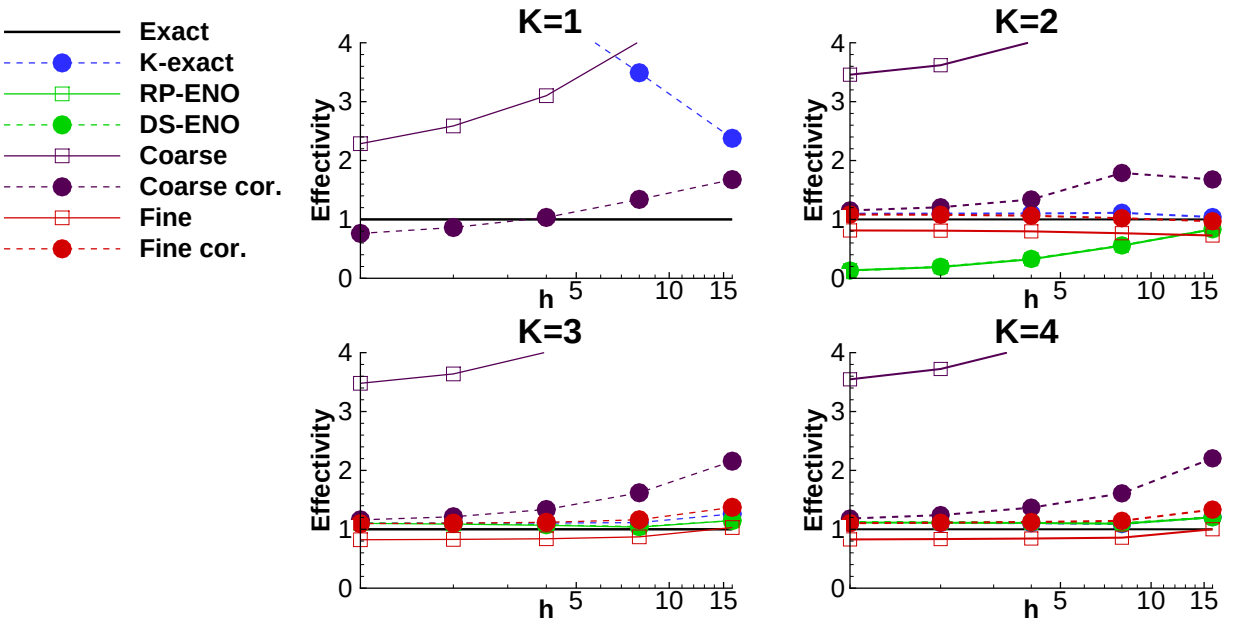


Figure 3.8: Navier-Stokes energy truncation error estimation comparisons on the curvilinear mesh for the subsonic solution with a viscosity of $\mu = 1\,Pa \cdot s$

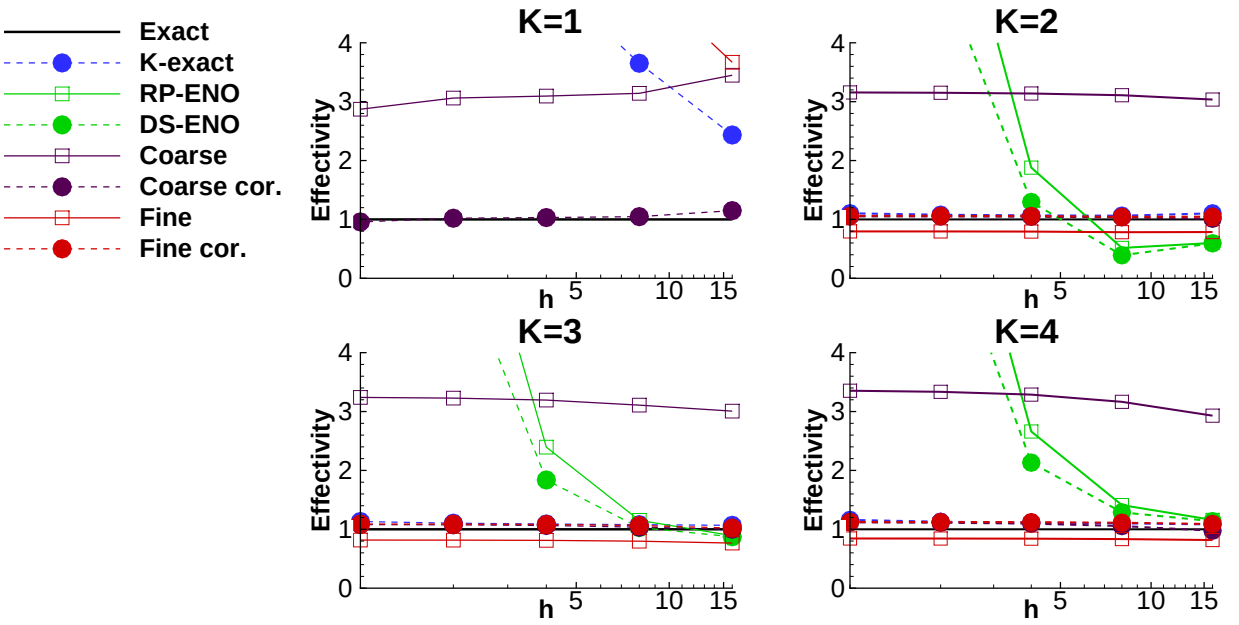


Figure 3.9: Navier-Stokes energy truncation error estimation comparisons on the curvilinear mesh for the subsonic solution with a viscosity of $\mu = 1\,Pa \cdot s$

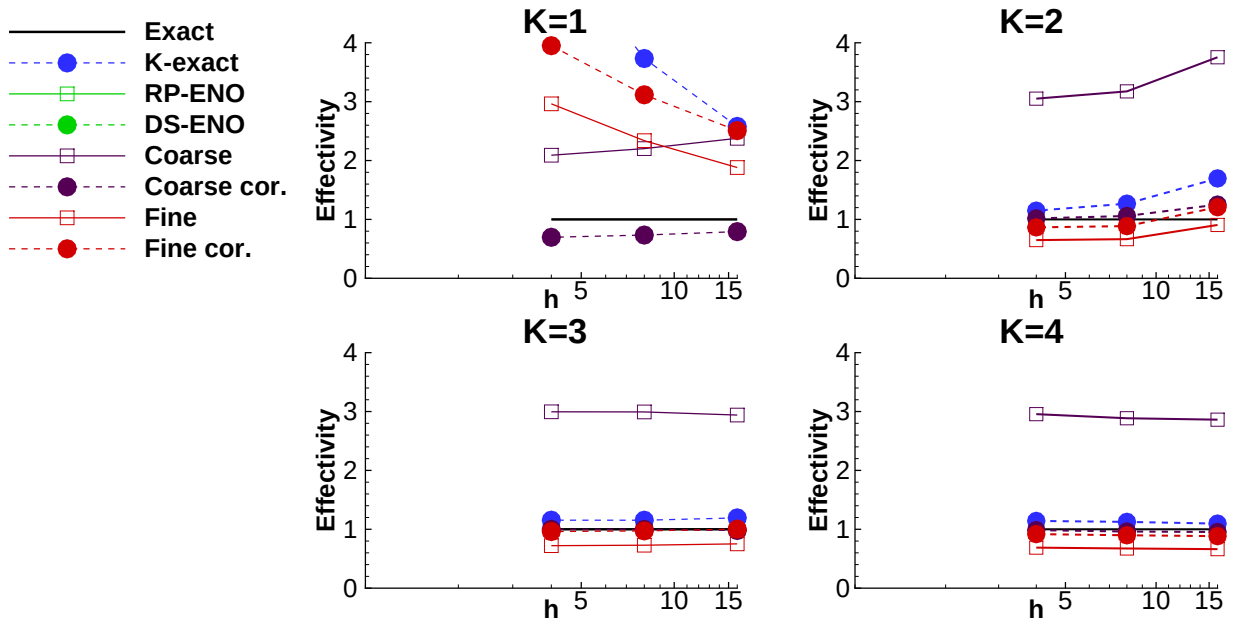


Figure 3.10: Navier-Stokes energy truncation error estimation comparisons on the curvilinear mesh for the supersonic solution with a viscosity of $\mu = 50 Pa \cdot s$

Contour plots of the Navier-Stokes energy truncation error estimates for the supersonic solution on the 33×33 grid are shown in Figure 3.11. The reconstruction order for each method was chosen from the best results from the previous results shown. All contour plots have the same contour levels. All contour plots compare qualitatively well to the exact truncation error. The k-exact and fine grid method with the correction term compare very well to the exact truncation error. The coarse grid method with the correction factor also compares well but has a few peaks that are not present in the exact truncation error. The ENO methods qualitatively match well, but the truncation error estimates are not as smooth. The truncation error for the ENO methods on the 33x33 grid is the most accurate estimate for the series of grid levels (see Figure 3.9, $k = 3$). On the 65x65 grid the truncation error estimate begins to diverge.

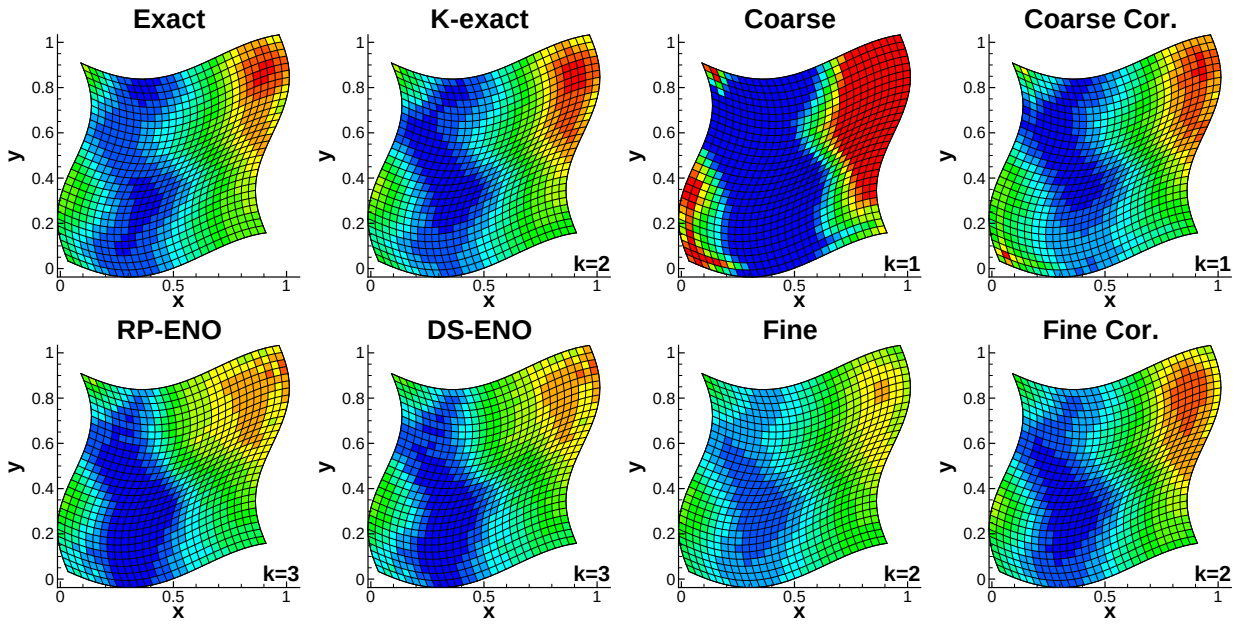


Figure 3.11: Truncation error estimate contours for Navier-Stokes energy equation on a 33×33 curvilinear grid

In the presence of shocks, the order of accuracy of the numerical scheme reduces to first-order accurate. The same results were computed for first-order Euler and Navier-Stokes solutions to determine the minimum required reconstruction order for the truncation error estimation methods. The results are shown in Figures 3.12. The results show that the reconstruction order is one less than what is required for second-order accurate solutions for both the Euler and Navier-Stokes equations. All methods are accurate for $k = 1$ except the ENO schemes which require $k = 2$. The coarse grid correction term reduces to one (i.e., it is not needed) for first-order solutions and the fine grid correction term is two as expected.

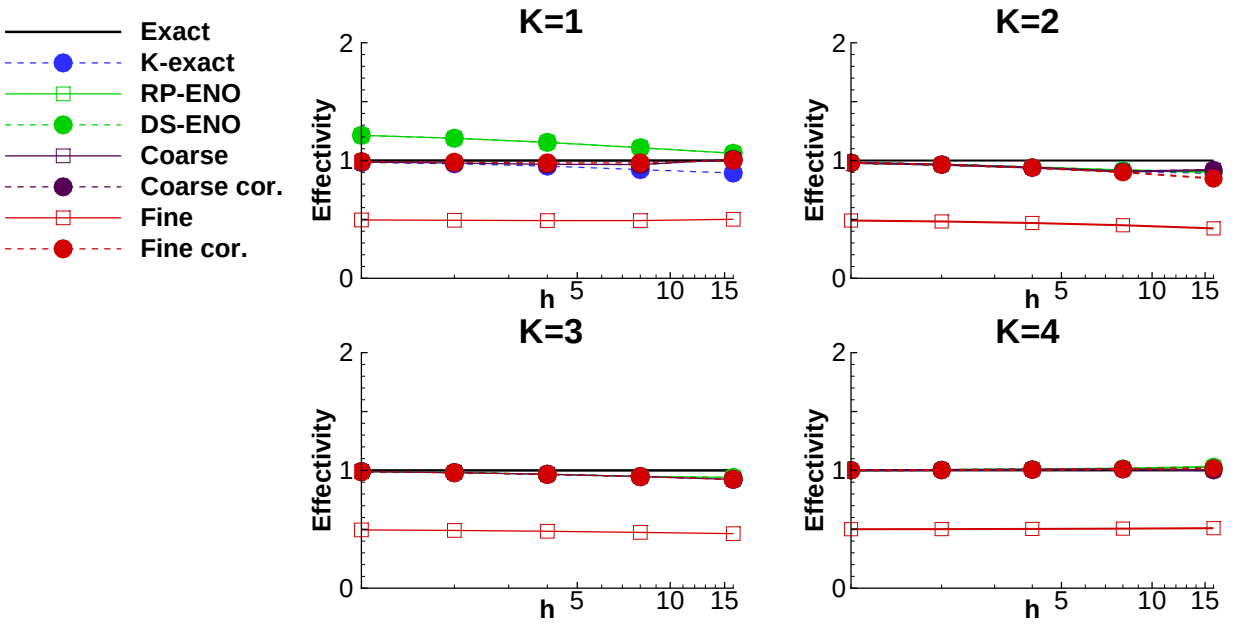


Figure 3.12: Navier-Stokes energy truncation error estimation comparisons on the curvilinear mesh for the supersonic first-order solution

The results for all the truncation error estimation methods are summarized in Table 3.2. The mass and momentum equations showed similar results. The two best performing methods are the k-exact and the fine grid method with the correction term. The k-exact method does not require the assumption of an order of accuracy which is a significant advantage, especially for more practical applications where the formal order of accuracy may not be achieved. For the first-order accurate methods, the ENO methods minimum required reconstruction order for the Navier-Stokes equations showed $k = 2$ is sufficient. This is because the diffusion terms (evaluated using Green's theorem) are still second-order accurate while the convective terms are first-order accurate. The convective error terms dominate the truncation error and results in truncation error estimates that are nearly identical to the Euler equations. This result suggest that the ENO methods could be used in the vicinity of flow singularies such as shock or contact discontinuities, where the order of accuracy reduces to first order or below.[33]

There are several uses for truncation error which include discretization error estimation and higher order solution correction. The possible discretization error estimation methods include the error transport equations,[13,14] defect correction,[15,16,17] and adjoint methods[18] for functionals. Defect correction is the least code intrusive to implement as the estimated truncation error is added as a source term, $L_h(\bar{u}) = \tau_h(I_h^k u_h)$. The solution $\bar{u}$ is an estimate of the exact solution, and the discretization error is estimated by $\bar{\varepsilon}_h = u_h - \bar{u}$. An example of defect correction is shown in Figure 3.13 computed using all truncation error methods for the supersonic Euler solution on the curvilinear grid. A slice is taken through the center of the domain at the $j = 16$ cell index. The k-exact and ENO truncation error estimation methods are the most accurate with the corrected coarse

Table 3.2: Summary of reconstruction orders required for accurate truncation error estimation

	Second-Order		First-Order	
Method	Euler	N-S	Euler	N-S
k-exact	2	2	1	1
RP-ENO	3	-	2	2
DS-ENO	3	-	2	2
Coarse	-	-	1	1
Coarse Cor.	1	2	1	1
Fine	2	2	1	1
Fine Cor.	2	2	1	1

grid method performing well. The uncorrected coarse grid method is not accurate and overestimates the error significantly. The fine grid methods do not capture the shape of the discretization error very well.

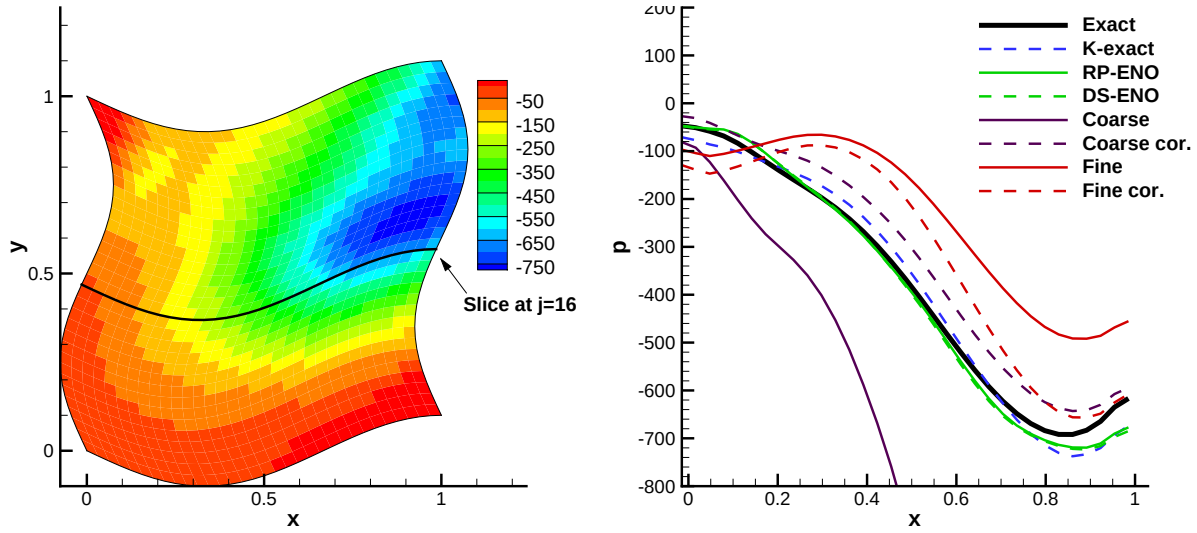


Figure 3.13: Defect correction example using the supersonic Euler solution on the 33x33 grid

The second solution $\bar{u}$ is a higher order accurate solution and can be used as a solution instead of estimating the discretization error. Again, for the supersonic Euler equations, the k-exact method is used to compute higher-order solutions shown in Figure 3.14. The expected order of accuracy is third-order because the k-exact method is estimating the leading truncation error terms which are $O(\Delta x^3)$. The order of accuracy begins very high and appears to be approaching third-order for the finest grid. The higher-order discretization error is not as asymptotic as the second-order solution; however, the discretization error on the finest grid is on the order of $1 \times 10^{-}8\%$ error. The discretization error in the corrected solution is lower for all solution variables except for the pressure which is higher than the second-order solution, but because of the higher order of accuracy

quickly decreases. The solver does not have any higher-order capability, the higher-order results come from the truncation error estimate only.

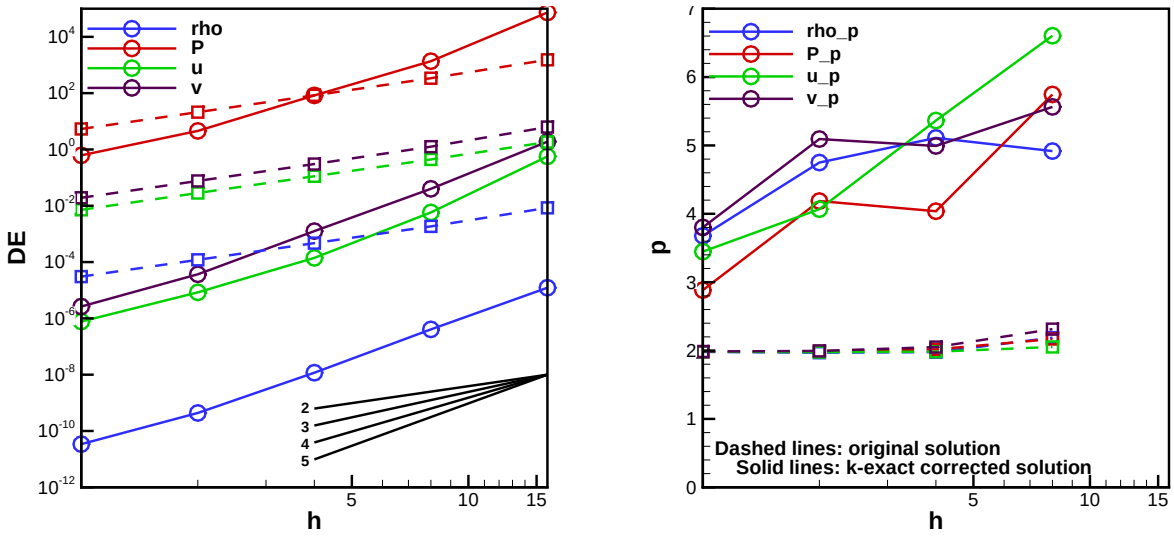


Figure 3.14: Discretization error in the defect correction solution discretization error (left) and the order of accuracy (right) computed using the k-exact truncation error estimation method for the supersonic Euler solution

3.7 Conclusions

Truncation error estimation is a key element for residual-based discretization error estimation and solution adaption. Different reconstruction methods were investigated for the purpose of truncation error estimation for 1D Burgers' equation and the 2D Euler and Navier-Stokes equations. The primary truncation error estimation method of interest computed the continuous residual by operating the governing equations on the reconstructed finite volume solution. The minimum required polynomial order was determined to be 4th order for Burgers' equation using a central discretization scheme. For the Euler equations, the k-exact method was the most accurate reconstruction method and required at least a minimum polynomial order of two for accurate truncation error estimation. The fine grid corrected method was second best with a m inimum polynomial reconstruction order of two. The ENO methods required a minimum polynomial order of three. It was also shown that a coarse grid and fine grid truncation error estimation method must be corrected using the formal order of accuracy of the truncation error to correct for the relative magnitude of truncation error between the two grids. The reconstruction methods were tested on very smooth solutions with smooth grids to determine the basic characteristics of each of the reconstruction methods under a best case scenario.

Acknowledgement

This work was funded by the Air Force Office of Scientific Research (AFOSR) Computational Mathematics Program managed by Dr. Fariba Fahroo under Grant FA9550-12-1-0173.

Bibliography

[1] C. Hirsch, *Numerical Computation of Internal and External Flows, Volume 2: COmputational Methods for Inviscid and Viscous FLows*. John Wiley and Sons, 1984.

[2] T. J. Barth, "Higher order solution of the euler equations on unstructured grids using quadratic reconstruction," Tech. Rep. AIAA-90-0013, 1990. Initial development of K-exact method.

[3] T. J. Barth, "Recent developments in higher order k-exact reconstruction on unstructured meshes," Tech. Rep. AIAA-93-0668, 1993. Improvements to the K-exact method with work towards unstructured grids and comparison to other competing technologies.

[4] C. R. Mitchell and R. W. Walters, "K-exact reconstruction for the navier stokes equations on arbitrary grids," Tech. Rep. AIAA-93-0536, 1993. Apply k-exact to structured and unstructred grids for Euler and Navier-Stokes equations.

[5] A. Harten and S. Osher, "Uniformly high-order accurate nonoscillatory schemes i," *SIAM Journal of Numerical Analysis*, vol. 24, pp. 279–309, 1987. First development of ENO scheme-1D.

[6] C.-W. Shu, "Numerical experiments on the accuracy of eno and modified eno schemes," *Journal of Scientific Computing*, vol. 5, no. 2, pp. 127–147, 1990. Modification to the ENO scheme-1D.

[7] A. G. Godfrey, C. R. Mitchell, and R. W. Walters, "Practical aspects of spatially high-order accurate methods," *AIAA Journal*, vol. 31, no. 9, pp. 1634–1642, 1993. More efficient ENO scheme compared to k-exact and standard 2D standard ENO scheme – DS-ENO.

[8] X. D. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *Journal of Computational Physics*, vol. 115, 1994. Initial development of WENO scheme.

[9] Y. Shen and G. Zha, "Improved seventh-order weno scheme," Tech. Rep. AIAA-2010-1451, 2010. 7th order accurate WENO scheme.

[10] W. Li and Y.-X. Ren, "High-order k-exact weno finite volume schemes for solving gas dynamic euler equations on unstructured grids," *International Journal for Numerical Methods in Fluids*, vol. 70, pp. 742–763, 2012. Combination of k-exact and WENO for simplicity and stabiltiy.

[11] C. J. Roy, "Review of discretization error estimators in scientific computing," AIAA Paper 2010-126, 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 4-7, 2010, 2010.

[12] C. J. Roy, "Strategies for driving mesh adaptation in cfd (invited)," Tech. Rep. AIAA-2009-1302, 2009.

[13] X. D. Zhang, D. Pelletier, and J.-Y. Trepanier, "Verification of error estimators for the euler equations," Tech. Rep. AIAA-2000-1001, 2000.

[14] Y. Qin and T. I.-P. Shih, "A discrete transport equation for error estimation in cfd," Tech. Rep. AIAA-2002-0906, 2002.

[15] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[16] H. J. Stetter, "The defect correction principle and discretization methods," *Numerische Mathematik*, vol. 29, pp. 425–443, 1978.

[17] R. D. Skeel, "Thirteen ways to estimate global error," *Numerische Mathematik*, vol. 48, pp. 1–20, 1986.

[18] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, pp. 233–260, 1988.

[19] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, no. 1, pp. 40 – 69, 2002.

[20] T. S. Phillips and C. J. Roy, "A new extrapolation-based uncertainty estimator for computational fluid dynamics," *AIAA-2013-0260*, 2013.

[21] T. I.-P. Shih and B. R. Williams, "Development and evaluation of an a posteriori method for estimating and correcting grid-indeuced errors in solutions of the navier-stokes equations," Tech. Rep. AIAA-2009-1499, 2009.

[22] S. R. Fulton, "On the accuracy of multigrid truncation error estimates," *Electronic Transactions on Numerical Analysis*, vol. 15, pp. 29–37, 2003.

[23] F. Fraysse, J. de Vicente, and E. Valero, "The estimation of truncation error by τ-estimation revisited," *Journal of Computational Physics*, vol. 231, pp. 3457–3482, 2012.

[24] C. W. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, pp. 197–205, 1960. http://www.digizeitschriften.de/dms/img/?PPN=GDZPPN001163442.

[25] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Dok. Akad. Nauk SSSR*, vol. 4, pp. 240–243, 1963. Found reference: Gerstner and Briebel, Numerical Algorithms 18 (1998) 209-232.

[26] J. Burkardt, "Clenshaw curtis quadrature rules." `http://people.sc.fsu.edu/~jburkardt/m_src/ccn_rule/ccn_rule.html`, 2010.

[27] J. M. Derlaga, T. S. Phillips, and C. J. Roy, "Sensei computational fluid dynamics code: A case study in modern fortran software development," Tech. Rep. AIAA-2013-2450, 2013.

[28] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 2002.

[29] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[30] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[31] P. J. Roache, *Fundamentals of Verification and Validation*. Hermosa Publishers, Albuquerque, NM., 2009.

[32] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, New York, 2000.

[33] J. W. Banks, T. Aslam, and W. J. Rider, "On sub-linear convergence for linearly degenerate waves in capturing schemes," *Journal of Computational Physics*, 2008.

Appendix A

Fine Grid Truncation Error Estimation

The fine grid correction term is derived using the GTEE for the computational grid and the fine grid,

$$L_h(u_h) = I^h L(I_h^q u_h) + \tau_h(I_h^q u_h) \tag{3.41}$$

and

$$I_{h/r}^h L_{h/r}(I_q^{h/r} I_h^q u_h) = I^h L(I_h^q u_h) + I_{h/r}^h \tau_{h/r}(I_h^q u_h), \tag{3.42}$$

respectively. Solving for the continuous operator in Equation 3.41 and substituting into Equation 3.42

$$I_{h/r}^h L_{h/r}(I_q^{h/r} I_h^q u_h) = -\tau_h(I_h^q u_h) + I_{h/r}^h \tau_{h/r}(I_h^q u_h). \tag{3.43}$$

Assume that the truncation error is decreasing at an order p, $\tau_{h/r} = \tau_h / r^p$, then Equation 3.43 becomes

$$I_{h/r}^h L_{h/r}(I_q^{h/r} I_h^q u_h) = \tau_h(I_h^q u_h)\left(1/r^p - 1\right). \tag{3.44}$$

Solve for τ_h results in

$$\tau_h(I_h^q u_h) = -I_{h/r}^h L_{h/r}(I_q^{h/r} I_h^q u_h)\left(\frac{r^p}{r^p-1}\right). \tag{3.45}$$

Coarse Grid Truncation Error Estimation

The coarse grid correction term is derived using the GTEE for the computational grid and the coarse grid with the numerical solution substituted,

$$L_h(u_h) = I^h L(I_h^q u_h) + \tau_h(I_h^q u_h) \tag{3.46}$$

and

$$I_q^h I_{rh}^q L_{rh}(I_h^{rh} u_h) = I^h L(I_h^q u_h) + I_q^h I_{rh}^q \tau_{hr}(I_h^q u_h), \tag{3.47}$$

respectively. Solving for the continuous operator in Equation 3.46 and substituting into Equation 3.47

$$I_q^h I_{rh}^q L_{rh}(I_h^{rh} u_h) = -\tau_h(I_h^q u_h) + I_q^h I_{rh}^q \tau_{rh}(I_h^q u_h). \tag{3.48}$$

Again, assume that the truncation error is decreasing at an order p, $\tau_{rh} = r^p \tau_h$, then Equation 3.48 becomes

$$I_q^h I_{rh}^q L_{rh}(I_h^{rh} u_h) = \tau_h(I_h^q u_h)\,(r^p - 1)\,. \tag{3.49}$$

Solve for τ_h results in

$$\tau_h(I_h^q u_h) = I_q^h I_{rh}^q L_{rh}(I_h^{rh} u_h)\left(\frac{1}{r^p-1}\right). \tag{3.50}$$

Chapter 4

Error Transport Equation Boundary Conditions

Error Transport Equation Boundary Conditions for the Euler and Navier-Stokes Equations

Tyrone S. Phillips, Joseph M. Derlaga, Christopher J. Roy,
and Jeff Borggaard

Virginia Tech, Blacksburg, Virginia, 24061

Discretization error is usually the largest and most difficult numerical error source to estimate for computational fluid dynamics, and boundary conditions often contribute a significant source of error. Boundary conditions are described with a governing equation to prescribe particular behavior at the boundary of a computational domain. The boundary condition equations are considered sufficient when discretized with the same order of accuracy as the primary governing equations; however, careless implementations of boundary conditions can result in significantly larger numerical error with a reduced order of accuracy. Investigations into different numerical implementations of Dirichlet and Neumann boundary conditions for Burgers' equation show a significant effect on the accuracy of the resulting discretization error estimates using Richardson extrapolation and error transport equations. The development of boundary conditions for Burgers' equation shows significant improvements in discretization error estimates in general and a significant improvement in truncation error estimation which is key to accurate residual-based discretization error estimation. Scheme consistent and scheme inconsistent implementations of inflow and outflow boundary conditions up to fourth order accurate and a formulation for a slip wall boundary condition for truncation error estimation are developed for the Navier-Stokes and Euler equations. Similar trends compared to Burgers' equation truncation error are apparent with the scheme consistent form resulting in much smoother truncation error near the boundaries and more accurate solutions and discretization error estimation.

4.1 Introduction

Computational Fluid Dynamics (CFD) has enormous potential to impact the analysis, design, and optimization of engineering systems. The predictive capability of CFD depends not only on the validity of the sub-models employed and the uncertainties present in the system and surroundings, but also on the ability to accurately and reliably estimate and reduce numerical errors. While there are many sources of numerical error in CFD simulations, the largest and most difficult to estimate is the error related to the spatial grid, i.e., the spatial discretization error.

The discretization error is defined as the difference between the exact solution to the discrete

equations on a grid with cell size metric h, u_h, and the exact solution to the partial differential equations (PDEs) or integral equations, $\tilde{u}$:

$$\varepsilon_h = u_h - \tilde{u} \tag{4.1}$$

For solutions on sufficiently fine grids the discretization error will reduce with grid refinement. The source of discretization error is the truncated terms of the infinite series used to approximate the PDEs and is called truncation error. As the grid is refined, higher-order terms in the truncation error decrease at a faster rate than the leading term. The rate that truncation error decreases with grid refinement is proportional to the grid size parameter raised to the power of the formal order of accuracy of the numerical scheme, h^p. The grid size parameter is a measure of the discretization size in each coordinate direction (e.g., Δx) and the formal order of accuracy is the exponent of the leading truncation error term. Since truncation error can be shown to serve as the local source for the discretization error, the discretization error is expected to decrease at the same rate, atleast for smoother grids.

Several different classifications of discretization error estimators are available and each has different advantages and disadvantages. Recovery methods estimate the discretization error by comparing the discrete solution to a higher-order estimate of the exact solution found by post-processing the numerical solution(s). This category includes Richardson extrapolation,[1,2,3] order refinement,[4] and finite element recovery methods.[5,6] The most widely used discretization error estimator is Richardson extrapolation which uses a series of solutions with increasingly finer grid resolution to construct a higher order accurate solution. Richardson extrapolation is simple to apply as a post-processing step and does not require code modification. The disadvantage of Richardson extrapolation is the requirement of multiple grid levels, since grid generation is often one of the most time consuming tasks for a numerical simulation. In addition to the expense of generating multiple grid levels, each grid level used in Richardson extrapolation must also be in the asymptotic range in order for the resulting error estimate to be accurate. Residual-based methods use a discrete solution combined with information about the PDE or discrete equations being solved to estimate the discretization error. This category includes defect correction methods,[7,8,9,10] Error Transport Equations (ETEs),[11,12,13,14,15,16,17] finite element residual methods,[18,19,20] and adjoint methods.[21,22] Residual-based methods require only one grid level which reduces the grid generation requirements and the methods have the potential to be more accurate than Richardson extrapolation as the residual is an estimate of the truncation error which is directly related to the solution, the discretization scheme, and the grid. If the exact truncation error is used then the exact solution to the PDEs should be recovered. Accurate truncation error estimation is a key element for an accurate discretization error estimate in addition to the proper formulation of the specific method, see for example Phillips et al.[23]

The residual-based methods of interest are the discretization ETEs, which are derived by linearizing the original governing equations about a solution so that the discretization error is directly computed. ETEs have been applied to several simple model problems such as the advection/diffusion equation, Burgers' equation, wave equation, etc.[12,13,14] The ETEs were developed for laminar vis-

cous flow,[15] and where solved ETEs on much more complex problems including an airfoil, an aircraft engine exhaust, and a rocket plume.[16,17]

The ETEs must be properly formulated to accurately capture the transport of truncation error through the domain to compute an accurate discretization error estimate. Phillips and Roy[24] compared different linearization methods for Burgers' equation and show that improper linearization results in a discretization error estimate which is offset by a constant value. Two new formulations were given: the first corrected the offset but dropped the higher order nonlinear terms; while, the second formulation did not neglect the nonlinear terms. Little difference in the accuracy of the discretization error methods was seen for Burgers' equation; however, Banks et al.[25] showed that for problems with strong discontinuities the nonlinear terms are required for an accurate discretization error estimate. For the purposes of this study, the nonlinear formulation is used.

The ETEs must be derived for the governing equation. The derivation for the PDEs was the focus of past work; however, ETEs for boundary conditions must also be derived. Phillips and Roy[24] formulated ETEs for Burgers' equation for Dirichlet boundary conditions. The purpose of this work is to develop boundary condition implementations for the Euler and Navier-Stokes equations with a focus on truncation error estimation and discretization error estimation for ETEs.

4.2 Truncation Error Framework and Error Transport Equations

A set of operators are used to represent the governing equations and prolongation/restriction operations. Prolongation is an operation that transforms a lower-spaced solution to a higher-spaced solution (e.g. solution reconstruction), and restriction is an operation that transforms a higher-spaced solution to a lower-spaced solution. The numerical solution to a given set of PDEs can be represented by the General Truncation Error Expression (GTEE)[26,27]

$$L_h(\cdot) = L(\cdot) + \tau_h(\cdot) \tag{4.2}$$

where $L_h(\cdot)$ represents the discretized governing equations and operates on a discrete space, $L(\cdot)$ represents the continuous governing equations and operates on a continuous space, and $\tau_h(\cdot)$ represents the truncation error. Note that different truncation error equations can be derived by evaluating the GTEE on either continuous or discretized solutions and apply the needed prolongation or restriction operators. For example, if f is a continuous function then

$$L_h(I^h f) = I^h L(f) + \tau_h(f) \tag{4.3}$$

where I^h is an interpolation operator and is used to move from a continuous space to a discrete space with mesh size h. In general, we consider interpolation operators I_a^b where a is the starting space and b is the target space. The discrete space h represents a solution on a mesh with spacing h and a blank sub/superscript represents a continuous (infinite dimensional) space with infinite

degrees of freedom. The restriction of a continuous space to a discrete space is relatively straightforward. For a finite difference method the restriction could simply be considered as the evaluation of the continuous function at the i-th x location

$$f_{i,h} = I^h f = f(x_i). \tag{4.4}$$

For a finite volume method the natural restriction to a discrete space is the average value of the function over the i-th cell with volume V_i

$$f_{i,h} = I^h f = \frac{1}{V_i} \int_{V_i} f(x) dV. \tag{4.5}$$

If we consider a discretized function, then the GTEE can be written as

$$L_h(f_h) = I^h L(I_h f_h) + \tau_h(I_h f_h). \tag{4.6}$$

The prolongation from discrete space to continuous space is an approximation that is better written as

$$I_h = I_h^q + O(h^{q+1}) \tag{4.7}$$

to account for the error in the prolongation, where q is the order of the reconstructed polynomial for which the k-exact and ENO schemes are $q+1$ order accurate. Equation 4.6 is then written as

$$L_h(f_h) = I^h L(I_h^q f_h + O(h^{q+1})) + \tau_h(I_h^q f_h + O(h^{q+1}))$$

or

$$L_h(f_h) = I^h L(I_h^q f_h) + \tau_h(I_h^q f_h) + O(h^{\bar{q}}) \tag{4.8}$$

where $\bar{q}$ is an order of accuracy which is related to q may be modified by nonlinearities in the $L(.)$ and $\tau_h(.)$ operators. Due to the error in the prolongation, it cannot be generally assumed that $f_h = I^h I_h f_h$.

As is evident from the GTEE, the truncation error is the difference between the discrete and continuous governing equations and has been shown to be the local source of discretization error in a numerical solution[27] where the discrete form of the discretization error is defined as

$$\varepsilon_h = u_h - I^h \tilde{u}. \tag{4.9}$$

Here, u_h is the exact solution to the discrete equations such that $L_h(u_h) = 0$, and $\tilde{u}$ is the exact solution to the original governing equations

The truncation error can be shown to be the local source of discretization error through the general derivation of the discrete error transport equations using the GTEE framework given by Roy.[26,27] Starting from the GTEE

$$L_h(I^h \tilde{u}) - I^h L(\tilde{u}) = \tau_h(\tilde{u}) \tag{4.10}$$

subtracting $L_h(u_h) = 0$, and noting that $L(\tilde{u}) = 0$

$$L_h(u_h) - L_h(I^h\tilde{u}) = -\tau_h(\tilde{u}) \tag{4.11}$$

If the discrete operator is linear or linearized then $L_h(u_h) - L_h(I^h\tilde{u}) = L_h(\varepsilon_h)$ which gives

$$L_h(\varepsilon_h) = -\tau_h(\tilde{u}). \tag{4.12}$$

The new discrete operator is a discrete equation that governs the transport of disretization error through the computational domain with truncation error as the source. The relationship between discretization error and truncation error is the basis of residual-based discretization error estimation. A similar process can be followed to derive the continuous error transport equations by substituting $I_h u_h$ into the GTEE and subtracting $L(\tilde{u})$ from both sides which results in

$$L(\varepsilon) = \tau_h(I_h u_h). \tag{4.13}$$

The form of the continuous ETE is similar to the discrete ETE; however, the continuous ETE must then be discretized and solved. This focus of this research is the discrete ETE. The accuracy of the discretization error estimate is determined by the accuracy of the truncation error estimate which is related through the following expression

$$\tau_h(\tilde{u}) \approx \tau_h(I_h u_h) \approx \tau_h\left(I_h^k u_h + O(h^{k+1})\right). \tag{4.14}$$

Truncation error tends to be larger near the boundaries for specific boundary condition compared to the truncation error of the interior discretization scheme. As a central focus of accurate truncation error estimation, different discretization methods for boundary conditions are investigated with the goal to reduce the magnitude of the boundary truncation error to the same order of magnitude as the interior scheme. This has the effect of improving truncation error smoothness and improves the accuracy of the reconstruction scheme. Also, reducing the truncation error near the boundary should improve solution accuracy since the local source of discretization error is reduced. Smooth truncatation error is also advantagious for solution adaption where truncation error is used as the adaptive driver.[28]

4.3 Governing Equations

4.3.1 Burgers' Equation

Burgers' equation is a simple scalar model equation for the Navier-Stokes equations with a non-linear convection term and a linear diffusion term with constant viscosity. The strong form is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = S \tag{4.15}$$

For the unsteady finite volume formulation the continuous operator is

$$L(u) = \frac{\partial u}{\partial t} + \int F(u)dx - S = 0 \tag{4.16}$$

where the flux is $F(u) = \frac{u^2}{2} - \nu\frac{\partial u}{\partial x}$ and S is a general source term. Our work will focus on steady-state solutions, but the temporal term will be retained to aid in marching in pseudo-time towards a steady-state solution.

The interface flux is discretized using a central flux scheme

$$F_h(u_i, u_{i+1}) = \frac{u_{i+1}^2 + u_i^2}{4} - \frac{\nu(u_{i+1} - u_i}{\Delta x} \tag{4.17}$$

which is formally second-order accurate. The discrete operator is

$$L_h(u_i) = \frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{F_h(u_i^n, u_{i+1}^n) - F_h(u_{i-1}^n, u_i^n)}{\Delta x} - S_i \tag{4.18}$$

and is solved by marching through time with an explicit time marching scheme to steady state.

Boundary Conditions

Two different formulations for Dirichlet and Neumann boundary conditions are investigated. Each boundary condition can be implemented in a scheme inconsistent or a scheme consistent way referring to how the boundary conditions are applied. The scheme inconsistent formulation sets the boundary condition at the boundary face for the prescribed boundary condition. The scheme consistent formulation is applied at ghost cells (additional computational cells added to the edge of the domain) and the interior flux scheme is used to compute the boundary flux.

Dirichlet boundary conditions prescribe the solution value from the known exact solution. For the scheme consistent formulation, the average value over the ghost cell ($i = 0$) for one dimension is computed

$$u_0 = \frac{1}{\Delta x}\int_{x_{-1/2}}^{x_{1/2}} \tilde{u}(x)dx. \tag{4.19}$$

Extension to two- and three-dimensional implementations are straight forward. The scheme inconsistent formulation is applied at the boundary face where

$$u_{1/2} = \tilde{u}(x_{1/2}) \tag{4.20}$$

where the $1/2$ index denotes a face and the integers denote cell averaged values. To compute the Burgers' equation flux, the derivative at the face must also be computed. Two different one-sided difference equations are derived and are consistent with the finite volume formulation. The one-sided differences are derived from a polynomial reconstruction in general coordinates consistent

with the finite volume method. The mesh transformation is implicitly included in the one-sided difference under the assumption of a smooth grid transformation. The third-order accurate one-sided difference is

$$\left(\frac{\partial u}{\partial x}\right)_{1/2} = -\frac{6u_{1/2} + 9u_1 - u_2}{3\Delta x} \tag{4.21}$$

and the fourth-order accurate formulation is

$$\left(\frac{\partial u}{\partial x}\right)_{1/2} = -\frac{66u_{1/2} - 85u_1 + 23u_2 - 4u_3}{18\Delta x} \tag{4.22}$$

When applied to the right boundary for n cells at $x_{n+1/2}$, the one-sided difference is modified by negating the one-sided difference.

Neumann boundary conditions prescribe the solution gradient. For both Neumann boundary conditions considered, the solution gradient is specified at the face so that

$$\left(\frac{\partial u}{\partial x}\right)_{1/2} = \frac{\partial}{\partial x}\tilde{u}(x_{1/2}). \tag{4.23}$$

The scheme consistent formulation uses the solution gradient at the boundary face to compute a ghost cell solution value which is then used to compute the flux at the boundary face from the flux scheme. Two different formulations for a backward extrapolation are derived. The third-order formulation is

$$u_0 = u_1 - \left(\frac{\partial u}{\partial x}\right)_{1/2} \Delta x \tag{4.24}$$

and the fourth-order formulation is

$$u_0 = \frac{-12\left(\frac{\partial u}{\partial x}\right)_{1/2}\Delta x + 9u_1 + 3u_2 - u_3}{11}. \tag{4.25}$$

When applied to the right boundary, the Neumann boundary condition is

$$\left(\frac{\partial u}{\partial x}\right)_{n+1/2} = \frac{\partial}{\partial x}\tilde{u}(x_{n+1/2}). \tag{4.26}$$

The third-order forward extrapolation is

$$u_{n+1} = u_n + \left(\frac{\partial u}{\partial x}\right)_{n+1/2} \Delta x, \tag{4.27}$$

and the fourth-order formulation is

$$u_{n+1} = \frac{12\left(\frac{\partial u}{\partial x}\right)_{1/2}\Delta x + 9u_n + 3u_{n-1} - u_{n-1}}{11}. \tag{4.28}$$

The scheme inconsistent formulation computes the flux at the boundary face using the prescribed solution derivative and a backwards extrapolation for the solution value. The third-order extrapolation is

$$u_{1/2} = -\frac{2\left(\frac{\partial u}{\partial x}\right)_{1/2}\Delta x - 7u_1 + u_2}{6} \tag{4.29}$$

and the fourth-order extrapolation is

$$u_{1/2} = -\frac{18\left(\frac{\partial u}{\partial x}\right)_{1/2}\Delta x - 85u_1 + 23u_2 - 4u_3}{66}. \tag{4.30}$$

For the right boundary the extrapolations are

$$u_{n+1/2} = -\frac{-4\left(\frac{\partial u}{\partial x}\right)_{n+1/2}\Delta x - 14u_n + 2u_{n-1}}{12} \tag{4.31}$$

and

$$u_{n+1/2} = -\frac{-72\left(\frac{\partial u}{\partial x}\right)_{n+1/2}\Delta x - 340u_n + 92u_{n-1} - 16u_{n-2}}{264}. \tag{4.32}$$

Error Transport Equations

The discrete error transport equation for Burgers' equation is

$$L_h(\varepsilon_h) = \frac{\partial \varepsilon_h}{\partial t} + \int F_h(\varepsilon_h)dx = \tau_h(\tilde{u}) \tag{4.33}$$

where the general error flux is derived using the equation

$$F_h(\varepsilon_h) = F_h(u_h) - F_h(I^h\tilde{u}). \tag{4.34}$$

Since the flux is nonlinear, the definition for discretization error is substituted into the discrete flux

$$F_h(\varepsilon_i, \varepsilon_{i+1}) = -\frac{\varepsilon_{i+1}^2 + \varepsilon_i^2}{4} + \frac{u_{i+1}\varepsilon_{i+1} + u_i\varepsilon_i}{2} - \nu\frac{\varepsilon_{i+1} - \varepsilon_i}{\Delta x}. \tag{4.35}$$

For comparison to the discrete flux, the continuous error transport flux is derived by substituting the definition of discretization error into the continuous flux

$$I^h F(\varepsilon_h) = \frac{u_h^2}{2} - \nu \frac{\partial u_h}{\partial x} - \left(\frac{(u_h - \varepsilon_h)^2}{2} - \nu \frac{\partial (u_h - \varepsilon_h)}{\partial x} \right) = -\frac{\varepsilon_h^2}{2} + u_h \varepsilon_h - \nu \frac{\partial \varepsilon_h}{\partial x}. \tag{4.36}$$

The steady state discretization error is computed using the same explicit time marching method as the original equations

$$L_h(\varepsilon_i) = \frac{\varepsilon_i^{n+1} - \varepsilon_i^n}{\Delta t} + \frac{F_h(\varepsilon_i^n, \varepsilon_{i+1}^n) - F_h(\varepsilon_{i-1}^n, \varepsilon_i^n)}{\Delta x} - \tau_h(\tilde{u}). \tag{4.37}$$

Note that the source term S_i is not included in the solution to the error transport equations, because this term cancels during the derivation of the ETE.

Boundary Conditions

The boundary conditions are derived from the boundary conditions used to compute the numerical solution. Since the boundary conditions are linear, the derivation of the ETEs is straight forward. For the scheme consistent Dirichlet formulation

$$\varepsilon_0 = 0. \tag{4.38}$$

For the scheme inconsistent Dirichlet formulation

$$\varepsilon_{1/2} = 0. \tag{4.39}$$

The gradient of the discretization error is computed using a third-order one-sided difference

$$\left(\frac{\partial \varepsilon}{\partial x} \right)_{1/2} = -\frac{9\varepsilon_1 - \varepsilon_2}{3\Delta x} \tag{4.40}$$

or a fourth-order backward difference

$$\left(\frac{\partial \varepsilon}{\partial x} \right)_{1/2} = -\frac{65\varepsilon_1 + 23\varepsilon_2 - 4\varepsilon_3}{18\Delta x} \tag{4.41}$$

For both Neumann boundary condition implementations, the error in the gradient at the face is zero

$$\left(\frac{\partial \varepsilon}{\partial x} \right)_{1/2} = 0 \tag{4.42}$$

The scheme consistent Neumann boundary condition sets the ghost cell value using a third-order extrapolation

$$\varepsilon_0 = \varepsilon_1 \tag{4.43}$$

or a fourth-order extrapolation

$$\varepsilon_0 = \frac{9\varepsilon_1 + 3\varepsilon_2 - \varepsilon_3}{11}. \tag{4.44}$$

The scheme inconsistent Neumann boundary condition computes the flux at the boundary face using the prescribed solution derivative and the solution extrapolated through the face using a third-order extrapolation

$$\varepsilon_{1/2} = -\frac{-7\varepsilon_1 + \varepsilon_2}{6} \tag{4.45}$$

and a fourth-order extrapolation

$$\varepsilon_{1/2} = -\frac{-85\varepsilon_1 + 23\varepsilon_2 - 4\varepsilon_3}{66}. \tag{4.46}$$

4.3.2 Navier-Stokes

The finite volume formulation of the Navier-Stokes equations is

$$L(\vec{U}) = \frac{\partial}{\partial t}\iiint_V \vec{U} dV + \iint_A \left(\vec{F}_i - \vec{F}_v\right) d\vec{A} - \iiint_V \vec{S} dV = 0 \tag{4.47}$$

where the inviscid flux is

$$\vec{F}_i = \begin{bmatrix} \rho\vec{v}\cdot\hat{n} \\ \rho u\vec{v}\cdot\hat{n} + n_x p \\ \rho v\vec{v}\cdot\hat{n} + n_y p \\ \rho w\vec{v}\cdot\hat{n} + n_z p \\ \rho h_t \vec{v}\cdot\hat{n} \end{bmatrix} \tag{4.48}$$

and the viscous flux is

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\theta_x + n_y\theta_y + n_z\theta_z \end{bmatrix} \tag{4.49}$$

The outward pointing face normal at the grid face is $\hat{n} = [n_x, n_y, n_z]$, the conserved variable vector is $\vec{U} = [\rho, \rho u, \rho v, \rho w, \rho e_t]$, the velocity vector is $\vec{V} = [u, v, w]$, the total energy e_t and total enthalpy h_t are computed using the calorically perfect gas assumption with an equation of state $p = \rho RT$,

$e = \frac{p}{\rho(\gamma-1)}$, and $h_t = \frac{p\gamma}{\rho(\gamma-1)} + \frac{|\vec{V}|^2}{2}$. The diffusive components of the viscous flux are

$$\tau_{xx} = 2\mu\left(\frac{\partial u}{\partial x} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{yy} = 2\mu\left(\frac{\partial u}{\partial y} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{zz} = 2\mu\left(\frac{\partial u}{\partial z} - \frac{1}{3}\nabla\cdot\vec{V}\right),$$
$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right), \quad \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right),$$
$$\theta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x}, \quad \theta_y = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y}, \quad \theta_z = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z}$$

and the heat conduction coefficient for a calorically perfect gas is $k = \frac{R\gamma\mu}{Pr(\gamma-1)}$. For all simulations, $Pr = 0.71$ and viscosity is set to a constant of $10Pa\cdot s$ so that diffusive terms are on the same order of magnitude as the convective terms. The Navier-Stokes equations are solved using a structured grid where the numerical solution is the cell-averaged value for a cell with volume V

$$\vec{U}_h = I^h\vec{U} = \frac{1}{V}\iiint_V \vec{U}(x)dx. \tag{4.50}$$

The discrete operator for the Navier-Stokes equations is

$$L_h(\vec{U}_h) = \frac{\vec{U}_h^{n+1} - \vec{U}_h^n}{\Delta t} + \frac{1}{V}\sum_{j=1}^{J_{max}} F(U_{h,j}^L, U_{h,j}^R)\cdot\hat{n}_j A_j - S_h = 0, \tag{4.51}$$

where n is the current time step and j is the face for a given cell with spacing h. The discrete equations are solved using an explicit time marching scheme to compute steady state solutions. The function F is a Riemann solver which operates on the left and right n^{th} cell face approximations $\vec{U}_L$ and $\vec{U}_R$. The flux scheme developed by van Leer[29] is used for all solutions. The face approximations are second order accurate using MUSCL extrapolation which has a total stencil width of five cells in each coordinate direction. For a given face, the left and right states are computed

$$u_{i+1/2}^R = u_{i+1} - \frac{1}{4}\left[(1-\kappa)(u_{i+2} - u_{i+1}) + (1+\kappa)(u_{i+1} - u_i)\right] \tag{4.52}$$

$$u_{i+1/2}^L = u_i + \frac{1}{4}\left[(1-\kappa)(u_i - u_{i-1}) + (1+\kappa)(u_{i+1} - u_i)\right] \tag{4.53}$$

where κ determines the order of accuracy and direction of the extrapolation. For all simulations, $\kappa = -1$ which is a fully upwinded second-order extrapolation.

Boundary Conditions

There are two different formulations of boundary conditions considered for the Navier-Stokes equations. The extent of the stencil for MUSCL extrapolation requires two cells to the left and right of each face. A boundary condition must be set for the boundary face $u_{1/2}$ to compute the flux at the boundary and a ghost cell at u_0 must be set to compute the left face for the first interior face $u_{3/2}$ from MUSCL extrapolation. The scheme inconsistent formulation sets the outside

state of the boundary face $u^L_{1/2}$, and the ghost cell u_0 is set from a higher order extrapolation from the interior which includes the left boundary face in the extrapolation. The scheme consistent formulation sets the first ghost cell u_0, and the second ghost cell u_{-1} is set from a higher order extrapolation. With two ghost cells, the interior MUSCL extrapolation scheme is used to set the outside boundary face $u^L_{1/2}$.

For example, consider a Dirichlet boundary condition for a function $\bar{f}(\vec{x})$. For the scheme inconsistent formulation, the boundary face is set to the face-averaged solution with area $A_{1/2}$

$$u^L_{1/2} = \frac{1}{A_{1/2}} \int_{A_{1/2}} \bar{f}(\vec{x}_{1/2}) d\vec{A} \tag{4.54}$$

and the first ghost cell is set using a second-order extrapolation using the boundary face and the first interior cell

$$u_0 = 2u^L_{1/2} - u_1 \tag{4.55}$$

For the scheme consistent formulation, the first ghost cell is set to the cell-averaged solution

$$u_0 = \frac{1}{V_0} \int_{V_0} \bar{f}(\vec{x}_0) dV \tag{4.56}$$

and the second ghost cell is set using a second-order extrapolation

$$u_{-1} = 2u_0 - u_1 \tag{4.57}$$

A second-order approximation to the face-averaged and cell-averaged solution is the function evaluated at the geometric face center and cell center; however, order of accuracy for extrapolation boundary condition implementations include second-, third-, and fourth-order. To prevent degradation in the order of accuracy, the face and cell integral is approximated using a Curtis-Clenshaw[30] higher-order quadrature. The boundary conditions under investigation are Dirichlet, supersonic inflow, subsonic inflow, supersonic outflow, subsonic outflow, and a slip wall. The number of characteristics that enter the domain determine how many boundary conditions are set, and the number of characteristics the leave the domain determine how many variables are extrapolated from the interior. In a one-dimensional sense, the subsonic inflow boundary condition has two characteristics entering the domain and one characteristic leaving the domain and is implemented so that density and velocity are set and pressure is extrapolated to the boundary. Supersonic inflow has three characteristics entering the domain so all primitive variables are set. The subsonic outflow boundary condition has one characteristic entering the domain so pressure is specified and density and velocity are extrapolated. The supersonic outflow boundary condition has no characteristics entering the domain so all primitive variables are extrapolated to the boundary. For the slip wall, the boundary condition is zero normal velocity $\vec{V} \cdot \hat{n} = 0$. There are four different extrapolations which are used to set the boundary conditions. All extrapolations are derived using Barth's k-exact reconstruction[31,32] from a one-dimensional, polynomial reconstruction which is consistent with the finite-volume method (i.e. for a general numerical solution u_h, $u_h = I^h_k I^k_h u_h$) where a q^{th}-

order polynomial takes the general form of $p(x) = \sum_{i=0}^{q} a_i x^i$. The first extrapolation is used for the scheme inconsistent boundary conditions which are derived from a one-sided reconstruction using a specified value at the boundary face and the numerical solution from interior cells. For a second-order extrapolation, the constraints for the polynomial reconstruction require the polynomial evaluated at the face is equal to the prescribed face value and the average of the polynomial over the first interior cell is equal to the numerical solution in the cell

$$u_{1/2} = p(x_{1/2}) \tag{4.58}$$

and

$$u_1 = \frac{1}{V_1} \int_{V_1} p(x) dV. \tag{4.59}$$

The polynomial reconstruction is then averaged over the ghost cell volume

$$f_0 = \frac{1}{V_0} \int_{V_0} p(x) dV. \tag{4.60}$$

Third-order adds a constraint from the second interior cell and fourth-order adds constraints from the second and third interior cells. The resulting extrapolation equations are summarized in Table A.1. The next three extrapolations are derived in a similar manner but use only grid cells in the reconstruction. The polynomial reconstruction constraints for a second-order method requires that the polynomial evaluated over the first two interior cells reproduce the numerical solution in the two cells

$$u_1 = \frac{1}{V_1} \int_{V_1} p(x) dV \tag{4.61}$$

and

$$u_2 = \frac{1}{V_2} \int_{V_2} p(x) dV \tag{4.62}$$

where V_1 is the cell volume of the first interior cell and V_2 is the cell volume of the second interior cell. The second method evaluates the reconstruction polynomial over the first ghost cell ($i = 0$)

$$g_0 = \frac{1}{V_0} \int_{V_0} p(x) dV \tag{4.63}$$

the third method evaluates the reconstruction polynomial over the second ghost cell ($i = -1$)

$$g_{-1} = \frac{1}{V_{-1}} \int_{V_{-1}} p(x) dV \tag{4.64}$$

and the fourth method evaluates the reconstruction polynomial at the boundary face

$$h_{1/2} = p(x_{1/2}). \tag{4.65}$$

A summary of the resulting extrapolation equations to the first ghost cell and to the face is pre-

sented in Table A.2 and Table A.3. The boundary conditions as they are implemented are shown in Table A.4 and Table A.5 for the scheme inconsistent and scheme consistent boundary conditions, respectively. The extrapolation equations to the second ghost cell are not necessary because the reconstruction polynomial used to derive g_0 is the same as g_{-1} with the only difference being the cell reconstruction polynomial is evaluated over a different cell. The final equation for extrapolating to the first ghost cell can be used in place of deriving an extrapolation equation for the second ghost cell by shifting the stencil (i.e., $g_{-1}(u_1,u_2) = g_0(u_0,u_1) = g_0(g_0(u_1,u_2),u_1)$).

The slip wall is implemented by exactly setting the flux at the boundary to satisfy the wall boundary conditions $\vec{V}\cdot\hat{n} = 0$

$$\vec{F}_{wall} = \begin{bmatrix} 0 \\ n_x p_{1/2} \\ n_y p_{1/2} \\ n_z p_{1/2} \\ 0 \end{bmatrix}. \tag{4.66}$$

The pressure at the wall is extrapolated to the face $p_{1/2} = h_{1/2}(p_1, p_2, \ldots, p_n)$. The inviscid flux in Equation 4.47 is modified for the appropriate boundaries to use F_{wall} instead of F_i given in Equation 4.48.

Error Transport Equation

To derive the nonlinear error transport equations for the Navier-Stokes equations, the discretization error inserted into the flux terms and expanding results in a very complicated expression. The energy term in the inviscid flux, $\rho h_t u$, expands to 38 terms for a Cartesian grid (i.e. $\hat{n} = [1,0,0]$). The Reimann solver would also have to be linearized with varying degrees of complexity depending on the flux scheme. To take into account all of the nonlinear terms in the least code intrusive way, the discrete operator given by Equation 4.51 is inserted into the ETE expression given by Equation 4.11 which results in

$$L_h(\vec{\varepsilon}_h) = \frac{\vec{\varepsilon}_h^{n-1} - \vec{\varepsilon}_h^{n}}{\Delta t} + \frac{1}{V}\sum_{j=1}^{j_{max}} F(\vec{U}_{h,j}^L, \vec{U}_{h,j}^R)\cdot\hat{n}_j A_j - \frac{1}{V}\sum_{j=1}^{j_{max}} F(\vec{U}_{h,j}^L - \vec{\varepsilon}_{h,j}^L, \vec{U}_{h,j}^R - \vec{\varepsilon}_{h,j}^R)\cdot\hat{n}_j A_j + \tau_h(\vec{U}) = 0. \tag{4.67}$$

The second term is a function of the old solution and can be pre-computed. The left and right face discretization errors are computed using MUSCL extrapolation modified for the error transport equations

$$\varepsilon_{i+1/2}^R = \varepsilon_{i+1} - \frac{1}{4}\left[(1-\kappa)(\varepsilon_{i+2} - \varepsilon_{i+1}) + (1+\kappa)(\varepsilon_{i+1} - \varepsilon_i)\right] \tag{4.68}$$

$$\varepsilon_{i+1/2}^L = \varepsilon_i + \frac{1}{4}\left[(1-\kappa)(\varepsilon_i - \varepsilon_{i-1}) + (1+\kappa)(\varepsilon_{i+1} - \varepsilon_i)\right] \tag{4.69}$$

MUSCL extrapolation is linear, so the ETE form is trivial. This also holds for the boundary condition extrapolations making the implementation of the ETE boundary conditions trivial. With the

exception of adding the second term in the discrete ETE operator and the ETE MUSCL extrapolation and boundary conditions, the solution process is unchanged.

4.4 Applications

4.4.1 Burgers' Equation Manufactured Solution

A sinusoidal manufactured solution (see for example Roache[33] and Oberkampf and Roy[34]) is used to investigate the truncation error and ETEs for Burgers' equation

$$\tilde{u}(x) = 4 + 3.5cos(\frac{x}{2}\pi). \tag{4.70}$$

The domain is $x = [-2.5, 2.5]$. The chosen solution is not an exact solution to Burgers' equation so a source term is added to the right-hand side of Burgers' equation. The source term is computed using the weak form of the MMS used by Derlaga et al.[35]

$$S_i = F\left(\tilde{u}\left(x_{i+1/2}\right)\right) - F\left(\tilde{u}\left(x_{i-1/2}\right)\right). \tag{4.71}$$

The domain and solution were chosen so that the gradient is not near zero to ensure larger truncation error near the boundaries. Different manufactured solutions were considered with varying boundary gradients. The results discussed for the chosen manufactured solution were consistent for all manufactured solutions considered.

4.4.2 Euler and Navier-Stokes Equations

Manufactured Solution

Two different manufactured solutions on a Cartesian and a curvilinear grid are used to investigate boundary condition truncation error and error transport equation discretization error estimation. Both manufactured solutions have the same form; however, the first is subsonic and the second is supersonic. The general form for both manufactured solutions is

$$f(x,y) = a_0 + a_x sin\left(\frac{b_x \pi x}{L} + c_x \pi\right) + a_y sin\left(\frac{b_y \pi y}{L} + c_y \pi\right) + a_{xy} sin\left(\frac{b_{xy} \pi xy}{L^2} + c_{xy} \pi\right). \tag{4.72}$$

The coefficients for the supersonic manufactured solution are shown in Table 4.1 and the coefficients for the subsonic manufactured solution are identical except the a coefficients for velocity are divided by ten. Sample grids with x-velocity exact solutions on a 33x33 grid are shown in Figure 4.1. Five different grid levels are used to compare convergence properties of the discretization error and truncation error estimation methods. This includes 17x17, 33x33, 65x65, 129x129, and

257x257 grid nodes.

Table 4.1: Supersonic manufactured solution coefficients

	a_0	a_x	b_x	c_x	a_y	b_y	c_y	a_{xy}	b_{xy}	c_{xy}
ρ	1.0	0.15	2.0	0.33	-0.1	1.0	-0.2	-0.05	1.0	0.25
u	800.0	50.0	0.66	0.5	-30.0	1.5	-0.125	-60.0	1.5	0.125
v	800.0	-75.0	1.66	-0.5	40.0	3.0	0.0	60.0	1.5	0.125
P	1.e5	-2.e4	0.5	0.5	5.e4	1.0	-0.5	-1.e4	0.75	-0.25

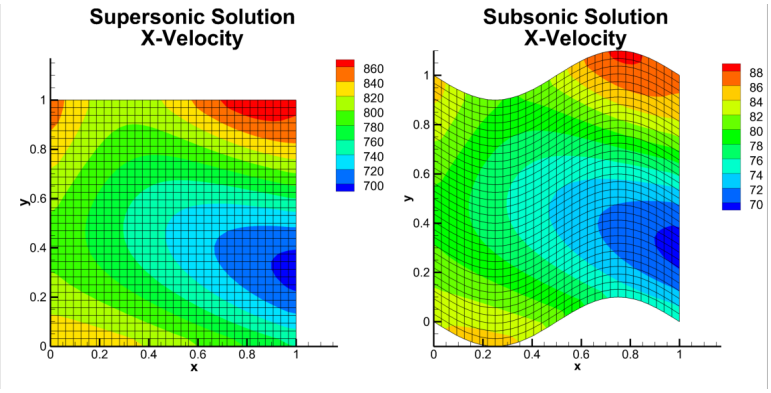


Figure 4.1: Cartesian and curvilinear grids showing the x-velocity contour for the supersonic manufactured solution (left) and the subsonic manufactured solution (right)

Supersonic Vortex Flow

Supersonic vortex flow[36] consists of a flow around a 90 degrees annulus and is an exact solution to the Euler equations. This numerical simulation requires three different boundary conditions: streamlines which are modeled as slip walls, a Dirichlet inflow profile, and a supersonic outflow boundary condition. Supersonic vortex flow has an exact solution so a numerical benchmark is not required; however, the boundary conditions introduce enough complexity for adequate evaluation of error estimators without the presence of numerical error. The exact solution is

$$\rho(r) = \rho_i \left(1 + \frac{\gamma-1}{2} M_i^2 \left(1 - \frac{R_i^2}{r^2}\right)\right)^{\frac{1}{\gamma-1}} \tag{4.73}$$

$$u(y,r) = \frac{yU}{r}, \quad v(x,r) = -\frac{xU}{r}, \quad P = \frac{\rho^\gamma}{\gamma} \tag{4.74}$$

$$U_i = M_i \rho_i^{\frac{\gamma-1}{2}}, \quad U = \frac{U_i R_i}{r} \tag{4.75}$$

The flow field is defined as a function of variables at the inner radius of the annulus denoted by the subscript i. The inner radius, r_i, is 2.0, the outer radius is 3.0, the inner density, ρ_i, is $1.0 kg/m^3$, and the inner Mach number, M_i, is 2.0. The inner and outer streamlines are modeled as walls shown in Figure 4.2. Four different grid levels are used to investigate the wall boundary condition implementation: 33x17, 65x33, 129x65, and 257x129.

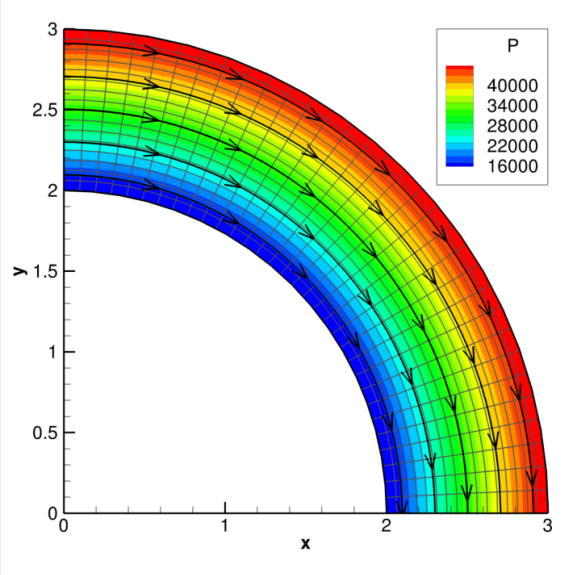


Figure 4.2: Supersonic vortex flow pressure contour and streamlines on the 33x17 node grid

4.5 Results

4.5.1 Burgers' Equation

The results investigate two different points regarding Neumann boundary conditions and the ETEs. First the exact truncation error for the different boundary conditions is computed, and second, the ETEs are solved for different combinations of boundary conditions using an estimated truncation error and compared to Richardson extrapolation.

Boundary Condition Truncation Error

The truncation error for the Neumann boundary conditions at the outflow is shown for $n = 33$ and $n = 65$ in Figure 4.3. Fourth-order extrapolations were used for both boundary conditions. The scheme inconsistent formulation of the Neumann boundary condition has a significantly larger error than the scheme consistent formulation. Although not shown, the truncation error for the scheme inconsistent Neumann boundary condition is only first order accurate and is significantly

larger than the interior truncation error. The scheme consistent Neumann boundary condition decreases at a second-order rate despite the fourth-order extrapolation because the boundary truncation error is dominated by the interior second-order accurate discretization scheme truncation error.

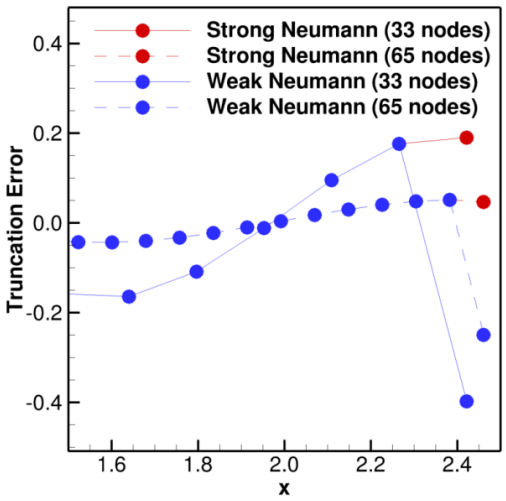


Figure 4.3: Truncation error for the scheme consistent and scheme inconsistent Neumann boundary condition formulation for Burgers' equation

Further investigation regarding the difference between the scheme consistent and scheme inconsistent Dirichlet boundary conditions showed similar results. Additionally, second-order and third-order extrapolations were also considered for the scheme consistent Neumann boundary conditions with both resulting in first-order accurate truncation error. Only the fourth-order extrapolation resulted in second-order accurate truncation error for the boundaries. A possible reason for the first-order behavior could be due to error cancelation required to achieve the required order of accuracy. For the scheme inconsistent formulation, the interior flux scheme is not applied at the boundary face. This is tested by reformulating the scheme inconsistent Dirichlet boundary condition. The prescribed solution value at the boundary face is used in addition to the first few interior cells to extrapolate a solution value to the ghost cell. The interior flux scheme is then used to compute the flux at the face resulting in second-order reduction in truncation error at the boundary. The Neumann boundary condition cannot be reformulated in the same manner so the scheme consistent Neumann boundary condition should be used instead. Regardless of the order of accuracy of the boundary condition truncation error, the order of accuracy of the discretization error in the domain is second-order for all cases as expected due to error cancelation.

Discretization Error Estimation

Discretization error estimates are shown for 33 node solutions. The truncation error used to compute the ETE solution is estimated from the numerical solution. Figure 4.4 uses the scheme consistent Dirichlet boundary conditions. The ETE estimate is slightly better overall. Richardson extrapolation does not capture the slight change in discretization error at the boundaries for this grid.

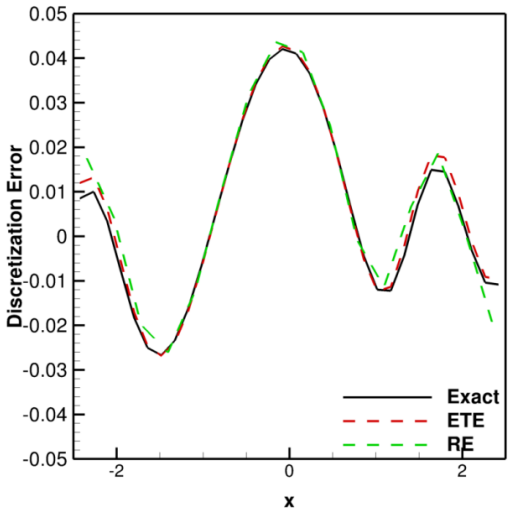


Figure 4.4: Discretization error for $N = 33$ with scheme consistent Dirichlet boundary conditions

Figure 4.5 is computed using the scheme consistent Neumann outflow boundary condition. Both Richardson extrapolation and the ETE estimates compare well to the exact error. The discrepancies between the two are largest at the boundaries with only slightly better agreement by the ETE estimate. Figure 4.6 is computed using the scheme inconsistent Neumann outflow boundary condition. In this case, the large error due to the scheme inconsistent formulation significantly affects the error estimate for both Richardson extrapolation and the ETE estimate. Richardson extrapolation near the outflow matches the exact discretization error better than the ETE solution. This is possibly due to the accuracy of the truncation error estimation. While the order of accuracy of the scheme inconsistent Neumann boundary condition did not seem to affect the discretization error order of accuracy, the effect is clear in both the Richardson extrapolation error and ETE estimates resulting in significantly larger discretization error and degraded accuracy near the boundary. This result provides further motivation for using the scheme consistent Neumann boundary condition.

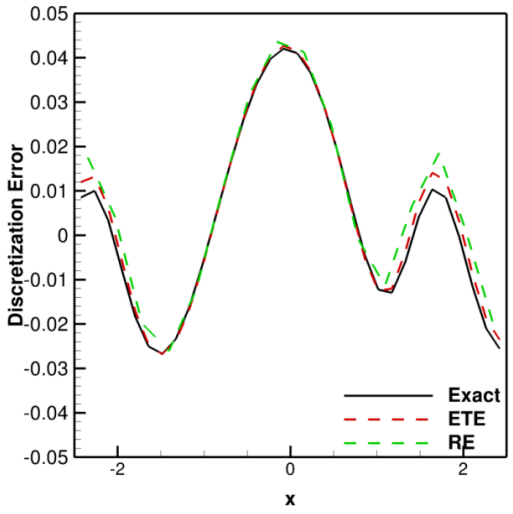


Figure 4.5: Discretization error for $N = 33$ with a scheme consistent Neumann outflow boundary condition

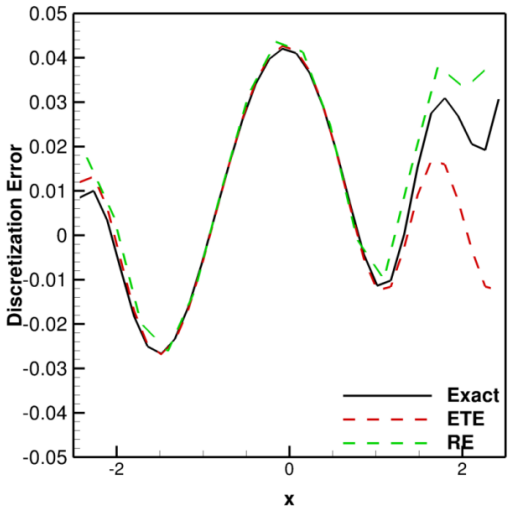


Figure 4.6: Discretization error for $N = 33$ with a scheme inconsistent Neumann outflow boundary condition

4.5.2 Euler and Navier-Stokes Equations

Truncation Error Analysis

Several different combinations of manufactured solutions and grid topology are used to investigate the effects of boundary condition extrapolation order and boundary condition implementation for the Euler or Navier-Stokes equations. The Euler supersonic manufactured solution truncation error for the series of Cartesian grids is shown in Figure 4.7 for the scheme inconsistent boundary condition implementation using a second-order extrapolation. Note that the pixelation of the contour plots are due to the fact that the true cell-averaged values produced by the finite volume method are plotted. A slice of the truncation error taken at the centerline $y = 0.5$ is also shown. The effect of the boundary condition implementation extends to the first two interior cells. Similar to the scheme inconsistent implementation for Burgers equation, the boundary condition truncation error decreases at a first-order rate while the interior truncation error decreases at a second-order rate. The first-order rate is observed through the contour plot where the legend is scaled to the maximum and minimum value over the entire domain and through the truncation error slice. The peak truncation error decreases by half with each successive grid level, and for the finer grids, the boundary condition truncation error is significantly larger than the interior truncation error.

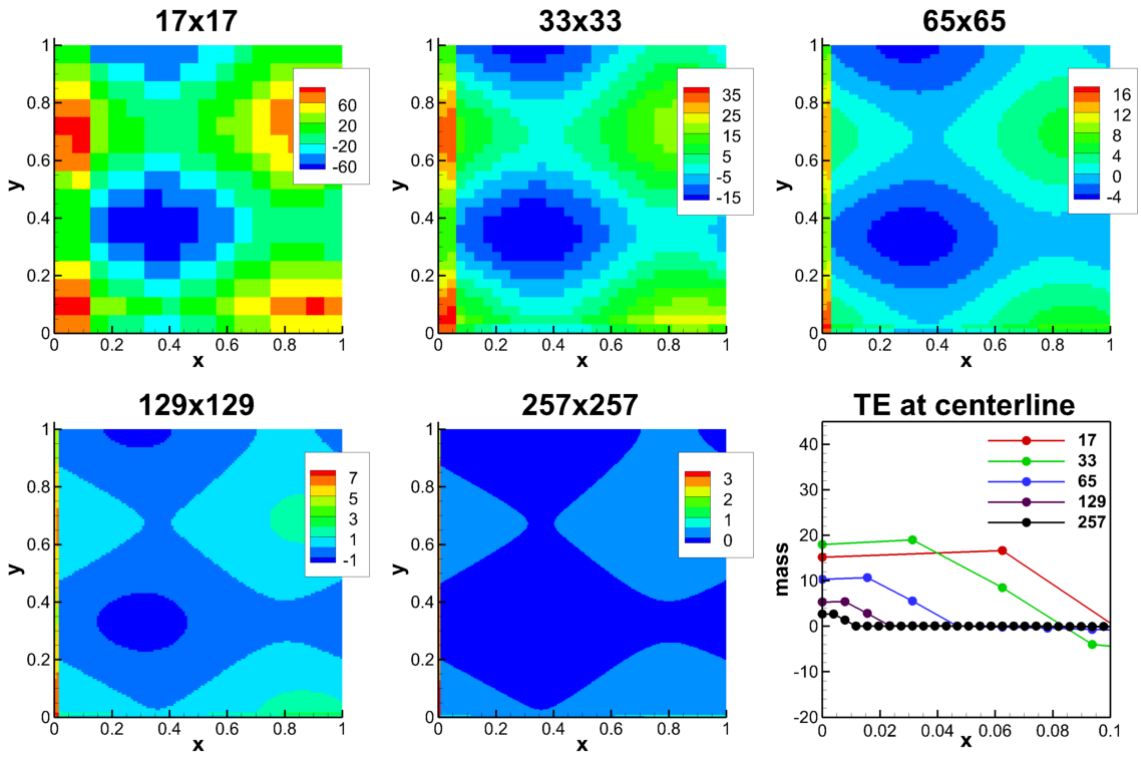
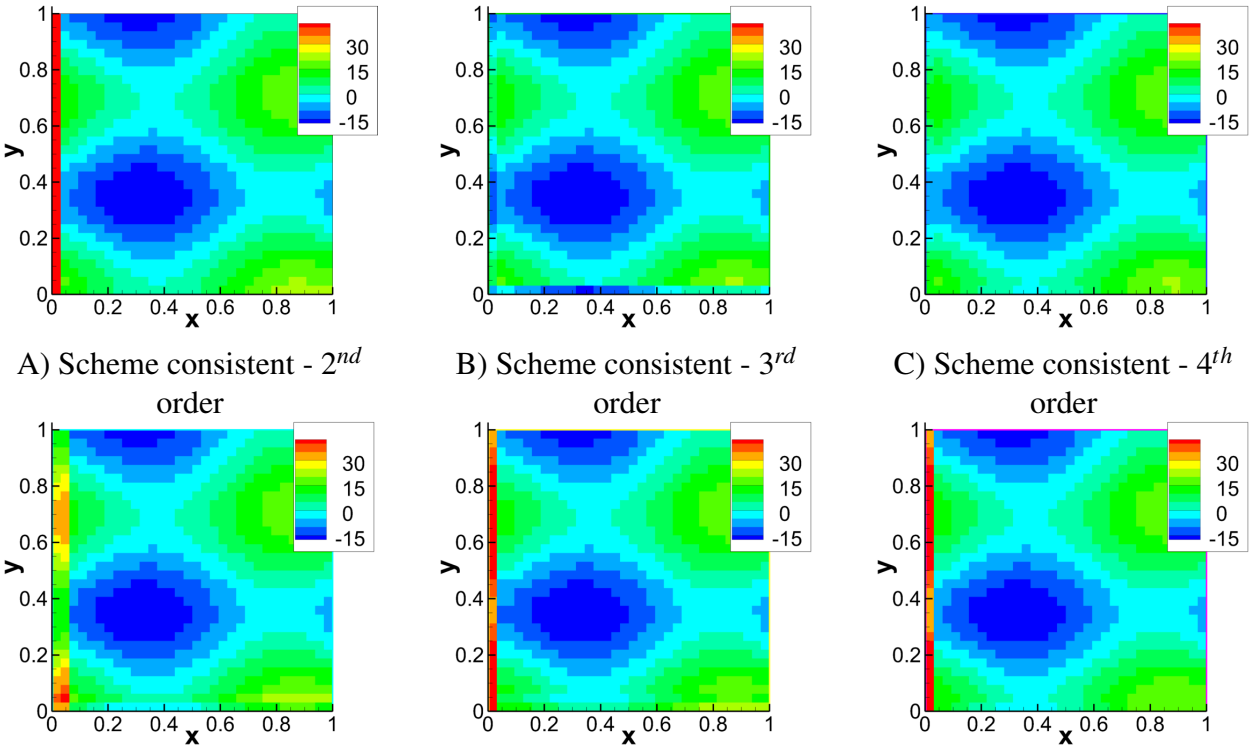


Figure 4.7: Supersonic manufactured solution truncation error in the conservation of mass equation on the series of Cartesian grids

Using the same solution and grid, the truncation error is compared for the boundary condition implementations with second-, third-, and fourth-order boundary condition extrapolations. The contour of the conservation of mass truncation error on the 33x33 Cartesian grid is shown in Figure 4.8, and a slice of the truncation error taken at $y = 0.5$ is shown in Figure 4.9. The trends in the truncation error are very similar to the trends shown in Burgers equation. The scheme inconsistent implementation of truncation has a significantly larger error near the boundaries which decreases at a rate lower than the rate of the interior discretization scheme. The scheme consistent second-order extrapolation has a first-order accurate truncation error; however, the third- and fourth-order scheme consistent boundary condition implementation truncation error reduces at a second-order rate and is indistinguishable from the interior truncation error. The scheme consistent third-order extrapolation has a larger boundary condition truncation error for the coarser grids compared to the scheme consistent fourth-order extrapolation. This is because the fourth-order solution is more asymptotic than the third-order solution, but after a few levels of grid refinement, the truncation error is dominated by the interior scheme. The scheme inconsistent boundary condition implementation truncation error reduces at a first-order rate for all orders of extrapolation and the truncation error actually increases for the scheme inconsistent third- and fourth-order extrapolations compared to the scheme inconsistent second-order extrapolation. The truncation error for the scheme inconsistent third- and fourth-order extrapolations are more asymptotic as the maximum trunca-

tion error reduces by a factor of two for each successive grid level where the scheme inconsistent second-order extrapolation truncation error is exhibiting oscillatory convergence for the 17x17, 33x33, and 65x65 grids. The reason for the observed behavior is due to lack of error cancelation for the cells near the boundary. For an interior cell the flux at the faces are set using MUSCL extrapolation. For a smoothly varying solution with small enough cells, the error in the flux at the faces due to MUSCL extrapolation is expected to be about the same order of magnitude and sign. This results in error cancelation when computing the net flux for a cell. For the scheme inconsistent boundary condition implementation, the boundary flux is set and the first interior face is set using MUSCL extrapolation. Because of the lack of error in the boundary flux, there is no error cancelation in the first interior cell which results in large truncation error near the boundary. The scheme consistent boundary condition implementation sets the boundary conditions in such a way that MUSCL extrapolation is used to set the boundary flux which adds the missing error and allows error cancelation and therefore a smooth truncation error. The truncation error is not smooth, however, for the scheme consistent second-order boundary condition. This is because of the addition of error due to the boundary condition extrapolation. For MUSCL extrapolation, the first two terms in the truncation error are second-order and third-order. If the order of accuracy of the extrapolation method matches or is greater than the second-term in the truncation error for MUSCL extrapolation (i.e., third-order), then the truncation error will be dominated by the second-order terms in MUSCL extrapolation. The results were the same for Burgers equation. Due to the central difference in the flux scheme, the first two terms in the truncation error are second-order and fourth-order. Only the fourth-order accurate boundary condition extrapolation resulted in second-order boundary condition truncation error. The results for the Euler supersonic Cartesian grid truncation error is similar for different combinations of equation set and grid topology.



A) Scheme consistent - 2^{nd} order

B) Scheme consistent - 3^{rd} order

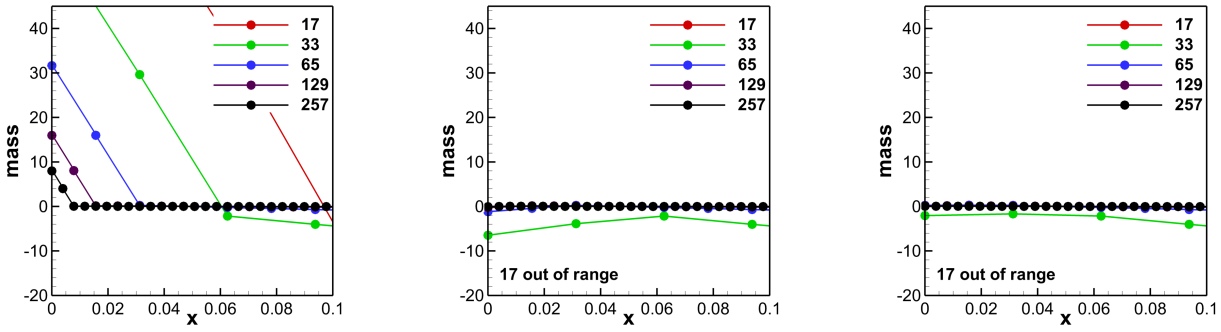
C) Scheme consistent - 4^{th} order

D) Scheme inconsistent - 2^{nd} order

E) Scheme inconsistent - 3^{rd} order

F) Scheme inconsistent - 4^{th} order

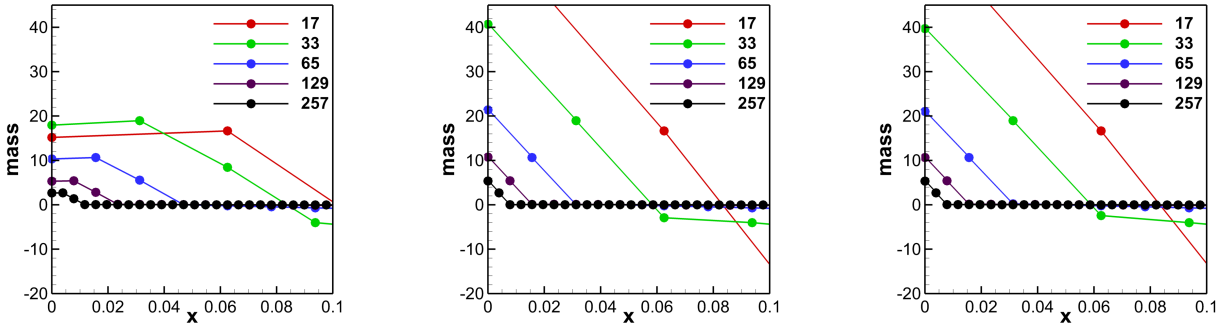
Figure 4.8: Supersonic manufactured solution truncation error contours for the conservation of mass equation on the 33x33 Cartesian grid



A) Scheme consistent - 2^{nd} order

B) Scheme consistent - 3^{rd} order

C) Scheme consistent - 4^{th} order



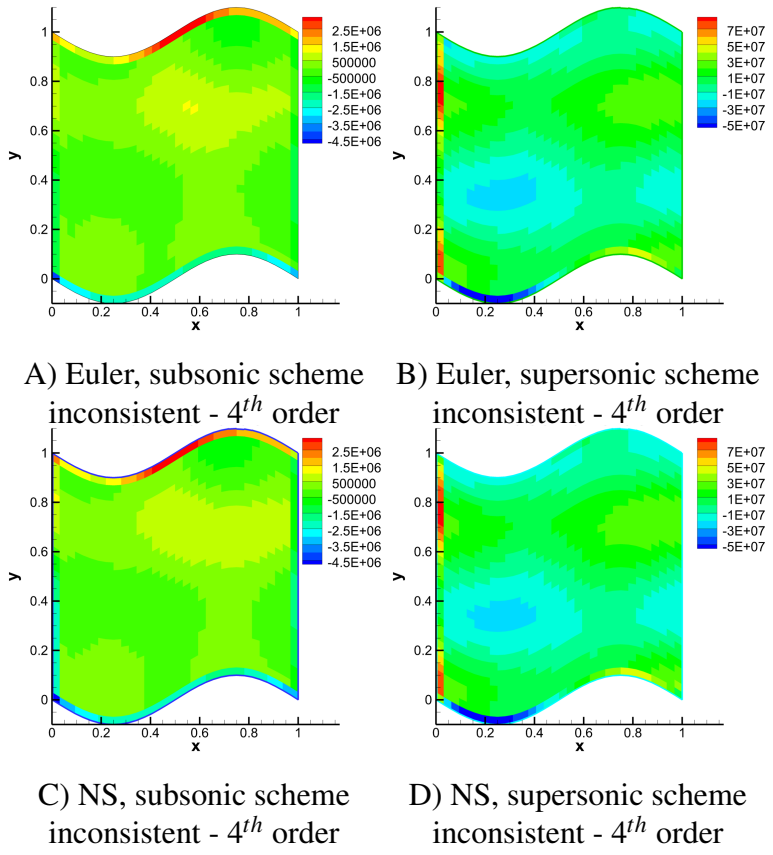
D) Scheme inconsistent - 2^{nd} order

E) Scheme inconsistent - 3^{rd} order

F) Scheme inconsistent - 4^{th} order

Figure 4.9: Supersonic manufactured solution truncation error slice at $y = 0.5$ in the conservation of mass equation on the series of Cartesian grids

Figure 4.10 compares the truncation error in the energy equation for the scheme inconsistent fourth-order boundary condition implementation on the curvilinear grid for the supersonic and subsonic manufactured solutions for the Euler and Navier-Stokes equations. The biggest difference in the truncation error between the supersonic and subsonic manufactured solutions is that a truncation error peak is present at the outflow (top and right boundary for the subsonic case). This is because the equations are now elliptic and the boundary condition implementation has a contribution to the truncation error in the domain. Comparing the Euler and Navier-Stokes equations, the truncation error differs slightly inside the domain; however, the truncation error peaks at the boundaries are present for both equations sets. The rate (first-order for all cases except for the scheme consistent fourth-order boundary condition which is second-order) that the boundary condition truncation error decreases is comparable for both the Euler and Navier-Stokes equations and for both the curvilinear and Cartesian grids.



A) Euler, subsonic scheme inconsistent - 4^{th} order

B) Euler, supersonic scheme inconsistent - 4^{th} order

C) NS, subsonic scheme inconsistent - 4^{th} order

D) NS, supersonic scheme inconsistent - 4^{th} order

Figure 4.10: Comparison between truncation error in the energy equation for the Euler and Navier-Stokes equations for both the subsonic and supersonic manufactured solutions on a 33x33 node grid

DE Analysis: Subsonic Euler Manufactured Solution

The fourth-order boundary condition implementations are chosen to compare truncation error and discretization error estimation accuracy for the subsonic Euler manufactured solution on the curvilinear grid with 33x33 nodes using fourth-order k-exact for the interior truncation error estimates. The estimated truncation error is compared to the exact truncation error in Figure 4.11. The interior truncation error is accurate for both boundary condition implementations as the interior is unaffected by the boundary condition implementation. The estimated truncation error for the scheme consistent implementation results in a peak near the lower left corner of the domain which is not present in the exact truncation error. The outflow truncation error is, however, accurately estimated. For the scheme inconsistent implementation, there is a significant peak in truncation error around all edges of the domain. The truncation error estimate for the scheme consistent boundary condition implementation is considerably more accurate than for the scheme inconsistent boundary condition implementation.

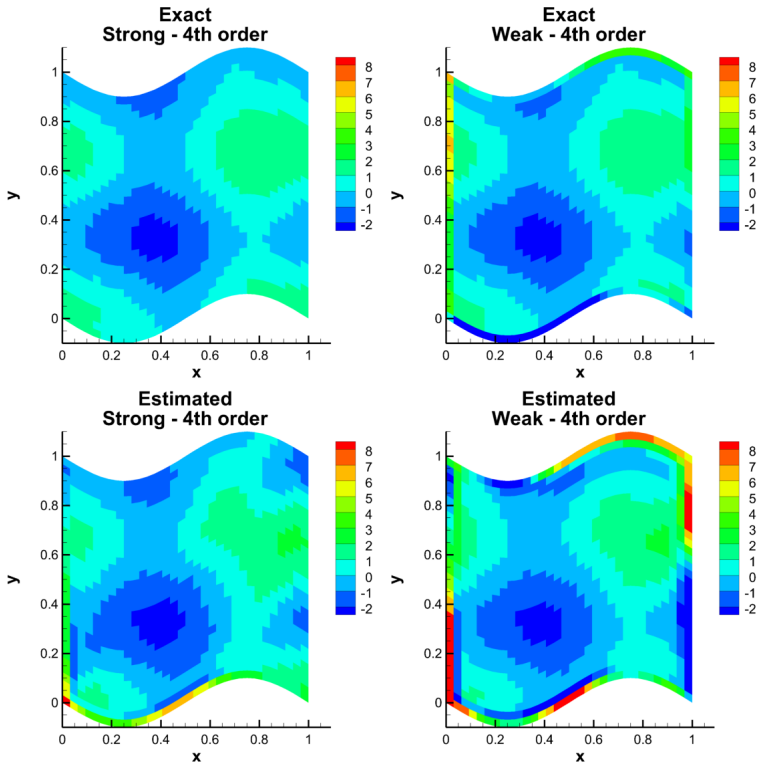


Figure 4.11: Estimated truncation error comparison for the Euler subsonic manufactured solution using the fourth-order boundary condition implementations on the 33x33 node grid

The truncation error estimates are used to solve the error transport equations to estimate the discretization error. Richardson extrapolation also requires the use of the 17x17 node grid in addition to the 33x33 node grid in order to estimate the discretization error. The 33x33 node solution is interpolated to the 17x17 node grid using volume-weighted average consistent with the finite volume method and adds no error to the solution. A slice at the center cell ($i = 16$) is used for further comparison of the discretization error. The discretization error is compared in Figure 4.12. From the contour plots, the discretization error estimates are comparable to the exact discretization error for the scheme consistent boundary condition implementation. Further comparison using the slice through the domain shows that the error transport equation almost exactly estimates the error and Richardson extrapolation compares very well. Quantitatively, the discretization error estimates compare well to the exact discretization error for the scheme inconsistent boundary condition implementation, but comparison using the slice through the domain shows under estimation of the peak discretization error for both the error transport equations and Richardson extrapolation. The error transport equation in this case again performs slightly better than Richardson extrapolation. From this case, the truncation error estimation and discretization error estimation are considerably more accurate for the scheme consistent boundary condition implementation and is consistent with what was also found for Burgers' equation.

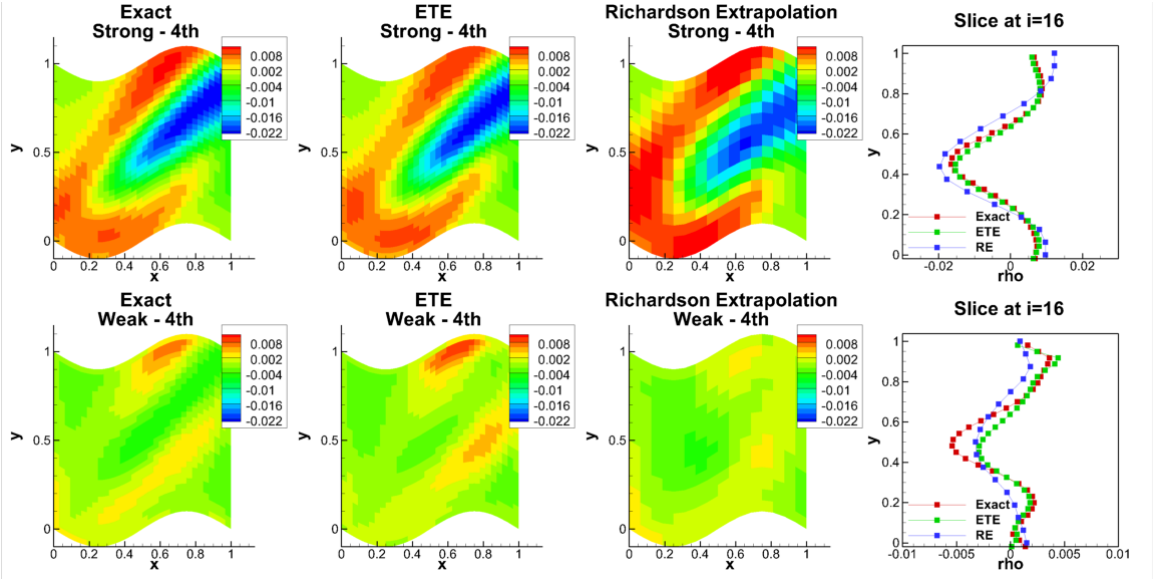


Figure 4.12: Discretization error estimate comparison for the Euler subsonic manufactured solution

DE Analysis: Supersonic Navier-Stokes Manufactured Solution

The fourth-order boundary condition implementations are again used to compare truncation error and discretization error estimation accuracy for the supersonic Navier-Stokes manufactured solution on the curvilinear grid with 33x33 nodes. The estimated truncation error is compared to the exact truncation in Figure 4.13. For this case, both boundary condition implementations accurately estimate their respective truncation errors. The scheme consistent boundary condition almost exactly reproduces the exact truncation error, while for the scheme inconsistent boundary condition, the contour plot is indistinguishable from the exact but underestimates the maximum truncation error by about 40 percent. Again, the truncation error estimate for the scheme consistent boundary condition implementation is considered more accurate than for the scheme inconsistent boundary condition implementation.

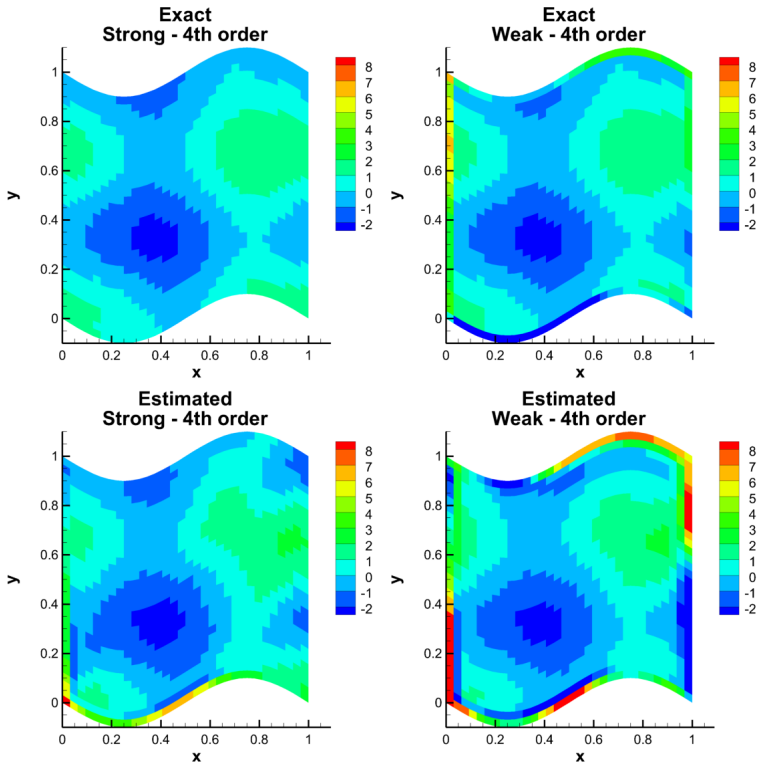


Figure 4.13: Truncation error estimated comparison for the Navier-Stokes supersonic manufactured solution using the fourth-order both boundary condition implementations

The truncation error estimates are used to solve the error transport equations to estimate the discretization error. Richardson extrapolation is used to estimate the truncation error using the 17x17 node grid in addition to the 33x33 node grid. The 33x33 node solution is interpolated to the 17x17 node grid using volume-weighted average consistent with the finite volume method and adds no error to the solution. A slice at the center cell ($i = 16$) is used for further comparison of the discretization error. The discretization error is compared in Figure 4.14. All discretization error estimates compare well for the scheme consistent boundary condition implementation. The largest discrepancy is in the error transport equation which underestimates the discretization near the top of the domain. The discretization error estimates for the scheme inconsistent boundary condition implementation compares well qualitatively, but the peak errors are slightly underestimated and mislocated. The error transport equation underestimates the discretization error near the top of the domain. Again, this case indicates that the truncation error and discretization error estimates are considerably more accurate for the scheme consistent boundary condition implementation and is consistent with what was also found for the subsonic Euler manufactured solution and Burgers' equations.

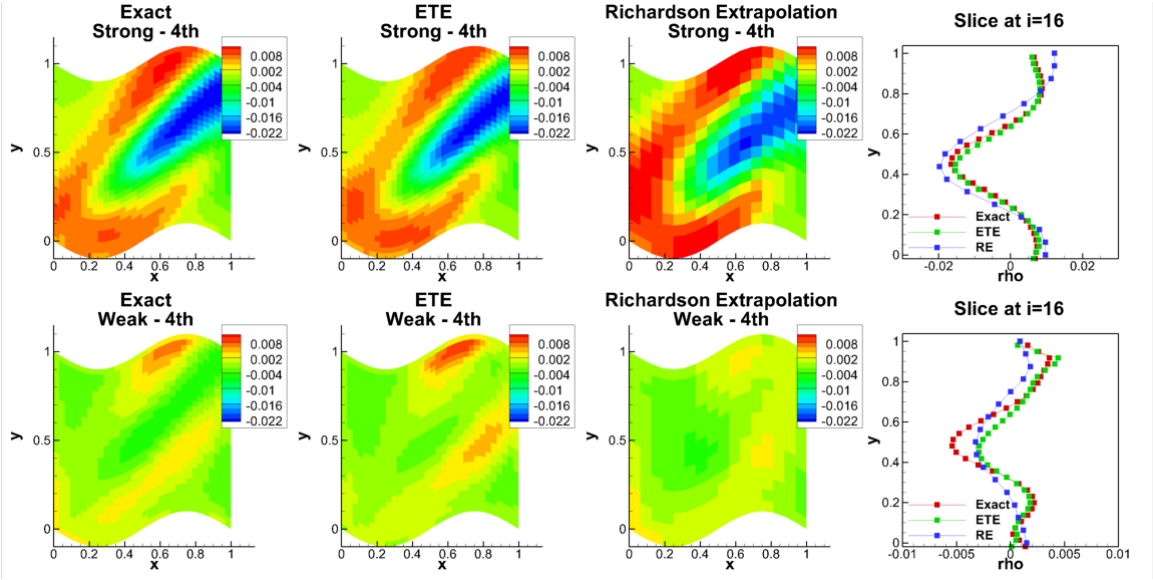


Figure 4.14: Discretization error estimate comparison for the Navier-Stokes supersonic manufactured solution

DE Analysis: Supersonic Vortex Flow

The truncation error is estimated for supersonic vortex flow using a polynomial order of two. The focus of the test case is the inner and outer wall truncation error estimation accuracy. The wall boundary condition is implemented by setting the flux at the boundary for a physically accurate implementation (i.e. $\vec{V} \cdot \hat{n} = 0$, a second ghost cell cannot be used as for the other boundary conditions. Figure 4.15 compares the exact and estimated truncation error with different boundary flux calculations. The most obvious descrepancy is on the inner wall. The truncation error in the first row of cells along the inner wall is on the same order of magnitude as the interior scheme. The second interior cell has a jump in truncation error because the first ghost cell is used to compute the flux at the first interior face. Using the interior flux at the boundary results in a poor estimate whereas the use of the wall flux at the boundary shows significant improvement.

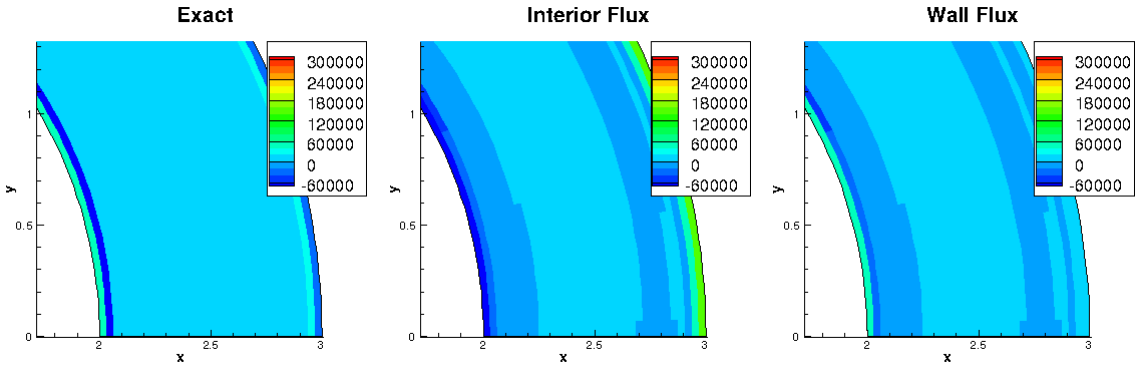
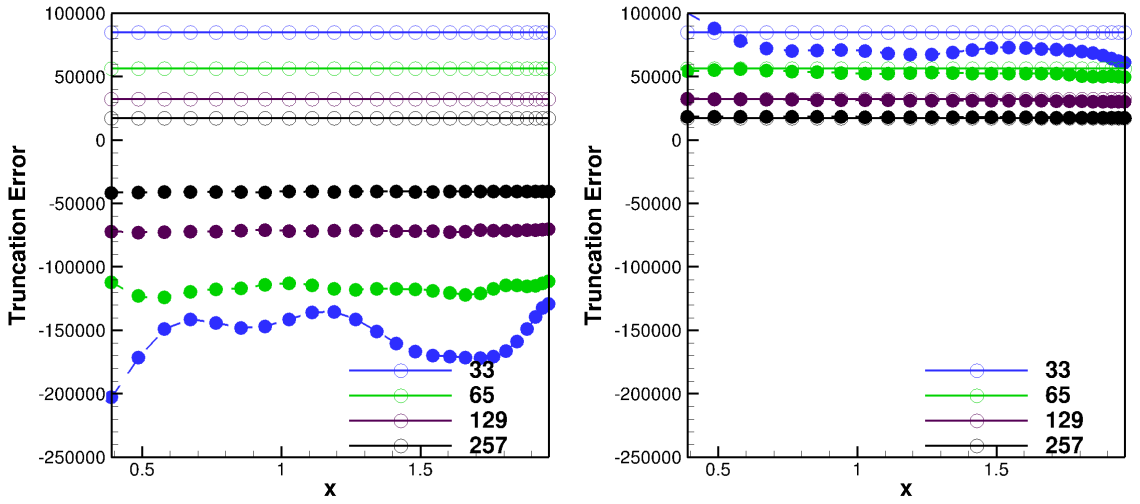


Figure 4.15: Supersonic vortex flow energy truncation error on the 65x33 node grid

The truncation error estimates along the inner wall are plotted for multiple grid levels shown in Figure 4.16. The interior flux at the boundary results in a very poor estimate of truncation error with the opposite sign as the exact truncation error. Using the correct wall flux is significantly more accurate, especially with grid refinement.



A) Interior flux B) Wall flux

Figure 4.16: Supersonic vortex flow energy truncation error on the inner wall (solid symbols are estimate, open symbols are exact)

The ETE discretization error estimate is computed using both truncation error estimates. The results are shown in Figure 4.17. The incorrect boundary flux results in a very poor estimate of discretization error, whereas the use of the wall flux results are considerably more accurate.

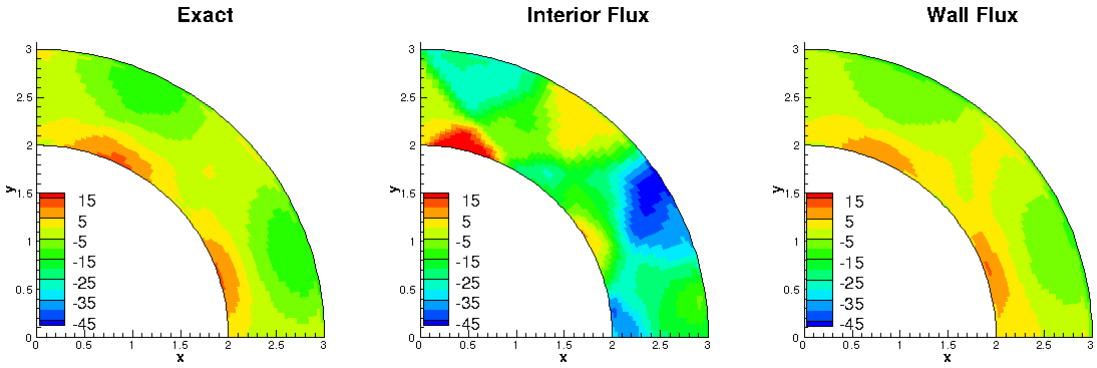


Figure 4.17: Supersonic vortex flow pressure ETE discretization error estimates on the 65x33 node grid

4.6 Conclusion

Burgers' equation was used to investigate different formulations for Neumann boundary conditions. The truncation error of each formulation was considered as well as the effect of the formulation on discretization error estimation. The truncation error for the scheme inconsistent formulation for both Dirichlet and Neumann boundary conditions was first-order accurate with significantly larger truncation errors compared to the scheme consistent formulation. The larger boundary condition truncation error affected both the exact discretization error and the estimated discretization error accuracy.

A similar process was applied to the Navier-Stokes and the Euler equations using a manufactured solution on a two-dimensional curvilinear grid. Scheme consistent and scheme inconsistent boundary condition implementations were developed for Dirichlet, inflow, and outflow boundary conditions with an extrapolation order of accuracy up to fourth-order. The intent was to reduce large truncation errors near the boundaries to improve smoothness for truncation error estimation (and therefore discretization error) and truncation error based mesh adaption. The boundary condition truncation error showed the same trends as Burgers' equation with the scheme inconsistent boundary condition implementation having a much larger truncation error near the boundaries with a lower order of accuracy and the scheme consistent form showing smooth truncation error near the boundaries for the third- and fourth-order accurate boundary condition extrapolation. The order of accuracy of the boundary condition extrapolation should be at least larger than the leading term in the truncation error of the interior discretization scheme in order of the boundary condition truncation error to be insignificant for asymptotic solutions. If this is the case, then the truncation error in the extrapolation will decrease faster than the leading truncation error term for the interior scheme and will result in an insignificant contribution to the truncation error for asymptotic solutions and therefore a smooth truncation error near the boundary. (It is important to note that for non-smooth grids near the boundary higher order extrapolations can have difficulty converging.) For all cases

studied, the truncation error estimation accuracy was much better for the scheme consistent boundary condition implementation. This was also the case for the discretization error accuracy for both Richardson extrapolation and the error transport equations. The scheme consistent boundary condition implementation allows for the boundary flux to be set using the interior flux scheme and, therefore, allow the for error cancelation resulting in smooth truncation error near the boundary as long as the extrapolation order of accuracy requirement is met.

Truncation error estimation for a slip wall boundary condition was also investigated. The wall boundary condition is the most physically accurate by directly setting the flux at the boundary to zero with the exception of the pressure flux which is extrapolated to the face from the interior. The result was a considerable increase in truncation error estimation and discretization error estimation accuracy.

Bibliography

[1] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 210, pp. pp. 307–357, 1911.

[2] P. J. Roache and P. M. Knupp, "Completed richardson extrapolation," *Communications in Numerical Methods in Engineering*, vol. 9, no. 5, pp. 365–374, 1993.

[3] S. A. Richards, "Completed richardson extrapolation in space and time," *Communications in Numerical Methods in Engineering*, vol. 13, pp. 573–582, 1997.

[4] E. Fehlberg, "Low-order classical runge-kutta formulas with step size control and their application to some heat transfer problems," *NACA Technical Report 315*, 1969.

[5] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 337–357, 1987.

[6] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates, part 2: Error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1365–1382, 1992.

[7] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[8] H. J. Stetter, "The defect correction principle and discretization methods," *Numerische Mathematik*, vol. 29, pp. 425–443, 1978.

[9] R. D. Skeel, "Thirteen ways to estimate global error," *Numerische Mathematik*, vol. 48, pp. 1–20, 1986.

[10] A. Naumovich, M. Foerster, and R. Dwight, "Algebraic multigrid within defect correction for the linearized euler equations," *Numerical Linear Algebra with Applications*, vol. 17, pp. 307–324, 2009.

[11] X. D. Zhang, D. Pelletier, and J.-Y. Trepanier, "Verification of error estimators for the euler equations," Tech. Rep. AIAA-2000-1001, 2000.

[12] Y. Qin and T. I.-P. Shih, "A discrete transport equation for error estimation in cfd," Tech. Rep. AIAA-2002-0906, 2002.

[13] I. Celik and G. Hu, "Single grid error estimation using error transport equations," *J of Fluid Eng*, vol. 126, pp. 778–790, 2004.

[14] Y. Qin, K. Chi, and T. I.-P. Shih, "Modeling the residual in error-transport equations for estimating grid-induced errors in cfd solutions," Tech. Rep. AIAA-2006-892, 2006.

[15] T. I.-P. Shih and B. R. Williams, "Development and evaluation of an a posteriori method for estimating and correcting grid-indeuced errors in solutions of the navier-stokes equations," Tech. Rep. AIAA-2009-1499, 2009.

[16] P. Cavallo and N. Sinha, "An error transport equation with practical applications," Tech. Rep. AIAA-2007-4092, 2007.

[17] P. Cavallo, N. Sinha, and M. R. O'Gara, "Viscous error transport equation for error quantification of turbulent flows," *AIAA-2008-3851*, 2008.

[18] I. Babuska and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal of Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[19] I. Babuska and W. C. Rheinboldt, "A-posteriori error estimates for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[20] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, New York, 2000.

[21] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[22] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from pde approximations," *SIAM Review*, vol. 42, no. 2, pp. 247–264, 2000.

[23] T. S. Phillips and C. J. Roy, "A new extrapolation-based uncertainty estimator for computational fluid dynamics," *AIAA-2013-0260*, 2013.

[24] T. S. Phillps and C. J. Roy, "Residual methods for discretization error estimation," *AIAA-2011-3870*, 2011.

[25] J. W. Banks, T. Aslam, and W. J. Rider, "On sub-linear convergence for linearly degenerate waves in capturing schemes," *Journal of Computational Physics*, 2008.

[26] C. J. Roy, "Review of discretization error estimators in scientific computing," AIAA Paper 2010-126, 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 4-7, 2010, 2010.

[27] C. J. Roy, "Strategies for driving mesh adaptation in cfd (invited)," Tech. Rep. AIAA-2009-1302, 2009.

[28] A. Choudhary and C. J. Roy, "Structured mesh r-refinement using truncation error equidistribution for 1d and 2d euler problems," *AIAA-2013-2444*, 2013.

[29] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[30] C. W. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, pp. 1–15, 1960.

[31] T. J. Barth, "Higher order solution of the euler equations on unstructured grids using quadratic reconstruction," Tech. Rep. AIAA-90-0013, 1990. Initial development of K-exact method.

[32] T. J. Barth, "Recent developments in higher order k-exact reconstruction on unstructured meshes," Tech. Rep. AIAA-93-0668, 1993. Improvements to the K-exact method with work towards unstructured grids and comparison to other competing technologies.

[33] P. J. Roache, *Fundamentals of Verification and Validation*. Hermosa Publishers, Albuquerque, NM., 2009.

[34] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[35] J. M. Derlaga, T. S. Phillips, and C. J. Roy, "Sensei computational fluid dynamics code: A case study in modern fortran software development," Tech. Rep. AIAA-2013-2450, 2013.

[36] C. Ollivier-Gooch, A. Nejat, and K. Michalak, "Obtaining and verifying higher-order unstructured finite volume solutions to the euler equations," *AIAA Journal*, vol. 47, no. 9, pp. 2105–2120, 2009.

Appendix B

Table A.1: Summary of finite-volume extrapolations for scheme inconsistent boundary conditions through the boundary face

Order of Accuracy	Extrapolation
1^{st}	$f_0(u_{1/2}) = u_{1/2}$
2^{nd}	$f_0(u_{1/2}, u_1) = 2u_{1/2} - u_1$
3^{rd}	$f_0(u_{1/2}, u_1, u_2) = 3u_{1/2} - \frac{5}{2}u_1 + \frac{1}{2}u_2$
4^{th}	$f_0(u_{1/2}, u_1, u_2, u_3) = 4u_{1/2} - \frac{13}{3}u_1 + \frac{5}{3}u_2 - \frac{1}{3}u_3$

Table A.2: Summary of finite-volume extrapolations to the first ghost cell

Order of Accuracy	Extrapolation
1^{st}	$g_0(u_1) = u_{1/2}$
2^{nd}	$g_0(u_1, u_2) = 2u_1 - u_2$
3^{rd}	$g_0(u_1, u_2, u_3) = 3u_1 - 3u_2 + u_3$
4^{th}	$g_0(u_1, u_2, u_3, u_4) = 4u_1 - 6u_2 + 4u_3 - u_4$

Table A.3: Summary of finite-volume extrapolations to the boundary face

Order of Accuracy	Extrapolation
1^{st}	$h_{1/2}(u_1) = u_1$
2^{nd}	$h_{1/2}(u_1, u_2) = \frac{3}{2}u_1 - \frac{1}{2}u_1$
3^{rd}	$h_{1/2}(u_1, u_2, u_3) = \frac{11}{6}u_1 - \frac{7}{6}u_2 + \frac{1}{3}u_3$
4^{th}	$h_{1/2}(u_1, u_2, u_3, u_4) = \frac{25}{12}u_1 - \frac{23}{12}u_2 + \frac{13}{12}u_3 - \frac{1}{4}u_4$

Table A.4: Summary of scheme inconsistent boundary conditions where n is the order of accuracy

	Variable	$u_{1/2}$	u_0	u_{-1}
Dirichlet	ρ	$\rho(x_{1/2})$	$f_0(\rho_{1/2},\rho_1,\ldots,\rho_{n-1})$	-
	$\vec{V}$	$\vec{V}(x_{1/2})$	$f_0(\vec{V}_{1/2},\vec{V}_1,\ldots,\vec{V}_{n-1})$	-
	P	$P(x_{1/2})$	$f_0(P_{1/2},P_1,\ldots,P_{n-1})$	-
Subsonic Inflow	ρ	$\rho(x_{1/2})$	$f_0(\rho_{1/2},\rho_1,\ldots,\rho_{n-1})$	-
	$\vec{V}$	$\vec{V}(x_{1/2})$	$f_0(\vec{V}_{1/2},\vec{V}_1,\ldots,\vec{V}_{n-1})$	-
	P	$h_{1/2}(P_1,\ldots,P_n)$	$g_0(P_1,\ldots,P_n)$	-
Supersonic Inflow	ρ	$\rho(x_{1/2})$	$\rho(x_{1/2})$	-
	$\vec{V}$	$\vec{V}(x_{1/2})$	$\vec{V}(x_{1/2})$	-
	P	$P(x_{1/2})$	$P(x_{1/2})$	-
Subsonic outflow	ρ	$h_{1/2}(\rho_1,\ldots,\rho_n)$	$g_0(\rho_1,\ldots,\rho_n)$	-
	$\vec{V}$	$h_{1/2}(\vec{V}_1,\ldots,\vec{V}_n)$	$g_0(\vec{V}_1,\ldots,\vec{V}_n)$	-
	P	$P(x_{1/2})$	$f_0(P_{1/2},P_1,\ldots,P_{n-1})$	-
Supersonic outflow	ρ	$h_{1/2}(\rho_1,\ldots,\rho_n)$	$g_0(\rho_1,\ldots,\rho_n)$	-
	$\vec{V}$	$h_{1/2}(\vec{V}_1,\ldots,\vec{V}_n)$	$g_0(\vec{V}_1,\ldots,\vec{V}_n)$	-
	P	$h_{1/2}(P_1,\ldots,P_n)$	$g_0(P_1,\ldots,P_n)$	-
Slip Wall	ρ	-	$g_0(\rho_1,\ldots,\rho_n)$	-
	$\vec{V}$	$\vec{V}\cdot\hat{n}=0$	$g_0(\vec{V}_1,\ldots,\vec{V}_n)$	-
	P	$h_{1/2}(P_1,\ldots,P_n)$	$g_0(P_1,\ldots,P_n)$	-

Table A.5: Summary of scheme consistent boundary conditions where n is the order of accuracy

	Variable	$u_{1/2}$	u_0	u_{-1}
Dirichlet	ρ	-	$\rho(x_0)$	$g_0(\rho_0,\ldots,\rho_{n-1})$
	$\vec{V}$	-	$\vec{V}(x_0)$	$g_0(\vec{V}_0,\ldots,\vec{V}_{n-1})$
	P	-	$P(x_0)$	$g_0(P_0,\ldots,P_{n-1})$
Subsonic Inflow	ρ	-	$\rho(x_0)$	$g_0(\rho_0,\ldots,\rho_{n-1})$
	$\vec{V}$	-	$\vec{V}(x_0)$	$g_0(\vec{V}_0,\ldots,\vec{V}_{n-1})$
	P	-	$g_0(P_1,\ldots,P_n)$	$g_0(P_0,\ldots,P_n)$
Supersonic Inflow	ρ	-	$\rho(x_0)$	$\rho(x_0)$
	$\vec{V}$	-	$\vec{V}(x_0)$	$\vec{V}(x_0)$
	P	-	$P(x_0)$	$P(x_0)$
Subsonic outflow	ρ	-	$g_0(\rho_1,\ldots,\rho_n)$	$g_0(\rho_0,\ldots,\rho_{n-1})$
	$\vec{V}$	-	$g_0(\vec{V}_1,\ldots,\vec{V}_n)$	$g_0(\vec{V}_0,\ldots,\vec{V}_{n-1})$
	P	-	$P(x_0)$	$g_0(P_0,\ldots,P_{n-1})$
Supersonic outflow	ρ	-	$g_0(\rho_1,\ldots,\rho_n)$	$g_0(\rho_0,\ldots,\rho_{n-1})$
	$\vec{V}$	-	$g_0(\vec{V}_1,\ldots,\vec{V}_n)$	$g_0(\vec{V}_0,\ldots,\vec{V}_{n-1})$
	P	-	$g_0(P_1,\ldots,P_n)$	$g_0(P_0,\ldots,P_{n-1})$

Chapter 5

Numerical Benchmark Solutions

Residual-based Discretization Error Estimators for CFD Part 1: Numerical Benchmarks

Tyrone S. Phillips, Joseph M. Derlaga, Christopher J. Roy,
and Jeff Borggaard
Virginia Tech, Blacksburg, Virginia, 24061

Discretization error is usually the largest and most difficult numerical error to estimate in a Computational Fluid Dynamics (CFD) simulation. Development and evaluation of discretization error estimators ideally requires exact solutions to the PDEs because the exact discretization error can be computed. Exact solutions, however, are only available for relatively simple cases. An alternative is to use numerical benchmarks which require a verified code and numerical solutions on a mesh which is sufficiently fine so that the discretization error in the benchmark solution is (essentially) negligible. The focus of part one of this work is to develop several numerical benchmark solutions to evaluate residual-based discretization error estimation methods which include defect correction and error transport equations for the Euler equations. This class of discretization error estimation methods requires accurate estimation of a residual/truncation error which will also be evaluated because the residual is the primary factor affecting the accuracy of the discretization error estimate, (i.e., an exact residual results in an exact discretization error estimate for defect correction). The benchmark solutions include a supersonic inlet, a blunted hypersonic cone, and an inviscid NACA 0012 airfoil at Mach numbers of 0.5 and 0.8.

5.1 Introduction

Computational Fluid Dynamics (CFD) has enormous potential to impact the analysis, design, and optimization of engineering systems. The predictive capability of CFD depends not only on the validity of the sub-models employed and the uncertainties present in the system and surroundings, but also on the ability to accurately and reliably estimate and reduce numerical errors. While there are many sources of numerical error in CFD simulations, the largest and most difficult to estimate is the error related to the resolution of the spatial grid, i.e., the spatial discretization error.

The discretization error is defined as the difference between the exact solution to the discrete equations on a grid with cell size metric h, u_h, and the exact solution to the Partial Differential Equations (PDEs), $\tilde{u}$:

$$\varepsilon_h = u_h - \tilde{u} \tag{5.1}$$

For solutions on sufficiently fine grids, the discretization error will reduce with grid refinement.

The source of discretization error are the approximations used to discretized the flow domain and the PDEs is called truncation error. As the grid is refined, higher-order terms in the truncation error decrease at a faster rate than the leading term. The rate that truncation error decreases with grid refinement is proportional to the grid size parameter raised to the power of the formal order of accuracy of the numerical scheme, h^p. The grid size parameter is a measure of the discretization size in each coordinate direction (e.g., Δx) and the formal order of accuracy is the rate that the the leading truncation error term decreases with respect to mesh spacing. Since truncation error can be shown to serve as the local source for the discretization error, the discretization error is expected to decrease at the same rate.

Several different classifications of discretization error estimators are available and each has different advantages and disadvantages. Higher-order methods estimate the discretization error by comparing the discrete solution to a higher-order estimate of the exact solution found by post-processing the numerical solutions. This category includes Richardson extrapolation,[1,2,3] order refinement,[4] and finite element recovery methods.[5,6] The most widely used discretization error estimator is Richardson extrapolation which uses a series of solutions with increasingly finer grid resolution to construct a higher order accurate solution. Richardson extrapolation is simple to apply as a post processing step and does not require code modification. The disadvantage of Richardson extrapolation is the requirement of multiple grid levels, since grid generation is one of the most time consuming tasks for a numerical simulation. In addition to the expense of generating multiple grid levels, each grid level used in Richardson extrapolation must also be in the asymptotic range for accurate discretization error estimation.

Residual-based methods use a discrete solution combined with information about the PDE or discrete equations being solved to estimate the discretization error. They generally require two solutions to be computed on the same grid. This category includes defect correction methods[7,8,9,10], Error Transport Equations (ETEs)[11,12,12,13,14,15,16,17], finite element residual methods[18,19], and adjoint methods[20,21]. Residual-based methods require only one grid level which reduces the grid generation requirements. These methods have the potential to be more accurate than Richardson extrapolation since they only require a single grid in the asymptotic range and they bring in additional information about the equations being solved. If the exact truncation error is used then the exact solution to the PDEs can potentially be recovered. Accurate truncation error estimation is a key element for an accurate discretization error estimate.

The residual-based methods of interest herein are defect correction and ETEs. Defect correction corrects the original numerical solution by re-solving the original problem with the truncation error estimate added to the right-hand side. The resulting solution is an estimate of the exact solution and the difference between the original solution and the defect correction solution is an estimate of the discretization error. Another perspective on defect correction is that it essentially cancels out the leading truncation error terms to give you a higher-order accurate solution which can be used to estimate the discretization error. The ETEs work in a similar manner except the ETEs solve directly for the discretization error by linearizing the original governing equations about a solution so that the discretization error is directly computed. ETEs have been applied to several simple model problems such as the advection/diffusion equation, Burgers' equation, wave

equation, etc.[12,13,14] Error transport equations were developed for laminar viscous flow,[15] and have been solved on much more complex problems including an airfoil, an aircraft engine exhaust, and a rocket plume.[16,17]

Previous work evaluated the discretization error estimates using a solution computed on a finer grid than the computational grid (generally refered to as a numerical benchmark).[12,12,13,14,15,16,17] The method used to estimate the residual (i.e. truncation error) used a multi-grid approach which is conducive to evaluating the discretization error estimates due to error cancelation. Error cancelation does not occur for single grid truncation error estimation methods which have exhibited several advantages over multi-grid truncation error estimation methods (see Phillips and Roy[22]). The purpose of this work is for several numerical benchmarks with carefully characterized numerical errors to evaluate a number of truncation error estimation methods and the resulting discretization error estimates. Phillips et al.[23] outlined requirements to develop a numerical benchmark, which include characterization of the discretization error in the numerical benchmark. The benchmark discretization error is then used to add appropriate error bars to the resulting truncation error and discretization error evaluations.

5.2 Numerical Benchmark Solutions

In order for a numerical solution to qualify as a benchmark solution, the problem statement, numerical scheme, and numerical solution accuracy should be documented.[24,25] Quantifying the numerical accuracy of benchmark solutions is often difficult, and at a minimum should include evidence that (1) the asymptotic convergence range has been achieved for the benchmark problem and (2) the code used to generate the benchmark solution has passed the order of accuracy code verification test for all of the options exercised in the benchmark problem.

In order to establish that the benchmark solution is in the asymptotic convergence range, iterative and round-off errors must be negligible compared to the discretization error in the benchmark solution. Following the rule-of-thumb given by Roy,[26] the magnitude of iterative and round-off errors should be at least two orders of magnitude smaller than the magnitude of the discretization error. This places further requirements on iterative convergence and round-off error in the benchmark solution (see Oberkampf and Roy[24] for mor information). Phillips et al.[23] developed a numerical benchmark for a turbulent flat plate. It was pointed out that a farfield boundary had a significant effect on the results when comparing a structured solver solution which has less numerical dissipation to an unstructured solver soution with more numerical dissipation. The farfield numerical error decreased at a first-order rate and is should be quantified when a farfield boundary condition is used in addition to iterative convergence error and round-off error.

If a numerical benchmark is to be useful, it should have numerical errors that are significantly smaller than the numerical errors in the other solutions that one is trying to evaluate. From a practical standpoint, this usually means that the benchmark numerical solution must be computed on an extremely fine mesh, with double precision computations, and with a low iterative convergence tol-

erance (i.e., with small iterative errors). The discretization error in the benchmark solution should be accurately estimated and reported along with a measure of confidence in the estimate. The discretization error in the benchmark solution is important to estimate the effects of the benchmark numerical error on the error evaluation calculations.

5.3 Numerical Error

Important source of numerical error present in the numerical benchmark are primarily discretization error and iterative error. For some applications, the numerical error due to the presence of the boundary farfield boundary can be a significant source of error as well. Norms are used to compare the magnitudes of each numerical error. The norms use are the discrte L_1 norm defined as

$$||\varepsilon||_{L_1} = \sum_i^N \frac{|\varepsilon_i|}{N} \tag{5.2}$$

and the discrete L_2 norm is defined as

$$||\varepsilon||_{L_2} = \sqrt{\sum_i^N \frac{\varepsilon_i^2}{N}}. \tag{5.3}$$

where $epsilon_i$ is the the estimated/exact error averaged over a cell for a finite volume method.

The iterative error and boundary condition error (if present) should be at least two orders lower than the discretization error in the numerical benchmark to be considered negligle. The effects of the numerical benchmark error are computed by propagating the discretization error through a given function. For example, consider the computation of the exact truncation error which is typically computed by inserting the exact solution into the discrete equations $L_h(\tilde{u}) = \tau_h \tilde{u}$. The exact truncation error is instead computed by inserting the numerical benchmark into the discrete equation $L_h(u_{NB}) \approx \tau \tilde{u}$. The percent error due to the presence of the discretization error is estimated by

$$\varepsilon_\tau = \frac{|L_h(u_{NB} - \bar{\varepsilon}_{NB}) - L_h(u_{NB})|}{L_h(u_{NB})} \times 100 \tag{5.4}$$

where ε_{NB} is the estimated discretization error in the numerical benchmark and $\tilde{u} \approx u_{NB} - \bar{\varepsilon}_{NB}$ is the estimate of the exact solution. The quantities of interest is the estimated truncation error and the estimated discretization error for a given solution computed on a much coarser grid than the numerical benchmark.

5.3.1 Discretization Error

Discretization error is often the largest and most difficult numerical error to estimate and is defined in Equation 5.1. The discretization error in the numerical benchmark is estimated using Richardson extrapolation.[1] Richardson extrapolation is the most commonly used discretization error estimate and is accurate for asymptotic solutions. The numerical benchmark and a solution computed on a grid coarsed by a factor of two (i.e. removing every other point from a structured mesh) are used to estimate the discretization error

$$\varepsilon_{de} = \frac{u_{rh} - u_h}{r^{p_f} - 1} \tag{5.5}$$

where u_h is the numerical benchmark and u_{rh} is the solution on the coarser grid coarsened by a factor $r = 2$.

5.3.2 Iterative Error

Iterative error a numerical error which is present when an iterative scheme is used to solve the discree equations. The iterative error is defined as

$$\varepsilon_{ie} = u_h^k - u_h \tag{5.6}$$

where u_h^k is the numerical solution at iteration k and u_h is the exact solution to the discrete equations. The iterative error is computed for each case on a coarser grid level then the benchmark grid level. The iterative error is then coorelated to the iterative residuals to determine the required level of convergence so that $\varepsilon_{ie} << \varepsilon_{de}$ for the benchmark solution. Figure 5.1 illustrates the iterative error and the iterative residuals. The estimated discretization error in the numerical benchmark is included to determine the convergence tolerance required for the given case. For more information regarding iterative error estimation see Oberkampf and Roy[24].

5.3.3 Boundary Condition Error

The farfield boundary condition can be a significant source of error in the numerical solution.[23] The farfield boundary condition is implemented only for the airfoil cases. The farfield boundary condition implemented in SENSEI decreases at a first-order rate relative to distance from the airfoil. To determine the boundary condition error, a coarse computational grid is modified by extracting sections of the grid to reduce the farfield distance. Richardson extrapolation can also be used to estimate the boundary condition error as defined in Equation 5.5 where p_f = 1. Figure 5.2 shows the modified grids used to estimate the boundary condition error and Figure 5.3 shows the estimated boundary condition error and the order of accuracy where h is the relative farfield distance where $h = 1$ is the farthest distance.

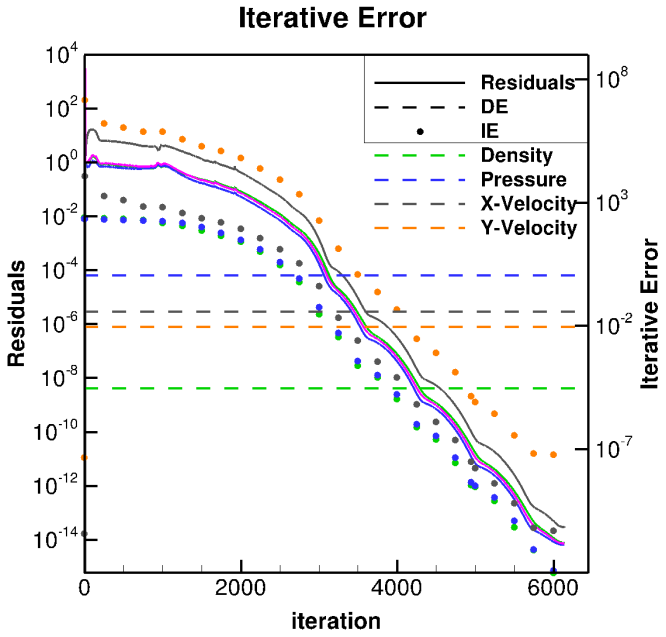


Figure 5.1: Iterative error and iterative residuals

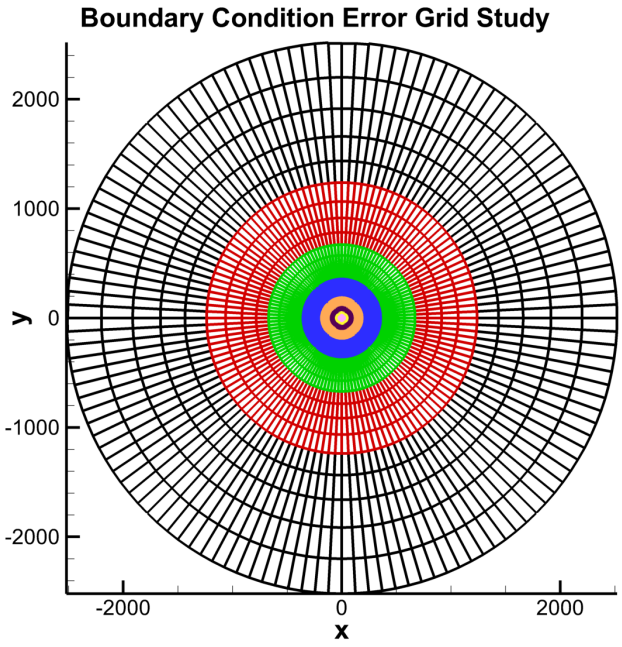


Figure 5.2: Grid family created to estimate farfield boundary condition error

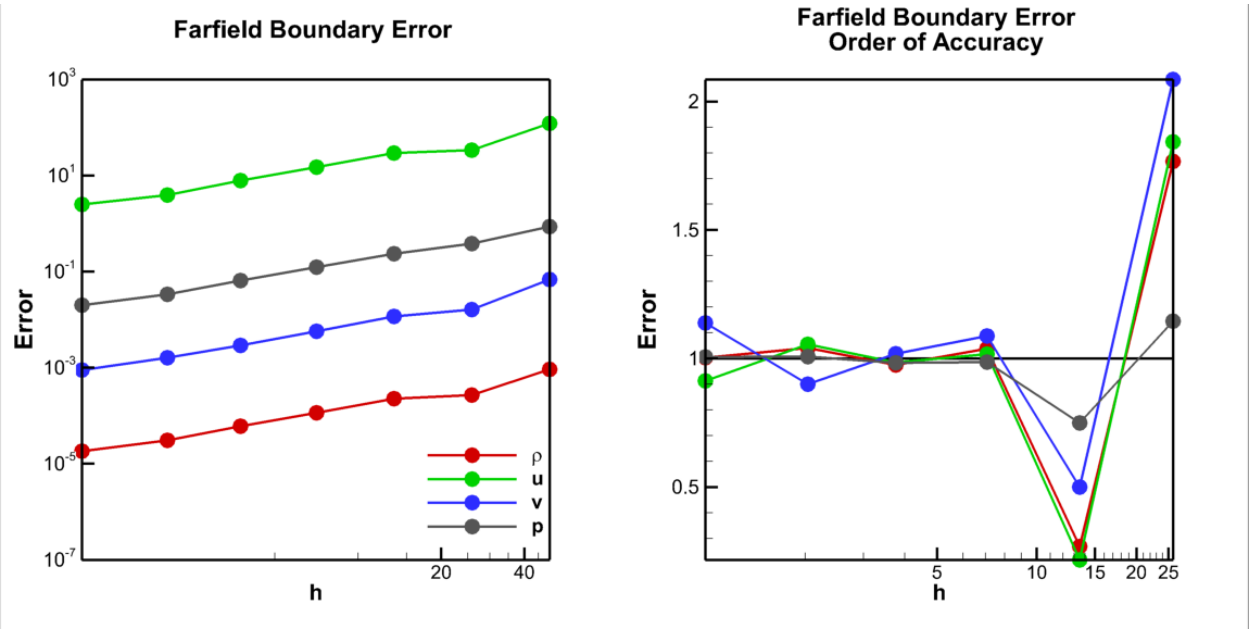


Figure 5.3: Example farfield boundary error and observed order of accuracy

5.4 Governing Equations

5.4.1 Navier-Stokes

The finite volume formulation of the Navier-Stokes equations is

$$L(\vec{U}) = \frac{\partial}{\partial t}\iiint_V \vec{U}\,dV + \iint_A \left(\vec{F}_i - \vec{F}_v\right) dA - \iiint_V \vec{S}\,dV = 0 \tag{5.7}$$

where the inviscid flux is

$$\vec{F}_i = \begin{bmatrix} \rho\vec{v}\cdot\hat{n} \\ \rho u\vec{v}\cdot\hat{n} + n_x p \\ \rho v\vec{v}\cdot\hat{n} + n_y p \\ \rho w\vec{v}\cdot\hat{n} + n_z p \\ \rho h_t\vec{v}\cdot\hat{n} \end{bmatrix} \tag{5.8}$$

the viscous flux is

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\theta_x + n_y\theta_y + n_z\theta_z \end{bmatrix} \tag{5.9}$$

The outward pointing face normal at side of a grid cell is $\hat{n} = [n_x, n_y, n_z]$, the conserved variable vector is $\vec{U} = [\rho, \rho u, \rho v, \rho w, \rho e_t]$, the velocity vector is $\vec{V} = [u, v, w]$, the total energy e_t and total enthalpy h_t are computed using the calorically perfect gas assumption with an equation of state $p = \rho RT$, $e_t = \frac{p}{\rho(\gamma-1)}$, and $h_t = \frac{p\gamma}{\rho(\gamma-1)} + \frac{|\vec{V}|^2}{2}$. The diffusive components of the viscous flux are

$$\tau_{xx} = 2\mu\left(\frac{\partial u}{\partial x} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{yy} = 2\mu\left(\frac{\partial u}{\partial y} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{zz} = 2\mu\left(\frac{\partial u}{\partial z} - \frac{1}{3}\nabla\cdot\vec{V}\right),$$
$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right), \quad \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right),$$
$$\theta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x}, \quad \theta_y = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y}, \quad \theta_z = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z}$$

and the heat conduction coefficient for a calorically perfect gas is $k = \frac{R\gamma\mu}{Pr(\gamma-1)}$. For all simulations, $Pr = 0.71$. The Navier-Stokes equations are solved using a structured grid where the numerical solution is the cell-averaged value for a cell with volume V

$$\vec{U}_h = I^h\vec{U} = \frac{1}{V}\iiint_V \vec{U}(x)dx. \tag{5.10}$$

The discrete operator for the Navier-Stokes equations is

$$L_h(\vec{U}_h) = \frac{\vec{U}_h^{n+1} - \vec{U}_h^n}{\Delta t} + \frac{1}{V}\sum_{j=1}^{J_{max}} F(\vec{U}_{h,j}^L, \vec{U}_{h,j}^R)\cdot\hat{n}_j A_j - S_h = 0, \tag{5.11}$$

where n is the current time step and j is the face for a given cell with spacing h. All solutions are steady-state and the temporal term is added from time marching to the steady-state solution. The discrete equations are solved using an implicit time marching scheme to compute steady state solutions. The function F is Riemann solver which operates on the left and right j^{th} cell face approximations $\vec{U}^L$ and $\vec{U}^R$. The flux scheme developed by Van Leer[27] is used for all solutions. The face approximations are second order accurate using MUSCL extrapolation which has a total stencil width of five cells in each coordinate direction. For a given face, the left and right states are computed

$$U_{i+1/2}^R = U_{i+1} - \frac{1}{4}\left[(1-\kappa)(U_{i+2} - U_{i+1}) + (1+\kappa)(U_{i+1} - U_i)\right] \tag{5.12}$$

$$U_{i+1/2}^L = U_i + \frac{1}{4}\left[(1-\kappa)(U_i - U_{i-1}) + (1+\kappa)(U_{i+1} - U_i)\right] \tag{5.13}$$

where i is the cell index and κ determines the order of accuracy and direction of the extrapolation. For all simulations, $\kappa = -1$ which is a fully upwinded extrapolation.

5.4.2 Axisymmetric Source Term

A source term is added to the Euler equations to simulate axisymmetric flow using a two-dimensional grid

$$\vec{S}_{axi} = -\frac{1}{r}\begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ 0 \\ v\left(\frac{P\gamma}{\gamma-1} - \rho\frac{u^2+v^2}{2}\right) \end{bmatrix}. \tag{5.14}$$

The geometric cell center is used as the radial distance $r = y_c$.

5.5 Applications

All applications are computed using the CFD code SENSEI.[28] SENSEI is a multiblock, finite-volume, structured Navier-Stokes flow solver. The solver has been verified using the method of manufactured solutions for all options used to compute the numerical benchmark solutions.[28] In addition, all simulations are iteratively converged to machine zero so that the iterative error is negligible relative to the discretization error.

5.5.1 Supersonic Inlet

The supersonic inlet application models an inviscid, two-dimensional inlet with a freestream Mach number of four. The inlet wedge is 30 degrees from the centerline exhausting into a duct $0.4\,m$ high. The height of the wedge is $0.2\,m$. The freestream properties are $p_\infty = 12,270\,Pa$ and $T_\infty = 233\,K$. The simulation is run as a calorically perfect gas with $\gamma = 1.4$ and air as the working fluid with $\hat{m} = 28.89\,kg/kmol$. The finest grid which is used as the numerical benchmark is 417x129 nodes. The numerical benchmark grid is coarsened by a factor of two five consecutive times to create a total of four grids where the coarsest grid is 53x17 nodes. The numerical solution computed on the 105x33 grid is shown in Figure 5.4.

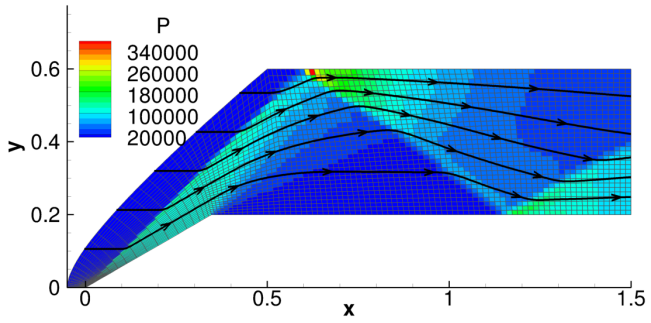


Figure 5.4: Supersonic inlet 105x33 node grid showing pressure contour and streamlines

5.5.2 JCEAP Blunted Cone

The Joint Computational/Experimental Aerodynamics Program (JCEAP) at Sandia National Laboratories performed a series of high quality validation experiments using a ten degree half angle blunted cone with a free stream Mach number of eight in a nitrogen flow (see references in Roy et al.[29]). An inviscid numerical benchmark is created using a simplified version of the geometry. The nose radius is 0.2 inches and the base diameter is 4 inches. The freestream properties are $P_\infty = 286.8\,Pa$ and $T_\infty = 47.7\,K$. The simulation is computed run as a calorically perfect gas with $\gamma = 1.4$ with nitrogen as the working fluid with $\hat{m} = 28.0\,kg/kmol$. The finest grid which is used as a numerical benchmark grid is 513x513 nodes. The numerical benchmark grid is coarsened by a factor of two five consecutive times to create a total of seven grids where the coarsest grid is 17x17 nodes. The numerical solution computed on a 65x65 node grid is shown in Figure 5.5.

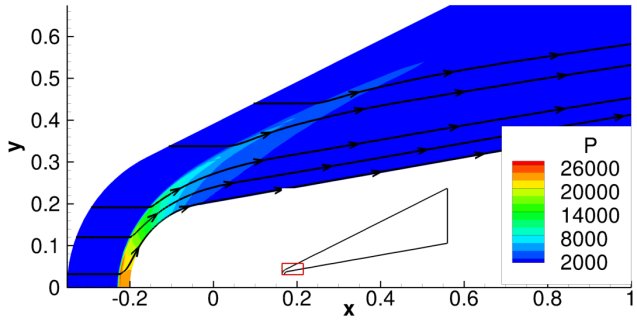


Figure 5.5: JCEAP blunted cone 65x65 node solution showing pressure contour and streamlines

5.5.3 NACA 0012 Airfoil

A NACA 0012 airfoil numerical benchmark is created where the grid is generated using the procedure described by Vassberg and Jameson.[30] The free stream Mach numbers are 0.5 and 0.8 with

freestream properties of $P_\infty = 101,325\,Pa$ and $T_\infty = 300\,K$, respectively, and the working fluid is air with $\hat{m} = 28.89\,kg/kmol$. Solutions are computed at an angle of attack of zero degrees and 1.25 degrees. An O-grid is created using the Karman-Trefftz conformal mapping to map the NACA 0012 geometry to a near circle. The O-grid is created for the circle and mapped back into physical space. The one modification to the grid generation from what was done by Vassberg and Jameson[30] was the grid spacing. Vassberg and Jameson[30] used equally spaced grid cells extending from the surface to a farfield boundary approximately 150 chord lengths from the airfoil. The farfield boundary error decreases at a first-order rate, so to avoid boundary issues the farfield was placed at 2400 chord lengths from the airfoil. A hyperbolic tangent distribution is used to cluster most of the grid cells near the airfoil. The spacing is set near the airfoil to be approximately $ds_1 = r \times 10^{-4}\,m$ for the finest grid created using a 4096x4096 node mesh and the farfield spacing is set to $ds_2 = 10\,m$. The hyperbolic spacing function is

$$A = \frac{\sqrt{ds_2}}{\sqrt{ds_1}}$$

$$B = \frac{1}{N\sqrt{ds_1 ds_2}}$$

where N is the number of cells. The value for δ is computed by solving

$$B = \frac{sinh\delta}{\delta}$$

$$u = \frac{1}{2}\left(1 + \frac{tanh\left(\delta\left(\frac{\xi}{N} - \frac{1}{2}\right)\right)}{tanh\left(\frac{\delta}{2}\right)}\right)$$

$$r = \frac{u}{A + (1-A)u} r_{max}$$

where ξ is the computational coordinate ranging from 0 to N. Figure 5.6 shows the NACA 0012 airfoil solution computed on a 129x65 node mesh at an angle of attack of zero degrees and Mach number of 0.8. The O-grid results in natural clustering near the leading and trailing edges of the airfoil. The numerical benchmark is computed on a 513x1025 node grid. The numerical benchmark is refined by a factor of two to create the coarsest grid with 17x33 nodes.

5.6 Results

The numerical errors for the numerical benchmark solutions are summarized in Table 5.1. The L_1 norm is used for the supersonic inlet and the JCEAP cone due to the presence of strong shocks and the L_2 norm is used for the airfoil cases to compute all quantities. The two uses of the numerical benchmarks are the estimation of truncation error and discretization error shown in the right two

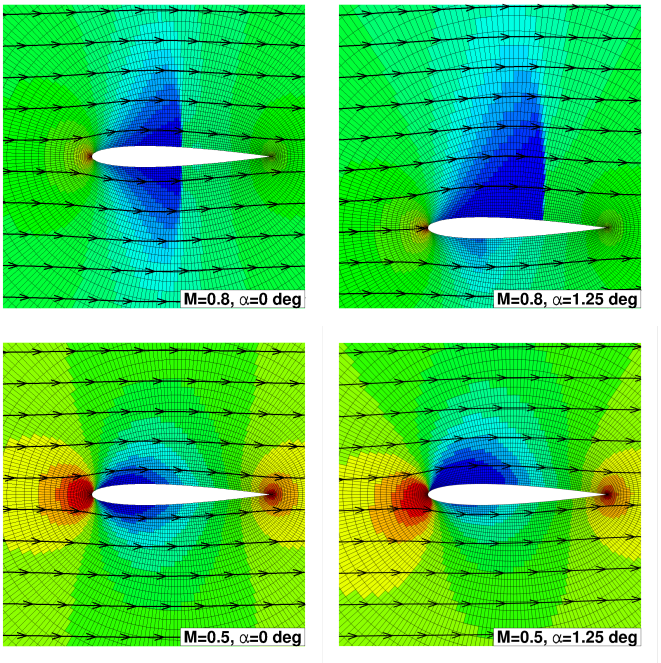


Figure 5.6: NACA 0012 Euler solutions showing pressure with streamlines

columns with the estimated percent error due to the numerical benchmarks. The important columns to consider with regard to the numerical benchmark are the iterative error and boundary condition error (where present) relative to the discretization error. For all applications, the iterative error is neglible. The boundary condition error was estimated only for the airfoils at an angle of attack of 1.25 degrees. The boundary condition error is more than two orders of magnitude lower than the discretization error for all variables except the y-velocity for the M=0.8 where the boundary condition error is only one order of magnitude lower than the discretization error.

The iterative and boundary condition numerical error sources in the numerical benchmark solutions are acceptably low compared to the benchmark discretization error. The benchmark discretization error has varying degrees of influence depending on the application. For all applications, the error in the exact truncation errors are less than three percent in the L_2-norm for all variables while most are less than one percent in the L_2-norm.

Considering the discretization error, the supersonic inlet and JCEAP cone have reasonably small errors. The largest error is about five percent in the L_2-norm in the density variable for the supersonic inlet. The numerical erros for the airfoil at M=0.8 and zero degrees have negligible error on the order of 1e-6 percent in the L_2-norm; whereas, the same application at an angle of attack of 1.25 degrees has the largest errors of all the applications with the largest error on the order of ten percent in the L_2-norm. The evaluated discretization error for the computational grid is on the same order as the zero degree case; however, the error in the numerical benchmark is an order of magnitude higher for all variables and the discretization error on the working grid is about one

order of magnitude lower than the discretization error for the zero degree angle of attack case. The opposite trends are seen in the airfoil at M=0.5. That is, the zero degree angle of attack case has larger error on the order of five percent, while at an angle of attack of 1.25 degrees the errors are very small. This is thought to be due to the relative magntidue of the discretization errors on the working grids. The discretization error for the zero degree angle of attack case is one to two orders of magnitude lower than the discretization error on the 1.25 degree angle of attack case. With the exception of density, the numerical benchmark discretization error for both angle of attacks is nearly identical (i.e., it agrees to two decimal places in the L_2 norm.

5.6.1 Summary of Numerical Error

The numerical benchmark errors for all applications are summarized in Table 5.1. All iterative errors were negligible and are omitted from the table. The working grid errors are computed using the method described in Section 5.3.

Table 5.1: Summary of numerical errors

Application	Var	NB Errors		Working Grid Errors	
		BC	DE	TE	DE
Supersonic Inlet	ρ	-	$5.0 \cdot 10^{-3}$	$1.77 \cdot 10^{3} \pm 0.2\%$	$1.0 \cdot 10^{-1} \pm 4.9\%$
M=4.0	u	-	$7.2 \cdot 10^{-1}$	$2.63 \cdot 10^{6} \pm 1.6\%$	$5.8 \cdot 10^{1} \pm 1.2\%$
53x17 nodes	v	-	$8.8 \cdot 10^{-1}$	$1.58 \cdot 10^{6} \pm 0.1\%$	$7.5 \cdot 10^{1} \pm 1.2\%$
	P	-	$1.8 \cdot 10^{1}$	$2.21 \cdot 10^{9} \pm 2.6\%$	$2.1 \cdot 10^{4} \pm 0.1\%$
JCEAP Cone	ρ	-	$1.1 \cdot 10^{-4}$	$3.1 \cdot 10^{2} \pm 2.7\%$	$3.8 \cdot 10^{-3} \pm 0.4\%$
M=8.0	u	-	$3.6 \cdot 10^{-2}$	$4.4 \cdot 10^{5} \pm 0.9\%$	$2.3 \cdot 10^{1} \pm 0.1\%$
65x65 nodes	v	-	$3.9 \cdot 10^{-2}$	$2.1 \cdot 10^{5} \pm 0.2\%$	$1.4 \cdot 10^{1} \pm 0.6\%$
	P	-	$1.2 \cdot 10^{1}$	$3.3 \cdot 10^{8} \pm 2.2\%$	$1.2 \cdot 10^{2} \pm 0.1\%$
Inviscid NACA 0012	ρ	-	$5.9 \cdot 10^{-4}$	$4.0 \cdot 10^{1} \pm 0.2\%$	$4.3 \cdot 10^{-5} \pm 0.0\%$
$M = 0.8, AOA = 0^{\circ}$	u	-	$2.1 \cdot 10^{-2}$	$3.0 \cdot 10^{4} \pm 0.3\%$	$6.8 \cdot 10^{-2} \pm 0.0\%$
33x65 nodes	v	-	$3.9 \cdot 10^{-3}$	$2.4 \cdot 10^{4} \pm 0.06\%$	$6.1 \pm 0.0\%$
	P	-	$6.9 \cdot 10^{1}$	$1.4 \cdot 10^{7} \pm 0.2\%$	$5.1 \pm 0.0\%$
Inviscid NACA 0012	ρ	$1.2 \cdot 10^{-5}$	$1.4 \cdot 10^{-3}$	$4.6 \cdot 10^{1} \pm 1.0\%$	$4.9 \cdot 10^{-5} \pm 6.2\%$
$M = 0.8, AOA = 1.25^{\circ}$	u	$7.0 \cdot 10^{-4}$	$4.1 \cdot 10^{-1}$	$2.6 \cdot 10^{4} \pm 1.1\%$	$3.2 \cdot 10^{-3} \pm 9.1\%$
33x65 nodes	v	$1.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$2.1 \cdot 10^{4} \pm 1.1\%$	$7.6 \cdot 10^{-3} \pm 5.6\%$
	P	$1.3 \cdot 10^{0}$	$1.7 \cdot 10^{2}$	$1.7 \cdot 10^{7} \pm 1.1\%$	$3.9 \cdot 10^{-1} \pm 4.7\%$
Inviscid NACA 0012	ρ	-	$6.4 \cdot 10^{-5}$	$2.6 \cdot 10^{1} \pm 0.6\%$	$1.3 \cdot 10^{-5} \pm 5.3\%$
$M = 0.5, AOA = 0^{\circ}$	u	-	$4.0 \cdot 10^{-1}$	$1.9 \cdot 10^{4} \pm 1.7\%$	$1.1 \cdot 10^{-2} \pm 6.3\%$
33x65 nodes	v	-	$1.6 \cdot 10^{-1}$	$1.3 \cdot 10^{4} \pm 0.9\%$	$2.9 \cdot 10^{-3} \pm 5.0\%$
	P	-	$5.6 \cdot 10^{1}$	$8.0 \cdot 10^{6} \pm 1.1\%$	$7.9 \cdot 10^{-2} \pm 4.6\%$
Inviscid NACA 0012	ρ	$3.3 \cdot 10^{-6}$	$6.4 \cdot 10^{-4}$	$2.5 \cdot 10^{1} \pm 0.7\%$	$3.0 \cdot 10^{-3} \pm 0.2\%$
$M = 0.5, AOA = 1.25^{\circ}$	u	$2.5 \cdot 10^{-4}$	$4.0 \cdot 10^{-1}$	$2.0 \cdot 10^{4} \pm 1.7\%$	$1.9 \pm 0.04\%$
33x65 nodes	v	$3.9 \cdot 10^{-3}$	$1.6 \cdot 10^{-1}$	$1.3 \cdot 10^{4} \pm 0.9\%$	$6.0 \cdot 10^{-1} \pm 0.01\%$
	P	$4.1 \cdot 10^{-1}$	$5.6 \cdot 10^{1}$	$8.0 \cdot 10^{6} \pm 1.1\%$	$3.1 \cdot 10^{2} \pm 0.0\%$

*very small percent error

5.7 Conclusion

Numerical benchmark solutions were computed for Euler and Navier-Stokes applications for the purpose of evaluating truncation error and discretization error estimation. Numerical errors including iterative error, boundary condition error, and discretization error were estimated for a numerical benchmark solution. Iterative error and boundary condition errors were negligible relative to the discretization error present in the numerical benchmark. The effects of the benchmark discretization error on the evaluation of the exact truncation and the exact discretization error were evaluated for solutions computed on a specific grid level. The effects on truncation error were fairly uniform

with L_1- and L_2-norm errors in the exact truncation error on the order of one percent. The errors in the evaluation of the exact discretization error range from negliglbe to errors on the order of ten percent. All errors are presented in the context of L_1 or L_2 norms. The effects of the benchmark discretizion error on global quantities are more easily compared and discussed within context of evaluating truncation error and discretization error estimation methods in part two of the manuscript.[31]

Bibliography

[1] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 210, pp. pp. 307–357, 1911.

[2] P. J. Roache and P. M. Knupp, "Completed richardson extrapolation," *Communications in Numerical Methods in Engineering*, vol. 9, no. 5, pp. 365–374, 1993.

[3] S. A. Richards, "Completed richardson extrapolation in space and time," *Communications in Numerical Methods in Engineering*, vol. 13, pp. 573–582, 1997.

[4] E. Fehlberg, "Low-order classical runge-kutta formulas with step size control and their application to some heat transfer problems," *NACA Technical Report 315*, 1969.

[5] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 337–357, 1987.

[6] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates, part 2: Error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1365–1382, 1992.

[7] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[8] H. J. Stetter, "The defect correction principle and discretization methods," *Numerische Mathematik*, vol. 29, pp. 425–443, 1978.

[9] R. D. Skeel, "Thirteen ways to estimate global error," *Numerische Mathematik*, vol. 48, pp. 1–20, 1986.

[10] A. Naumovich, M. Foerster, and R. Dwight, "Algebraic multigrid within defect correction for the linearized euler equations," *Numerical Linear Algebra with Applications*, vol. 17, pp. 307–324, 2009.

[11] X. D. Zhang, D. Pelletier, and J.-Y. Trepanier, "Verification of error estimators for the euler equations," Tech. Rep. AIAA-2000-1001, 2000.

[12] Y. Qin and T. I.-P. Shih, "A discrete transport equation for error estimation in cfd," Tech. Rep. AIAA-2002-0906, 2002.

[13] I. Celik and G. Hu, "Single grid error estimation using error transport equations," *J of Fluid Eng*, vol. 126, pp. 778–790, 2004.

[14] Y. Qin, K. Chi, and T. I.-P. Shih, "Modeling the residual in error-transport equations for estimating grid-induced errors in cfd solutions," Tech. Rep. AIAA-2006-892, 2006.

[15] T. I.-P. Shih and B. R. Williams, "Development and evaluation of an a posteriori method for estimating and correcting grid-indeuced errors in solutions of the navier-stokes equations," Tech. Rep. AIAA-2009-1499, 2009.

[16] P. Cavallo and N. Sinha, "An error transport equation with practical applications," Tech. Rep. AIAA-2007-4092, 2007.

[17] P. Cavallo, N. Sinha, and M. R. O'Gara, "Viscous error transport equation for error quantification of turbulent flows," *AIAA-2008-3851*, 2008.

[18] I. Babuska and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal of Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[19] I. Babuska and W. C. Rheinboldt, "A-posteriori error estimates for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[20] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[21] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from pde approximations," *SIAM Review*, vol. 42, no. 2, pp. 247–264, 2000.

[22] T. S. Phillips and C. J. Roy, "A new extrapolation-based uncertainty estimator for computational fluid dynamics," *AIAA-2013-0260*, 2013.

[23] T. S. Phillips, J. M. Derlaga, and C. J. Roy, "Numerical benchmark solutions for laminar and turbulent flows," *AIAA-2012-3074*, 2012.

[24] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[25] W. L. Oberkampf and T. G. Trucano, "Verification and validation benchmarks," *Nuclear Engineering and Design*, vol. 238, no. 3, pp. 716–743.

[26] C. J. Roy, "Review of code and solution verification procedures for computational simulation," *Journal of Computational Physics*, vol. 205, no. 1, pp. 131 – 156, 2005.

[27] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[28] J. M. Derlaga, T. S. Phillips, and C. J. Roy, "Sensei computational fluid dynamics code: A case study in modern fortran software development," Tech. Rep. AIAA-2013-2450, 2013.

[29] Roy, "Grid convergence error analysis for mixed-order numerical schemes," *AIAA Journal*, vol. 41, pp. 595–604, April 2003.

[30] J. C. Vassberg and A. Jameson, "In pursuit of grid convergence for two-dimensional euler solutions," *Journal of Aircraft*, vol. 47, no. 4, pp. 1152–1166, 2010.

[31] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Residual-based discretization error estimators for cfd part 2: Error estimates," *In Preparation*, 2014.

Chapter 6

Truncation error and Discretization Error Estimation

Residual-based Discretization Error Estimators for CFD Part 2: Error Estimates

Tyrone S. Phillips, Joseph M. Derlaga, Christopher J. Roy,
and Jeff Borggaard

Virginia Tech, Blacksburg, Virginia, 24061

Discretization error is usually the largest and most difficult numerical error to estimate in a Computational Fluid Dynamics (CFD) simulations. Development and evaluation of discretization error estimators ideally requires exact solutions to the PDEs which are only available for relatively simple cases. An alternative is to use numerical benchmarks which require a verified code and numerical solutions on a mesh which is sufficiently fine so that the discretization error in the benchmark solution is (essentially) negligible. The focus of this work is to use several numerical benchmark solutions to evaluate residual-based discretization error estimation methods which include defect correction and error transport equations for the Euler equations. This class of discretization error estimation methods requires accurate estimation of a residual/truncation error which will also be evaluated because the residual is the primary factor affecting the accuracy of the discretization error estimate, (i.e., an exact residual results in an exact discretization error estimate). The benchmark solutions include a supersonic inlet, a blunted hypersonic cone, and a NACA 0012 airfoil.

6.1 Introduction

Computational Fluid Dynamics (CFD) has enormous potential to impact the analysis, design, and optimization of engineering systems. The predictive capability of CFD depends not only on the validity of the sub-models employed and the uncertainties present in the system and surroundings, but also on the ability to accurately and reliably estimate and reduce numerical errors. While there are many sources of numerical error in CFD simulations, the largest and most difficult to estimate is the error related to the resolution of the spatial grid, i.e., the spatial discretization error.

The discretization error is defined as the difference between the exact solution to the discrete equations on a grid with cell size metric h, u_h, and the exact solution to the Partial Differential Equations (PDEs), $\tilde{u}$:

$$\varepsilon_h = u_h - \tilde{u} \tag{6.1}$$

For solutions on sufficiently fine grids the discretization error will reduce with grid refinement. The source of discretization error is the truncated terms of the infinite series used to approximate the PDEs and is called truncation error. As the grid is refined, higher-order terms in the truncation

error decrease at a faster rate than the leading term. The rate that truncation error decreases with grid refinement is proportional to the grid size parameter raised to the power of the formal order of accuracy of the numerical scheme, h^p. The grid size parameter is a measure of the discretization size in each coordinate direction (e.g., Δx) and the formal order of accuracy is the exponent of the leading truncation error term. Since truncation error can be shown to serve as the local source for the discretization error, the discretization error is expected to decrease at the same rate for structured grids. The truncation error can also be shown to be equivalent to the residuals found by inserting the numerical solution into the PDEs or the exact solution to the PDEs into the discretization scheme.[1,2] Approaches that estimate the truncation error are thus often referred to as residual-based methods

Several different classifications of discretization error estimators are available and each has different advantages and disadvantages. Higher-order methods estimate the discretization error by comparing the discrete solution to a higher-order estimate of the exact solution found by post-processing the numerical solutions. This category includes Richardson extrapolation,[3,4,5] order refinement,[6] and finite element recovery methods.[7,8] The most widely used discretization error estimator is Richardson extrapolation which uses a series of solutions on increasingly finer grid resolution to construct a higher order accurate solution. Richardson extrapolation is simple to apply as a post processing step and does not require code modification. The disadvantage of Richardson extrapolation is the requirement of multiple grid levels, since grid generation is one of the most time consuming tasks for a numerical simulation. In addition to the expense of generating multiple grid levels, each grid level used in Richardson extrapolation must also be in the asymptotic range for the estimates to be reliable.

Residual-based methods use a discrete solution combined with information about the PDE or discrete equations being solved to estimate the discretization error. They generally require two solutions to be computed on the same grid. This category includes defect correction methods,[9,10,11,12] Error Transport Equations (ETEs),[13,14,14,15,16,17,18,19] finite element residual methods,[20,21] and adjoint methods.[22,23]. Residual-based methods require only one grid level which reduces the grid generation requirements. These methods have the potential to be more accurate than Richardson extrapolation since they only require a single grid in the asymptotic range and they bring in additional information about the equations being solved. If the exact truncation error is used then the exact solution to the PDEs can potentially be recovered. Accurate truncation error estimation is a key element for an accurate discretization error estimate.

The residual-based methods of interest are defect correction and ETEs. Defect correction corrects the original numerical solution by re-solving the original problem with the truncation error estimate added to the right-hand side. The resulting solution is an estimate of the exact solution and the difference between the original solution and the defect correction solution is an estimate of the discretization error. The ETEs work in a similar manner except the ETEs solve directly for the discretization error by linearizing the original governing equations about a solution so that the discretization error is directly computed. ETEs have been applied to several simple model problems such as the advection/diffusion equation, Burgers' equation, wave equation, etc.[14,14,15,16] The ETEs were developed for laminar viscous flow,[17] and were solved on much more complex

problems including an airfoil, an aircraft engine exhaust, and a rocket plume.[18,19]

The ETEs must be properly formulated to accurately capture the transport of discretization error through the domain in order to produce an accurate discretization error estimate. Phillips and Roy[24] compared different linearization methods for Burgers' equation and show that improper linearization results in a discretization error estimate which is offset by a constant value. Two new formulations were given: the first corrected the offset but dropped the higher order nonlinear terms, while the second formulation included the nonlinear terms. Little difference in the accuracy of the discretization error methods was seen for Burgers' equation; however, Banks et al.[25] showed that for problems with strong discontinuities the nonlinear terms are required for an accurate discretization error estimate.

Previous work evaluate the discretization error estimates using a solution computed on a finer grid than the computational grid (generally refered to as a numerical benchmark).[14,14,15,16,17,18,19] The method used to estimate the residual (i.e. truncation error) used a multi-grid approach which is condusive to the resulting discretization error evaluations published due to error cancelation. Error cancelation does not occur for single grid truncation error estimation methods which have exhibited several advantages over multi-grid truncation error estimation methods (see Phillips and Roy.[26]) Part one of this work[27] developed several numerical benchmarks with carefully characterized numerical errors. The discretization error in the numerical benchmark is used to add appropriate error bars to the resulting truncation error and discretization error evaluations.

6.2 TE Framework

A set of operators are used to represent the governing equations and prolongation/restriction operations. Prolongation is an operation that transforms a lower-spaced solution to a higher-spaced solution (e.g. solution reconstruction), and restriction is an operation that transforms a higher-spaced solution to a lower-spaced solution. The numerical solution to a given set of PDEs can be represented by the General Truncation Error Expression (GTEE)[28]

$$L_h(\cdot) = L(\cdot) + \tau_h(\cdot) \tag{6.2}$$

where $L_h(\cdot)$ represents the discretization operator, $L(\cdot)$ represents the continuous governing equations, and $\tau_h(\cdot)$ represents the truncation error. Note that different truncation error equations can be derived by evaluating the GTEE on either continuous or discretized solutions and apply the needed prolongation or restriction operators. For example, if f is a continuous function then

$$L_h(I^h f) = I^h L(f) + \tau_h(f) \tag{6.3}$$

where I^h is an interpolation operator and is used to move from a continuous space to a discrete space with mesh size h. In general, we consider interpolation operators I_a^b where a is the starting space and b is the target space. The discrete space h represents a solution on a mesh with spac-

ing h and a blank sub/superscript represents a continuous (infinite dimensional) space with infinite degrees of freedom. The restriction of a continuous space to a discrete space is relatively straightforward. For a finite difference method the restriction could simply be considered as the evaluation of the continuous function at the i-th x location

$$f_{i,h} = I^h f = f(x_i). \tag{6.4}$$

For a finite volume method the natural restriction to a discrete space is the average value of the function over the i-th cell with volume V_i

$$f_{i,h} = I^h f = \frac{1}{V_i} \int_{V_i} f(x) dV. \tag{6.5}$$

If we consider a discretized function, then the GTEE can be written as

$$L_h(f_h) = I^h L(I_h f_h) + \tau_h(I_h f_h). \tag{6.6}$$

The prolongation from discrete space to continuous space is an approximation that is better written as

$$I_h = I_h^q + O(h^{q+1}) \tag{6.7}$$

to account for the error in the prolongation, where q is the order of the reconstructed polynomial for which the k-exact and ENO schemes are $q+1$ order accurate. Equation 6.6 is then written as

$$L_h(f_h) = I^h L(I_h^q f_h + O(h^{q+1})) + \tau_h(I_h^q f_h + O(h^{q+1}))$$

or

$$L_h(f_h) = I^h L(I_h^q f_h) + \tau_h(I_h^q f_h) + O(h^{\bar{q}}) \tag{6.8}$$

where $\bar{q}$ is an order of accuracy which is related to q may be modified by nonlinearities in the $L(.)$ and $\tau_h(.)$ operators. Due to the error in the prolongation, it cannot be generally assumed that $f_h = I^h I_h f_h$.

As is evident from the GTEE, the truncation error is the difference between the discrete and continuous governing equations and has been shown to be the local source of discretization error in a numerical solution[28] where the discrete form of the discretization error is defined as

$$\varepsilon_h = u_h - I^h \tilde{u}. \tag{6.9}$$

Here, u_h is the exact solution to the discrete equations such that $L_h(u_h) = 0$, and $\tilde{u}$ is the exact solution to the original governing equations such that $L(\tilde{u}) = 0$.

6.3 Truncation Error Estimation

The accurate estimation of truncation error is a key aspect to several residual-based discretization error estimation methods such as the error transport equations,[13,14] defect correction,[29,10,11] and adjoint methods.[30] Also, as the local source of discretization error, truncation error has been shown to be an effective mesh adaption indicator,[28] possibly with adjoint weighting.[31]

Inserting the exact solution to the discrete equations u_h into Equation 6.2, and noting that $L_h(u_h) = 0$, results in

$$\tau_h(I_h^k u_h) = -I^h L(I_h^k u_h) + O(h^{\bar{k}}). \tag{6.10}$$

The accuracy of the truncation error estimate is determined by comparing the estimated truncation error to the exact truncation error which requires the exact solution to the governing equations. It can be found by inserting the exact solution to the PDEs into the GTEE and, noting that $L(\tilde{u}) = 0$, gives

$$\tau_h(\tilde{u}) = L_h(I^h \tilde{u}) \tag{6.11}$$

where the exact and approximate truncation errors are related by

$$\tau_h(\tilde{u}) \approx \tau_h(I_h^k u_h). \tag{6.12}$$

Various method of estimating truncation error have been developed and tend to be specific to the application. There are two general categories: single grid and multiple grid truncation error estimation methods, both of which will be examined herein.

6.3.1 Single Grid Truncation Error Estimation

Single grid truncation error estimation is based on Equation 6.8. The distinction between the methods depend on the solution reconstruction scheme. Rewriting Equation 6.8 as

$$\tau_h(I_h^k f_h) = L_h(I_k^h I_h^k f_h) - I^h L(I_h^k f_h) \tag{6.13}$$

shows that there are two components to the single grid truncation error estimation. Ideally, a reconstruction scheme should reproduce the mean value of the numerical solution for the given cell for the finite volume method (or the nodal value for finite difference) so that $u_h = I_k^h I_h^k u_h$. If the mean is conserved then

$$\tau_h(I_h^k f_h) = -I^h L(I_h^k f_h). \tag{6.14}$$

A few different reconstruction schemes are considered and are based off solution reconstruction schemes used for higher order discrete soutions. The k-exact method developed by Barth[32,33] is a full dimension reconstruction method (e.g. two dimensional stencil for a two dimensional simulation). Essentially Non-Oscillatory (ENO) methods have been developed for better discontinuity resolution and implement an adaptive one-dimensional stencil based on divided differences. Two-dimensional and one-dimension ENO methods were compared by Phillips et al.[26] and resulted in

nearly identical truncation error and discretization error estimates. The dimensionally split ENO method developed by Godfrey et al.[34] is less computationally expensive. A least squares method is also included which does not conserve the mean of the numerical solution. The method is exactly the same as the k-exact method but a larger stencil is used.

K-exact and Least Squares Methods

The k-exact method developed by Barth[32,33] was designed to conserve the mean value of the cell, reconstruct polynomials of degree k or less exactly, be compact, and be computationally efficient. The k-exact method is modified so that the higher dimension polynomial is a tensor product of one-dimensional polynomials with $(k+1)^d$ unknowns where d is the dimension and $N_{st} = (k+1)^d$. This results in more polynomial terms. The number of terms results in a square stencil (the number of stencil points in each dimensional direction) where the number of unknowns is equal to the number of equations so that the mean is conserved for every cell in the stencil. For a polynomial of order k there are $(k+1)^d$ unknowns and a stencil width of $N_{st} = k+1$ where d is the dimension. A centered stencil is used for the interior of the domain and a shifted stencil near the computational boundary. The k-exact method developed by Barth[32,33] is solved in a least squares method by increasing the size of the stencil so that the mean is conserved only for the cell of interest, whereas the method implemented here results in the smallest possible stencil and is equally sized in each coordinate dimension.

The least squares method is setup exactly the same as the k-exact method except the stencil is larger by one cell in each dimension $N_{st} = k+2$. The least squares method does not conserve the mean of any of the cells in the stencil so the term $L_h(I_k^h I_h^k u_h)$ should be included in the truncation error estimate.

ENO Method

The ENO method developed by Harten[35] was extended to arbitrary dimensions by Godfrey et al.[34]. Godfrey et al.[34] also introduced a less computationally expensive dimensionally split ENO (DS-ENO) method but does not include cross-derivative terms. For the Euler equations, the cross-derivative terms are not important; however, for the Navier-Stokes equations, the accuracy of the truncation error estimate degrades signficantly in high sheer regions of the flow.[27]

The basis of the ENO method is the selection of the stencil to reduce the variation of the solution used to compute the reconstruction. Starting at cell i, the difference between the solutions to the left $(i-1)$ and right $(i+1)$ are compared and the minimum is added to the reconstruction stencil. The divided difference is repeated until the stencil has N_{st} points. The reconstruction is computed over the stencil using a one dimensional k-exact reconstruction method. The computational cost of the DS-ENO method is an order of magnitude cheaper than the k-exact and least squares reconstruction methods; however, in our case, computational expense is less of a concern since the truncation error

is generally estimated once during the solution process.

6.3.2 Multiple Grid Truncation Error Estimation

Multiple grid truncation error estimation methods are used for a variety of residual-based error estimators. Adjoint methods typically use an embedded grid approach developed by Venditti and Darmafal[31]. This method interpolates a numerical solution onto a finer grid. The discrete residual on the finer grid is related to the truncation error

$$\tau_h(I_h u_h) = -I^h_{h/r} L_{h/r}\left(I^{h/r}_k I^k_h u_h\right) + O(h^{\bar{k}}) \tag{6.15}$$

where h represents the coarse grid and h/r represents the fine grid (refined by factor r). This method was modified by Phillips et al.[26] to improve accuracy since the truncation error on the fine grid solution is still significant for typical refinement factors of two. Equation 6.15 is modified to include the effects the fine grid truncation error assuming that the truncation error is asymptotic so that $\tau_h = \frac{\tau_{h/r}}{r^{p_f}}$ where p_f is the formal order of accuracy of the truncation error

$$\tau_h(I_h u_h) = -I^h_{h/r} L_{h/r}\left(I^{h/r}_k I^k_h u_h\right)\left(\frac{r^{p_f}}{r^{p_f}-1}\right) + O(h^{\bar{k}}). \tag{6.16}$$

A detailed derivation of this correction factor is given by Phillips et al[36]. The formal order of truncation error is usually the same as the formal order of accuracy of the discretization error; however, for unstructured grids or grids with randomly perturbed nodes (i.e., non smooth grid transformations), the formal order of truncation error can be lower than the formal order of the discretization error. Shih and Williams[17] estimated truncation error for the error transport equations by interpolating a finer grid solution onto a coarse grid and computing the discrete residual

$$\tau_h(I_h u_h) = L_h\left(I^h_{h/r} u_{h/r}\right). \tag{6.17}$$

Fulton[37] used a similar embedded grid approach, but adjusted the truncation error estimate to take into account the relative magnitudes of the truncation error between the two grids (again assuming the formal order of accuracy)

$$\tau_h(I_h u_h) = L_h\left(I^h_{h/r} u_{h/r}\right)\left(\frac{r^{p_f}}{r^{p_f}-1}\right). \tag{6.18}$$

Instead of using a numerical solution computed on a finer grid, Phillips et al.[26] interpolated the current numerical solution onto a coarser grid and the resulting discrete residual/truncation error is reinterpolated back to the computational grid

$$\tau_h(I_h u_h) = I^h_k I^k_{rh} L_{rh}\left(I^{rh}_h u_h\right)\left(\frac{1}{r^{p_f}-1}\right) + O(h^{k+1}). \tag{6.19}$$

6.4 Discretization Error Estimation

6.4.1 Defect Correction

Defect correction corrects the original numerical solution by re-solving the original problem with the truncation error estimate added to the right-hand side. Defect correction requires minimal code intrusion. The exact defect correction problem is defined as

$$L_h(I^h\tilde{u}) = \tau_h(\tilde{u}). \tag{6.20}$$

If the exact truncation error $\tau_h(\tilde{u})$ is added as a source term, the resulting numerical solution is the exact solution to the PDEs or integral equations restricted to the computational grid. Our defect correction implementation takes the form

$$L_h(\bar{u}) = \tau_h(I_h^k u) \tag{6.21}$$

where $\bar{u} \approx \tilde{u}$ and the discretization error estimate is

$$\varepsilon_h \approx \bar{\varepsilon}_h = u_h - \bar{u}. \tag{6.22}$$

Defect correction requires a second numerical solution on the original computational mesh; however, computational expense is reduced because the original numerical solution is used to initialize the second simulation, thus providing a very good starting solution.

6.4.2 Error Transport Equations

The discrete ETEs are derived by substituting the exact solution into Equation 6.2 and subtracting $L_h(u_h) = 0$ and noting that $L(\tilde{u}) = 0$

$$L_h(u_h) - L_h(I^h\tilde{u}) = -\tau_h(\tilde{u}). \tag{6.23}$$

If the discrete operator is linear or linearized then $L_h(u_h) - L_h(I^h\tilde{u}) = L_h(\varepsilon_h)$ which gives

$$L_h(\varepsilon_h) = -\tau_h(\tilde{u}). \tag{6.24}$$

The new discrete operator is a discrete equation that governs the transport of disretization error through the computational domain with truncation error as the source. Nonlinear equations can be linearized using Taylor's series

$$L_h(I^h\tilde{u}) = L_h(u_h) - \bar{\varepsilon}_h \frac{\partial L_h(u_h)}{\partial u} + \frac{\bar{\varepsilon}_h^T}{2} \frac{\partial^2 L_h(u_h)}{\partial u^2} \bar{\varepsilon}_h - O(\bar{\varepsilon}_h^3) \tag{6.25}$$

and the final ETE reduces to

$$\frac{\partial L_h(u_h)}{\partial u}\bar{\varepsilon}_h = -\tau_h(I_h u_h) + O(\bar{\varepsilon}_h^2). \tag{6.26}$$

The derivative is the left-hand size of the linearized discrete operator which is required to implement an implicit scheme. The left-hand side is commonly available in flow solvers; however, the left-hand side required to solve the ETEs should be the full higher order (e.g., second-order) left-hand side where only the first-order left-hand side may be required for an implicit solve. The ETE is approximate so the exact discretization error may not result if the exact truncation error is used; however, the error in Equation 6.26 decreases at a much higher rate and would introduce non-negliglbe error only in the case of very large discretization errors.

6.5 Governing Equations

6.5.1 Navier-Stokes

The finite volume formulation of the Navier-Stokes equations is

$$L(\vec{U}) = \frac{\partial}{\partial t}\iiint_V \vec{U}dV + \iint_A \left(\vec{F}_i - \vec{F}_v\right)dA - \iiint_V \vec{S}dV = 0 \tag{6.27}$$

where the inviscid flux is

$$\vec{F}_i = \begin{bmatrix} \rho\vec{v}\cdot\hat{n} \\ \rho u\vec{v}\cdot\hat{n} + n_x p \\ \rho v\vec{v}\cdot\hat{n} + n_y p \\ \rho w\vec{v}\cdot\hat{n} + n_z p \\ \rho h_t\vec{v}\cdot\hat{n} \end{bmatrix} \tag{6.28}$$

the viscous flux is

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\theta_x + n_y\theta_y + n_z\theta_z \end{bmatrix} \tag{6.29}$$

The outward pointing face normal at side of a grid cell is $\hat{n} = [n_x, n_y, n_z]$, the conserved variable vector is $\vec{U} = [\rho, \rho u, \rho v, \rho w, \rho e_t]$, the velocity vector is $\vec{V} = [u, v, w]$, the total energy e_t and total enthalpy h_t are computed using the calorically perfect gas assumption with an equation of state

$p = \rho RT$, $e_t = \frac{p}{\rho(\gamma-1)}$, and $h_t = \frac{p\gamma}{\rho(\gamma-1)} + \frac{|\vec{V}|^2}{2}$. The diffusive components of the viscous flux are

$$\tau_{xx} = 2\mu\left(\frac{\partial u}{\partial x} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{yy} = 2\mu\left(\frac{\partial u}{\partial y} - \frac{1}{3}\nabla\cdot\vec{V}\right), \quad \tau_{zz} = 2\mu\left(\frac{\partial u}{\partial z} - \frac{1}{3}\nabla\cdot\vec{V}\right),$$
$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right), \quad \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right),$$
$$\theta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x}, \quad \theta_y = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y}, \quad \theta_z = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z}$$

and the heat conduction coefficient for a calorically perfect gas is $k = \frac{R\gamma\mu}{Pr(\gamma-1)}$. For all simulations, $Pr = 0.71$. The Navier-Stokes equations are solved using a structured grid where the numerical solution is the cell-averaged value for a cell with volume V

$$\vec{U}_h = I^h\vec{U} = \frac{1}{V}\iiint_V \vec{U}(x)dx. \tag{6.30}$$

The discrete operator for the Navier-Stokes equations is

$$L_h(\vec{U}_h) = \frac{\vec{U}_h^{n+1} - \vec{U}_h^n}{\Delta t} + \frac{1}{V}\sum_{j=1}^{J_{max}} F(\vec{U}_{h,j}^L, \vec{U}_{h,j}^R)\cdot\hat{n}_j A_j - S_h = 0, \tag{6.31}$$

where n is the current time step and j is the face for a given cell with spacing h. All solutions are steady-state and the temporal term is added from time marching to the steady-state solution. The discrete equations are solved using an implicit time marching scheme to compute steady state solutions. The function F is Riemann solver which operates on the left and right j^{th} cell face approximations $\vec{U}^L$ and $\vec{U}^R$. The flux scheme developed by Van Leer[38] is used for all solutions. The face approximations are second order accurate using MUSCL extrapolation which has a total stencil width of five cells in each coordinate direction. For a given face, the left and right states are computed

$$U_{i+1/2}^R = U_{i+1} - \frac{1}{4}\left[(1-\kappa)(U_{i+2} - U_{i+1}) + (1+\kappa)(U_{i+1} - U_i)\right] \tag{6.32}$$

$$U_{i+1/2}^L = U_i + \frac{1}{4}\left[(1-\kappa)(U_i - U_{i-1}) + (1+\kappa)(U_{i+1} - U_i)\right] \tag{6.33}$$

where i is the cell index and κ determines the order of accuracy and direction of the extrapolation. For all simulations, $\kappa = -1$ which is a fully upwinded extrapolation.

6.5.2 Axisymmetric Source Term

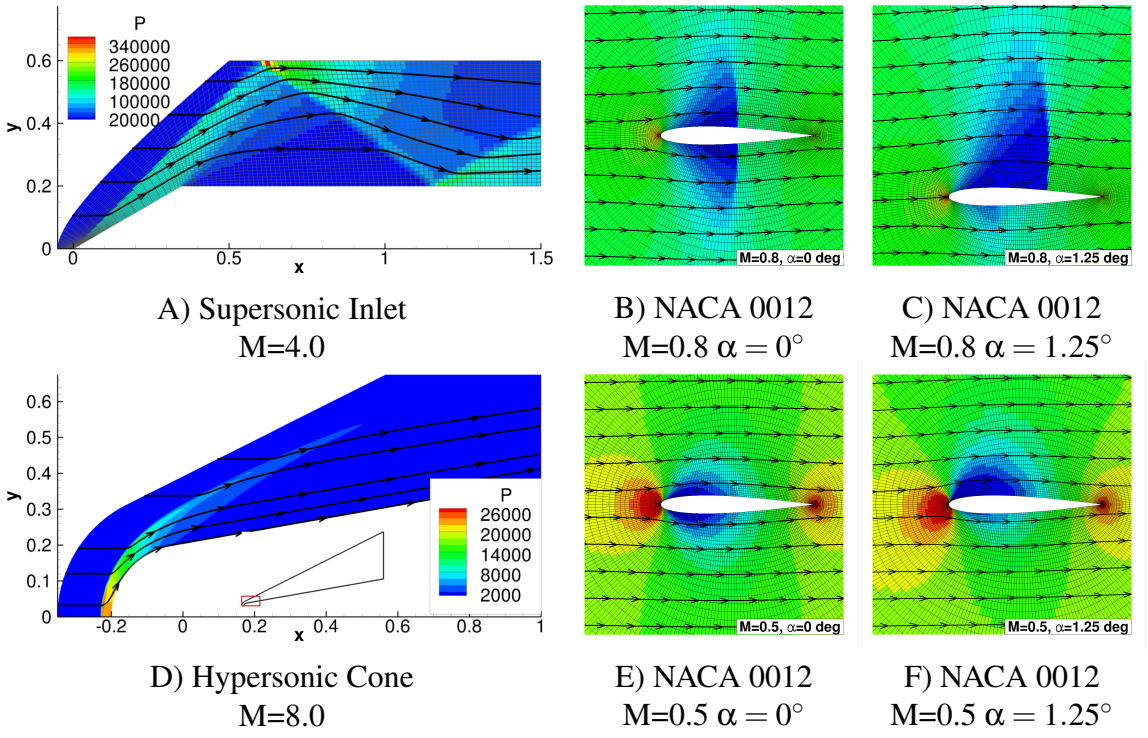
A source term is added to the Euler equations to simulate axisymmetric flow using a two-dimensional grid

$$\vec{S}_{axi} = -\frac{1}{r}\begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ 0 \\ v\left(\frac{P\gamma}{\gamma-1} - \rho\frac{u^2+v^2}{2}\right) \end{bmatrix}. \tag{6.34}$$

The geometric cell center is used as the radial distance $r = y_c$.

6.6 Applications

Numerical benchmark solutions were computed for the purpose of evaluating truncation error and dicretization error estimation methods for a variety of applications including a supersonic inlet, hypersonic cone, transonic two-dimensional airfoil solutions and subsonic two-dimensional airfoil solutions. The numerical solutions are shown in Figure 6.1. For more details regarding these benchmark solutions, see Phillips et al.[39]



A) Supersonic Inlet M=4.0

B) NACA 0012 M=0.8 $\alpha = 0°$

C) NACA 0012 M=0.8 $\alpha = 1.25°$

D) Hypersonic Cone M=8.0

E) NACA 0012 M=0.5 $\alpha = 0°$

F) NACA 0012 M=0.5 $\alpha = 1.25°$

Figure 6.1: Euler solutions showing pressure with streamlines

6.7 Results

For each application, the truncation error is estimated using the k-exact, least squares (LSQ), DS-ENO single grid methods as well as the coarse grid method (with and without the correction term) and the fine grid method (with and without the correction term). Defect correction and ETEs are solved with each of the truncation error estimates.

6.7.1 NACA 0012

The truncation error estimates for the subsonic airfoil at zero degrees angle of attack are shown in Figure 6.2. The k-exact and fine grid method with the correction term capture the truncation error around the leading edge of the airfoil. The least squares method compares well also with the exception of a large truncation error estimate at the leading edge. The DS-ENO method results in a very noisy truncation error estimate. The discretization error in pressure is shown in Figure 6.3. The accuracy of the truncation error estimates directly correspond to the accuracy of the discretization error estimate. The k-exact estimate compares well to the exact discretization error and the DS-ENO method is relatively noisy.

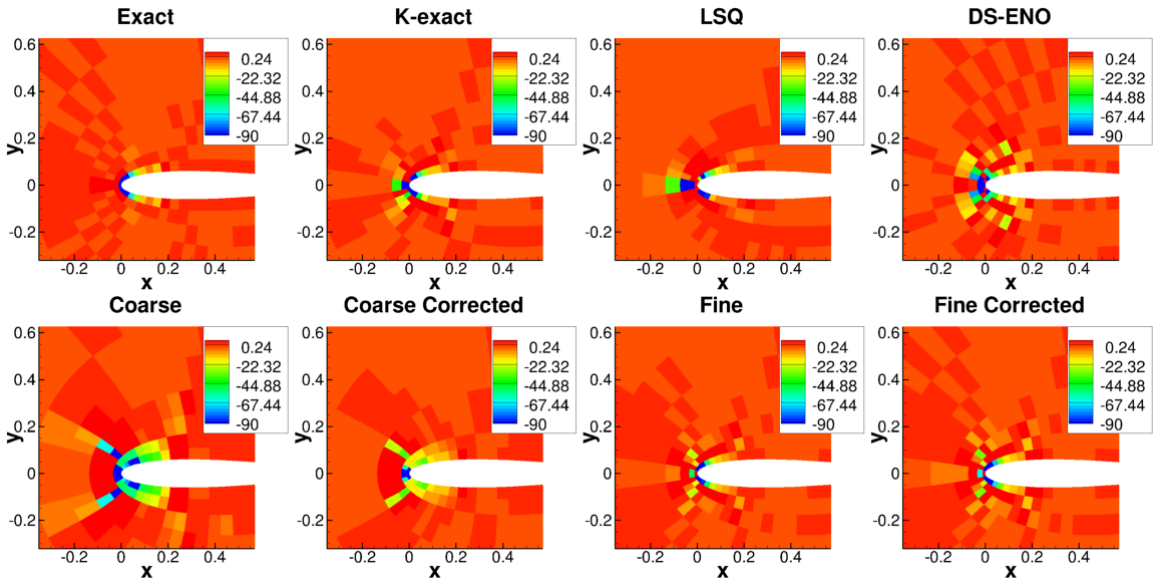


Figure 6.2: Estimated truncation error for the mass equation for the NACA 0012 airfoil at M=0.5 and $\alpha = 0^{\circ}$

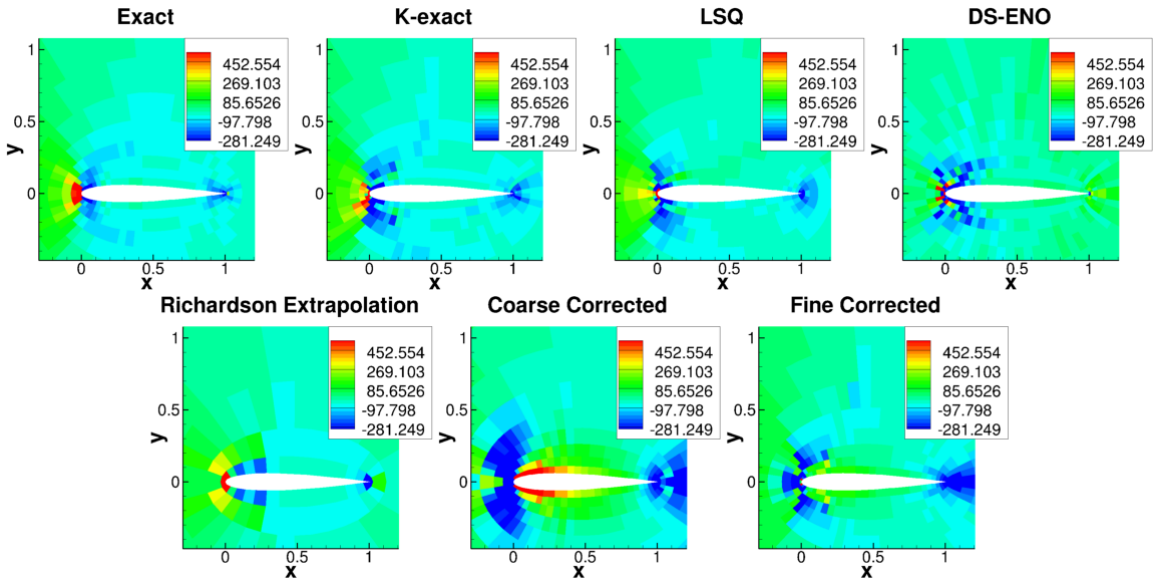


Figure 6.3: Estimated defect correction discretization error estimates for pressure for the NACA 0012 airfoil at M=0.5 and $\alpha = 0°$

To compare the difference between defect correction and the ETEs, the exact, defect correction estimate, and two ETE estimates are shown in Figure 6.4. One ETE estimate is computed using a Jacobian computed for first-order discrete equations and the other is computed using a Jacobian computed for the second order discrete equations. The results are very nearly identical and only differ by fractions of a percent. The pressure on the upper wall is also compared in Figure 6.5. The results are nearly identical for the ETE estimates. The result is fortuitous because a first order Jacobian is often readily available when using an implicit solver. The solution time for the ETEs using the first order Jacobian is negligible compared to the expense of the defect correction estimate. For comparison, the primal solution took 24 minutes while using the k-exact truncation error estimation method, the defect correction error estimate requires about nine minutes to solve while the first-order ETE took less than a second to compute with three iterations. The second-order ETE cost about five seconds to compute with seven iterations. The first-order ETE is so inexpensive because the equations are linear. The difference between the first- and second-order ETE is negligible for all applications so only the first-order ETE results are shown. The ETE estimates slightly underestimate the discretization error at the leading edge compared to defect correction. Both residual-based methods accurately estimate the discretization error while Richardson extrapolation misses the sign and magnitude near the leading edge significantly.

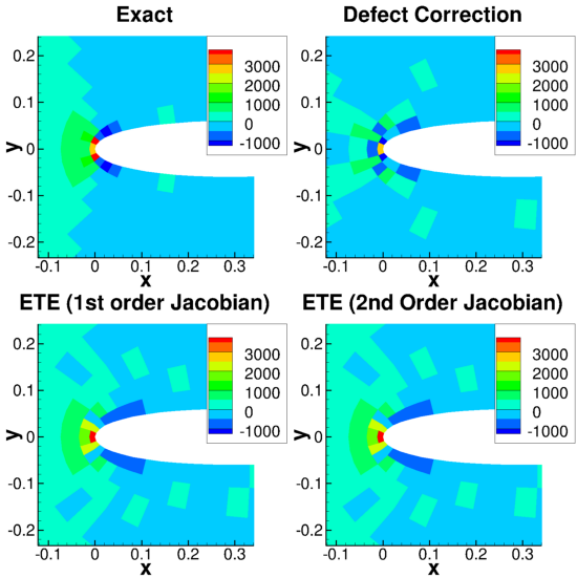


Figure 6.4: Estimated defect correction and ETE discretization error estimates for pressure for the NACA 0012 airfoil at M=0.5 and $\alpha = 0^\circ$ using the k-exact reconstruction method (solid is the benchmark solution)

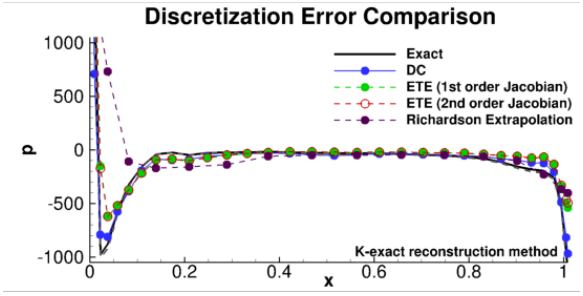


Figure 6.5: Estimated defect correction and ETE discretization error estimates for pressure along the upper side of the the NACA 0012 airfoil at M=0.5 and $\alpha = 0^\circ$ using the k-exact reconstruction method (solid is the benchmark solution)

Figures 6.6 and 6.7 compare the pressure on the wall for defect correction and ETE discretization error estimates compared to the exact discretization error. For all cases, the ETE estimate compares well to the defect correction estimate. There are slight differences near discretization error peaks, and overall, the defect correction estimate is more accurate because the estimate results from nonlinear equations compared to the linearized ETEs. In terms of truncation error estimation, the k-exact method is the most accurate with only a minor difference in peak error near the leading edge. The LSQ, DS-ENO, and fine grid method with the correction factor compare well for the entire airfoil with larger differences near the leading edge of the airfoil. All methods compare better than Richardson extrapolation except for the coarse grid methods which are inaccurate over

the entire airfoil.

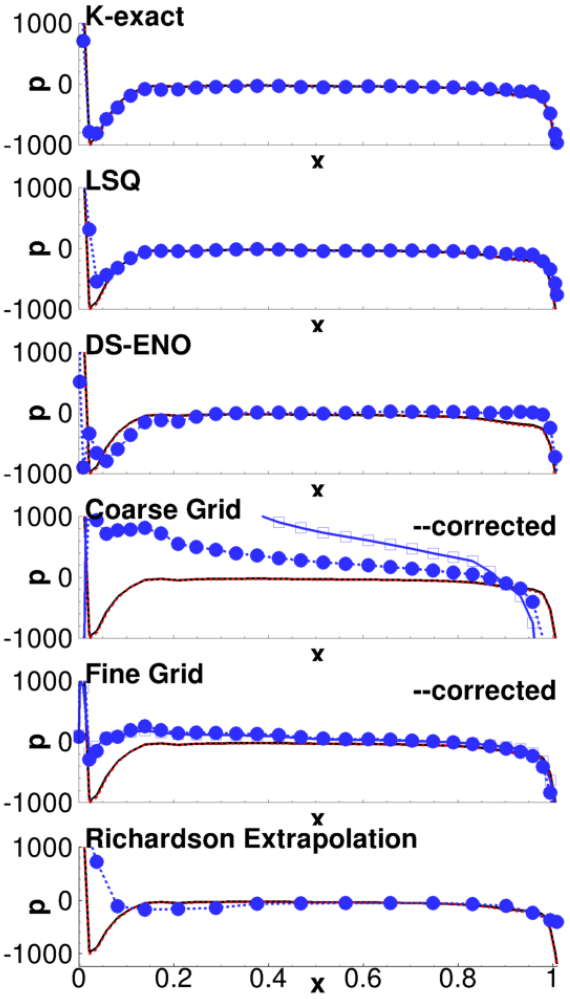


Figure 6.6: Estimated defect correction discretization error estimates for pressure along the upper side of the NACA 0012 airfoil at M=0.5 and $\alpha = 0^\circ$ (solid is the benchmark solution)

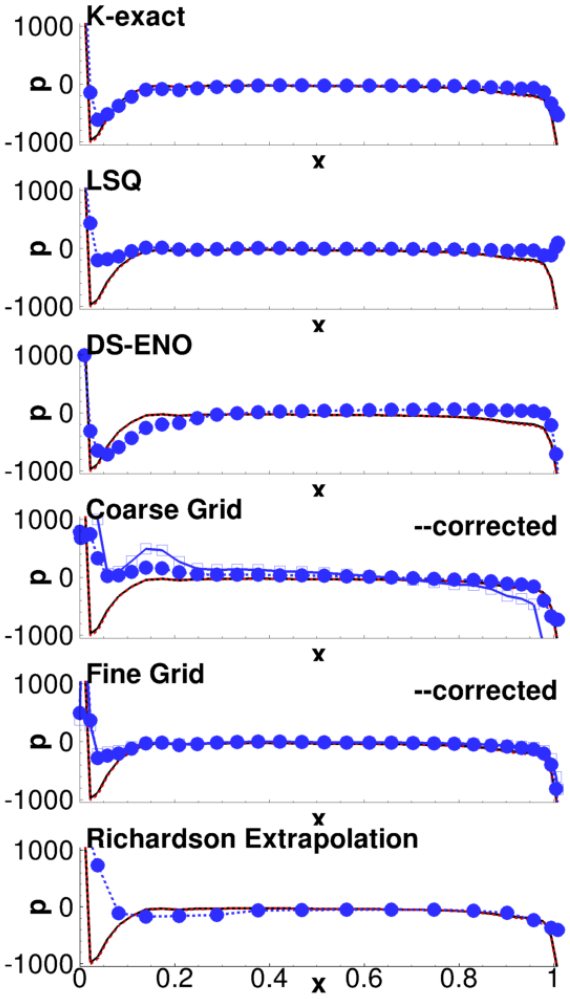


Figure 6.7: Estimated ETE discretization error estimates for pressure along the upper side of the the NACA 0012 airfoil at M=0.5 and $\alpha = 0^\circ$ (solid is the benchmark solution)

The same process is applied at an angle of attack of 1.25 degrees. The pressure along the upper surface of the airfoil is shown in Figures 6.8 and 6.9 for defect correction and ETE, respectively. For this case, the k-exact ETE method is the most accurate followed by the DS-ENO ETE method. Defect correction using the fine grid method truncation error estimate is one of the most accurate estimates with slight discrepancies in the first two cells. Defect correction using the k-exact truncation error estimate is not accurate near the leading edge. This is most likely due to a poor truncation error estimate at the leading edge due to poor grid resolution. The corresponding ETE

estimate is more accurate because the linearized equations dampen the effect of the overestimate of truncation error. All methods perform better than Richardson extrapolation except for the defect correction k-exact and LSQ estimates.

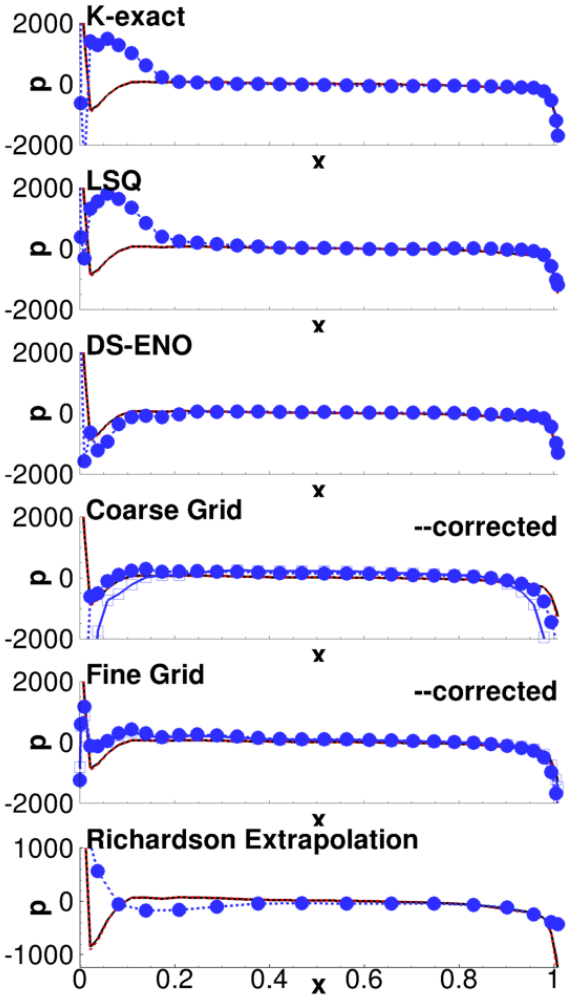


Figure 6.8: Estimated defect correction discretization error estimates for pressure along the upper side of the NACA 0012 airfoil at M=0.5 and $\alpha = 1.25°$

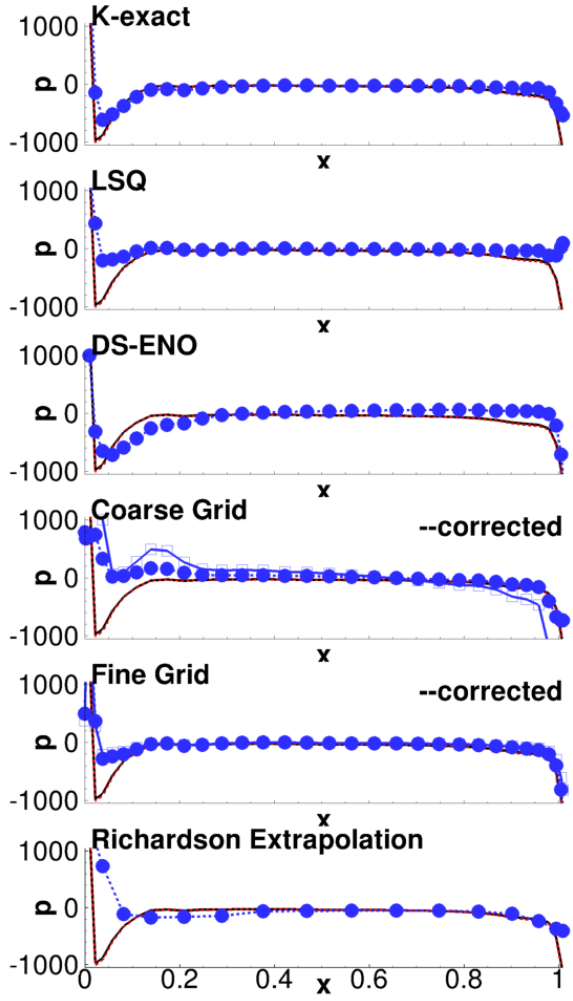


Figure 6.9: Estimated ETE discretization error estimates for pressure along the upper side of the the NACA 0012 airfoil at M=0.5 and $\alpha = 1.25°$

The discretization error in pressure on the upper side of the airfoil for a freestream Mach of 0.8 at zero degress angle of attack is shown in Figures 6.10 and 6.11 for defect correction and ETE estimates, respectively. The defect correction DS-ENO estimate is the most accurate estimate capturing the error across the shock very well. The defect correction k-exact estimate is also accurate and only misses the shape of the error across the shock but the magnitude is accurate. The fine grid method with and with-out the correction term perform well. The corresponding ETE

estimates also perform well, however, the magnitude of the discretization error near the shock is slightly over estimated compared to the corresponding defect correction estimates. Richardson extrapolation does not capture the shape of the discretization error.

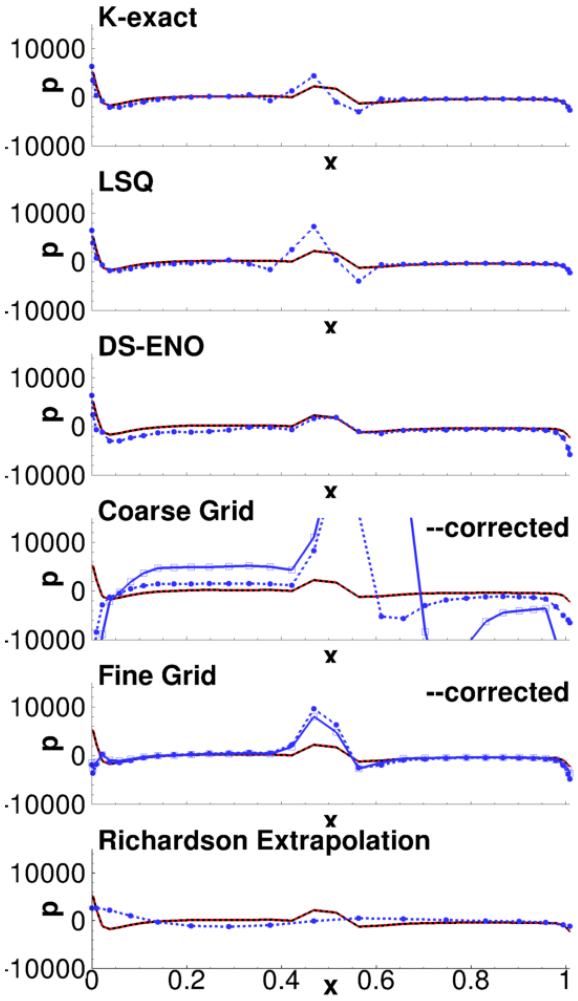


Figure 6.10: Estimated defect correction discretization error estimates for pressure along the upper side of the NACA 0012 airfoil at M=0.8 and $\alpha = 0^\circ$

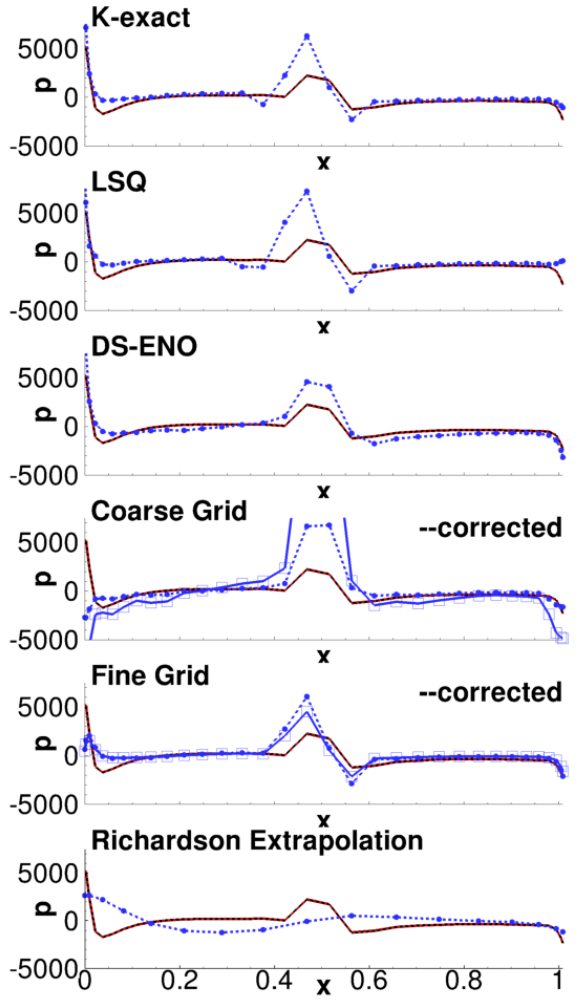


Figure 6.11: Estimated ETE discretization error estimates for pressure along the upper side of the the NACA 0012 airfoil at M=0.8 and $\alpha = 0^\circ$

For the M=0.8 airfoil at an angle of attack of 1.25 degrees, the shock is considerably stronger. The stronger shock has the affect of decreasing the accuracy of the discretization error estimates. The contour plots of pressure are shown in Figure 6.12. For all residual methods, the discretization error in pressure is over estimated and with the wrong sign. The fine grid method with the correction factor is the most accurate but in the estimate near the shock is considerable. The pressure along the upper surface of the airfoil is shown in Figures 6.13 and 6.14 for defect correction and ETEs.

Comparable to the contour plots, all residual methods over estimate the error near the shock. All methods, however, are accurate over the rest of the airfoil. Richardson extrapolation can be considered accurate, but follows a similar trend as seen for other cases which is a smooth discretization error estimate and does not capture the peak errors well.

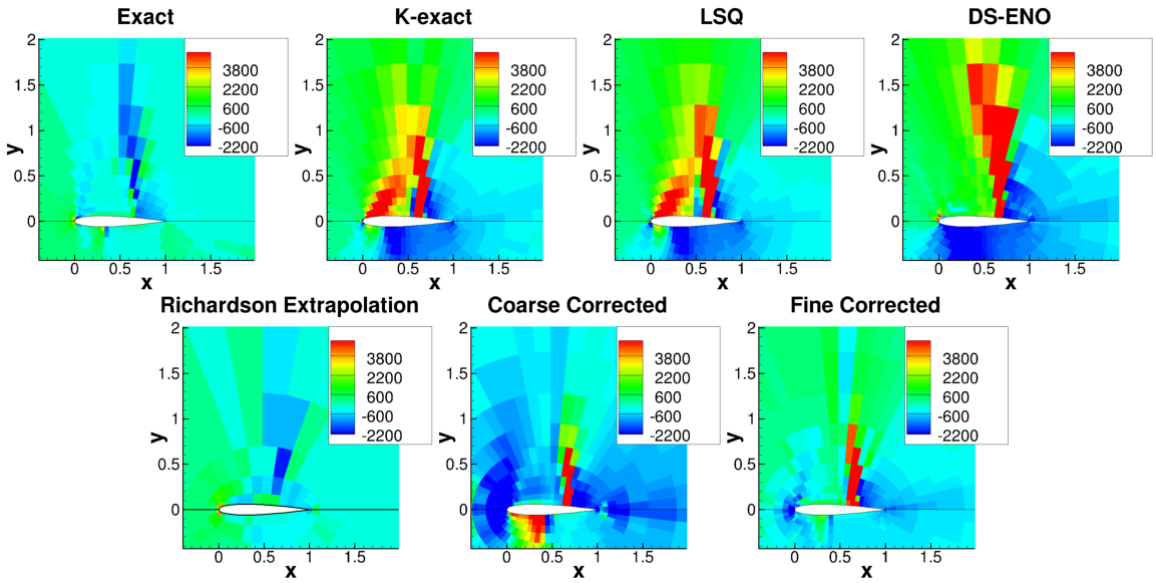


Figure 6.12: Estimated defect correction discretization error estimates for pressure for the NACA 0012 airfoil at M=0.8 and $\alpha = 1.25^{\circ}$

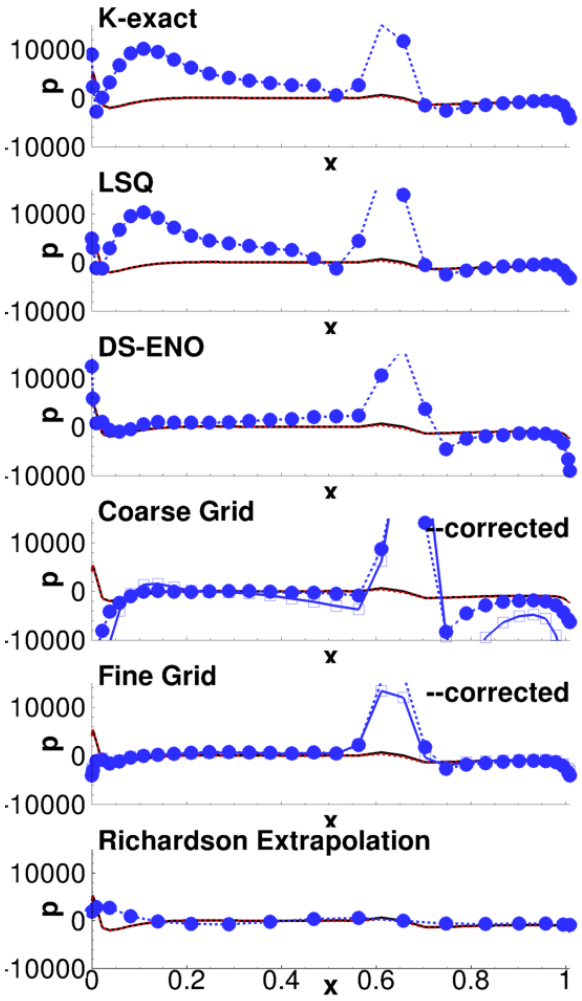


Figure 6.13: Estimated defect correction discretization error estimates for pressure along the upper side of the NACA 0012 airfoil at M=0.8 and $\alpha = 1.25^\circ$

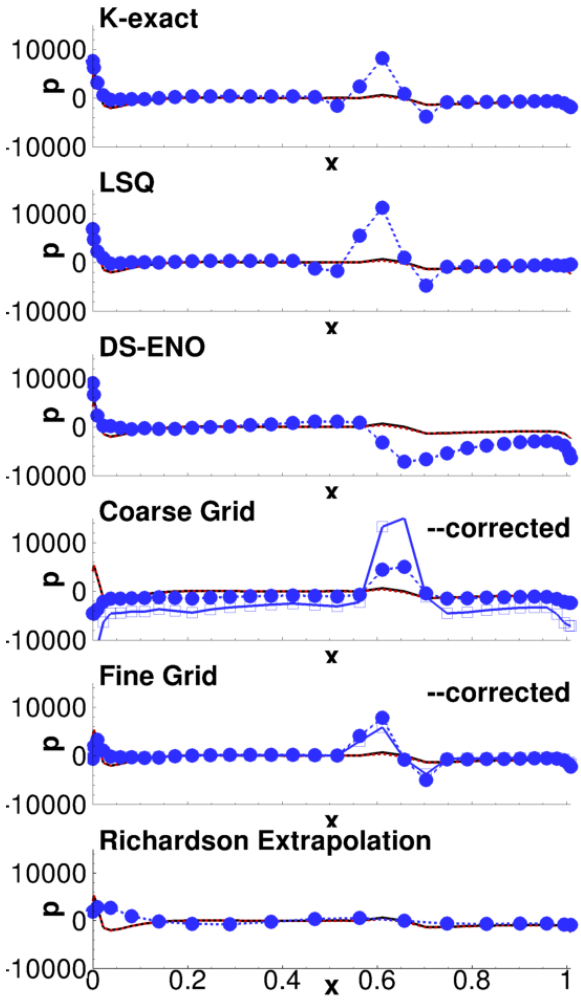


Figure 6.14: Estimated ETE discretization error estimates for pressure along the upper side of the the NACA 0012 airfoil at M=0.8 and $\alpha = 1.25^\circ$

6.7.2 Supersonic Inlet

The supersonic inlet has a free stream Mach number of four. The truncation error estimates are shown in Figure 6.15. The k-exact method compares well to the exact truncation error but is missing the spikes near the reflected shock. The LSQ, DS-ENO, and fine grid corrected methods perform reasonably well. The DS-ENO method underestimates the truncation error more than the other single grid methods and the fine grid correcte method. The coarse grid methods underestimate the truncation error. This is primary because the estimate truncation error is distributed across more cells so the truncation error peaks are smeared with less shock resolution.

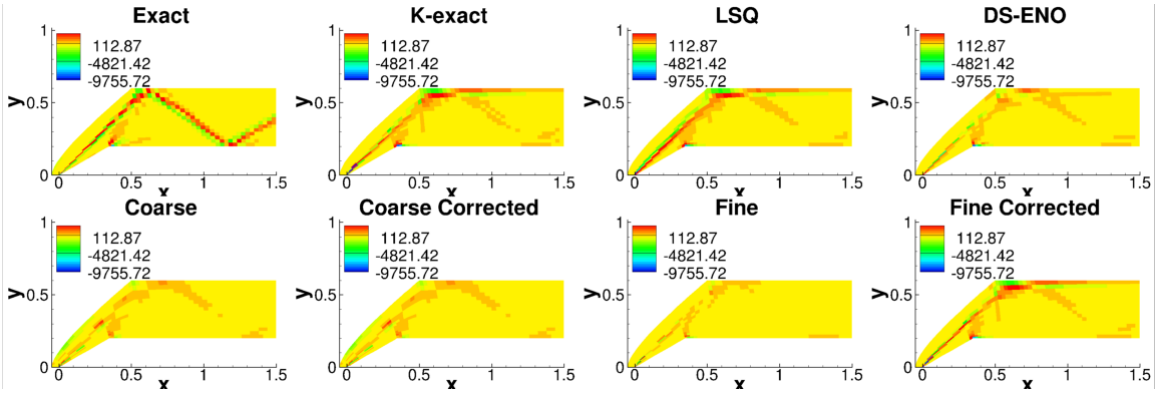


Figure 6.15: Estimated truncation error for the mass equation for the supersonic inlet at M=4.0

The resulting defect correction error estimates for pressure are shown in Figure 6.16. The best estimate is the k-exact method followed closely by the LSQ and fine grid method with the correction term. Richardson extrapolation underestimates the peak discretization error near the shocks. The coarse grid corrected method barely resolves the shock and the error estimate near the reflected shock has the wrong sign.

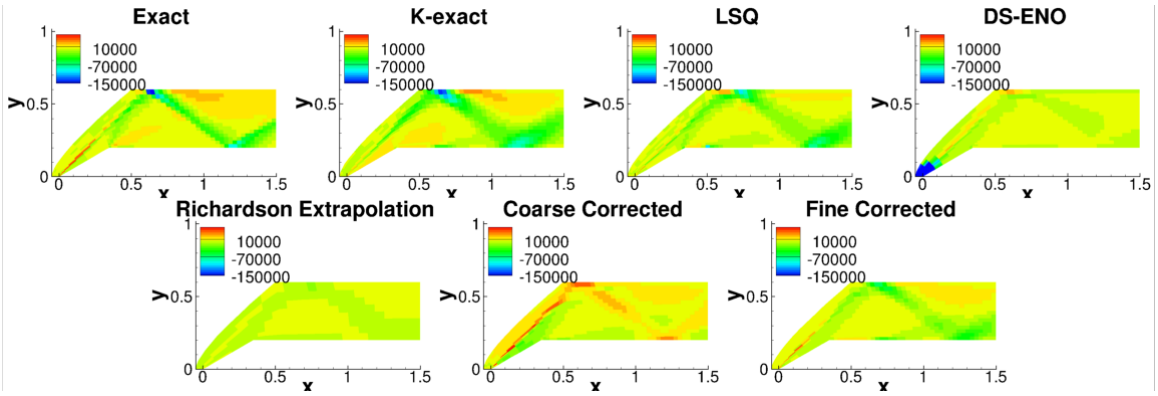


Figure 6.16: Estimated defect correction discretization error estimates for pressure for the supersonic inlet at M=4.0

The discretization error estimates for the outflow pressure profile are compared in Figures 6.17 and 6.18 for defect correction and ETEs, respectively. The most accurate estimate is the defect correction k-exact method followed by the ETE k-exact method (which has a slightly offset peak). The fine grid method with the correction factor can be considered accurate but underestimates the peak error and the location. Richardson extrapolation underestimates the discretization error significantly.

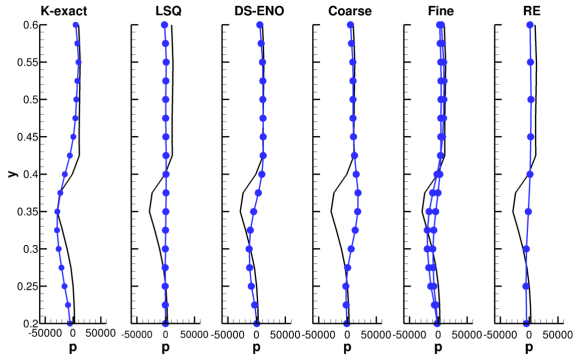


Figure 6.17: Estimated defect correction discretization error estimates for the outflow pressure for the supersonic inlet at M=4.0

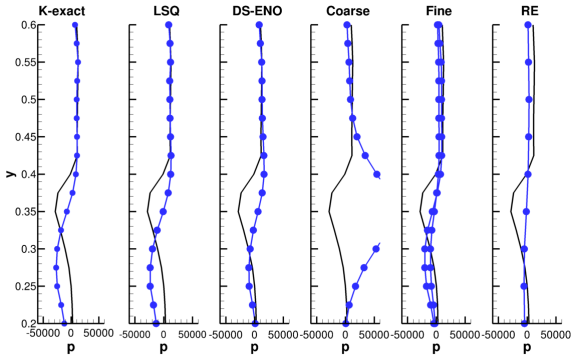


Figure 6.18: Estimated ETE discretization error estimates for the outflow pressure for the supersonic inlet at M=4.0

6.7.3 JCEAP Cone

The truncation error estimates are shown in Figure 6.19 for the hypersonic cone. The truncation error estimates can be considered accurate for the k-exact, LSQ, DS-ENO, and fine grid method with the correction factor; however, the resulting defect correction error estimates shown in Figure 6.20 are not. The fine grid method with the correction factor is the only discretization error estimate which is accurate. The single grid methods result in a large area of high magnitude error near the shock. Richardson extrapolation results in larger error near the shock but underestimates the error in the rest of the domain.

Figures 6.21 and 6.22 show the pressure along the cone. The most accurate estimate is the defect correction fine grid corrected method. The k-exact defect correction method is fairly accurate away from the nose; however, the discretization error is significantly overestimated near the nose. Similar to the other applications, Richardson extrapolation is fairly smooth and does not capture the peak discretization error near the nose of the cone. Richardson extrapolation, however, is considerably more accurate over the rest of the cone compared to the residual-based methods. The accuracy of the residual-based methods has degraded with increasing shock strength. Spurious oscillations in the reconstruction introduce larger truncation error peaks than what are normally present which are then propagated through the rest of the domain. The only method which has reliably estimated the discretization error (while not always the most accurate) is the fine grid method with the correction factor. Although, without the correction factor, the estimates are still considerably reliable.

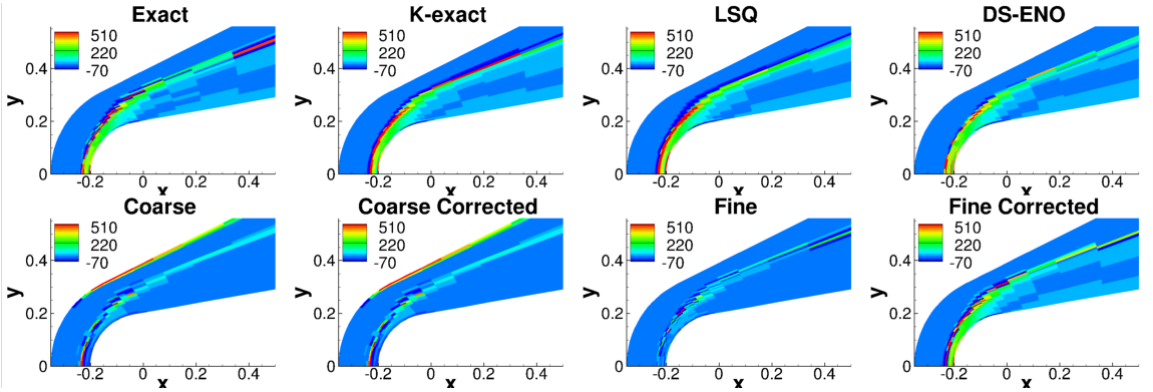


Figure 6.19: Estimated truncation error for the mass equation for the JCEAP cone at M=8.0

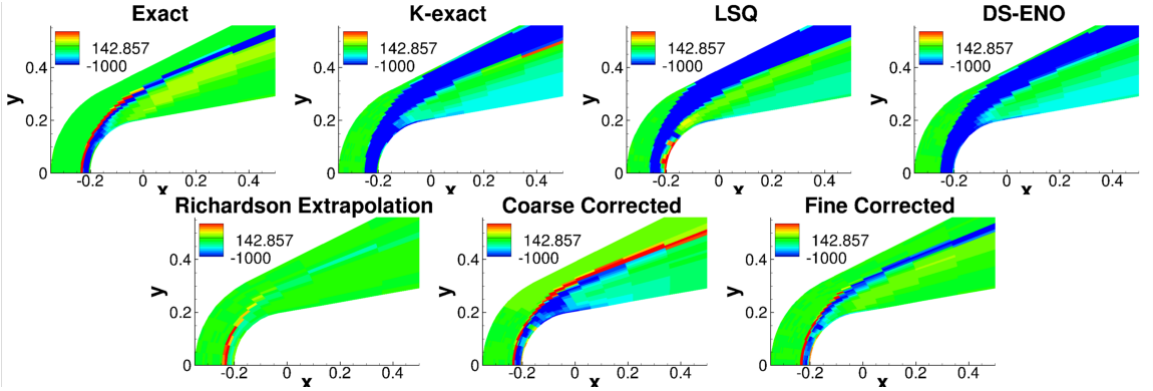


Figure 6.20: Estimated defect correction discretization error estimates for the JCEAP cone at M=8.0

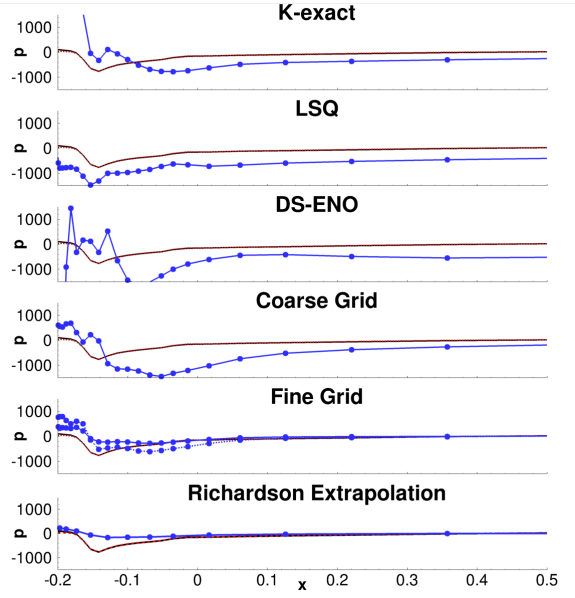


Figure 6.21: Estimated defect correction discretization error estimates for the wall pressure for the JCEAP cone at M=8.0

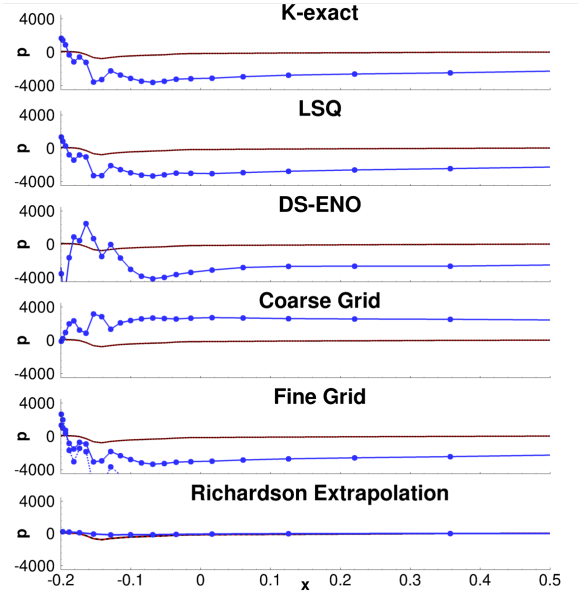


Figure 6.22: Estimated ETE discretization error estimates for the wall pressure for the JCEAP cone at M=8.0

6.8 Conclusions

Residual-based discretization error estimators were evaluated for six Euler applications and the results were compared to Richardson extrapolation. The ETEs were shown to offer a significant advantage with regards to cost of computing an estimate for all cases. The linearized system is solved in a few iterations with a relative cost which can be considered free (a few seconds computational time compared to nine minutes for a defect correction estimate). The use of first- and second-order Jacobians in the ETE were also compared. There was essential no difference between the two resulting discretization error estimates. This makes implementing the ETEs significantly easier as the first-order Jacobians are easier to derive and are often readily available if the code employs an implicit scheme.

The defect correction discretization error estimates are more accurate than the ETEs because the full nonlinear system is solved. There were a few results where the ETEs were more accurate than the defect correction estimates; however, this is more than likely due to error cancelation and is application specific. For general discretization error estimation, the ETEs offer a significant time advantage for a small cost in accuracy over defect correction. For redundancy, final design discretization error, or ease of implementation, defect correction can be used.

The most accurate truncation error estimator was the k-exact method; however, the accuracy of the error estimates significantly degraded as the shock strength of the problem increased. The most

reliable truncation error estimator (most consistent results across all cases) was the fine grid method with the correction factor although it was not always the most accurate. The resulting discretization error estimates were not the most accurate for smooth problems, but for the supersonic inlet and JCEAP cone, the discretization error estimates were the only accurate estimates whereas the single grid methods tended to suffer from spurious oscillations. The coarse grid methods did not perform well. The estimates suffered from accuracy even for smooth problems often with significant errors in the discretization error estimates.

Bibliography

[1] C. J. Roy, "Review of discretization error estimators in scientific computing," AIAA Paper 2010-126, 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 4-7, 2010, 2010.

[2] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[3] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 210, pp. pp. 307–357, 1911.

[4] P. J. Roache and P. M. Knupp, "Completed richardson extrapolation," *Communications in Numerical Methods in Engineering*, vol. 9, no. 5, pp. 365–374, 1993.

[5] S. A. Richards, "Completed richardson extrapolation in space and time," *Communications in Numerical Methods in Engineering*, vol. 13, pp. 573–582, 1997.

[6] E. Fehlberg, "Low-order classical runge-kutta formulas with step size control and their application to some heat transfer problems," *NACA Technical Report 315*, 1969.

[7] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 337–357, 1987.

[8] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates, part 2: Error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1365–1382, 1992.

[9] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[10] H. J. Stetter, "The defect correction principle and discretization methods," *Numerische Mathematik*, vol. 29, pp. 425–443, 1978.

[11] R. D. Skeel, "Thirteen ways to estimate global error," *Numerische Mathematik*, vol. 48, pp. 1–20, 1986.

[12] A. Naumovich, M. Foerster, and R. Dwight, "Algebraic multigrid within defect correction for the linearized euler equations," *Numerical Linear Algebra with Applications*, vol. 17, pp. 307–324, 2009.

[13] X. D. Zhang, D. Pelletier, and J.-Y. Trepanier, "Verification of error estimators for the euler equations," Tech. Rep. AIAA-2000-1001, 2000.

[14] Y. Qin and T. I.-P. Shih, "A discrete transport equation for error estimation in cfd," Tech. Rep. AIAA-2002-0906, 2002.

[15] I. Celik and G. Hu, "Single grid error estimation using error transport equations," *J of Fluid Eng*, vol. 126, pp. 778–790, 2004.

[16] Y. Qin, K. Chi, and T. I.-P. Shih, "Modeling the residual in error-transport equations for estimating grid-induced errors in cfd solutions," Tech. Rep. AIAA-2006-892, 2006.

[17] T. I.-P. Shih and B. R. Williams, "Development and evaluation of an a posteriori method for estimating and correcting grid-indeuced errors in solutions of the navier-stokes equations," Tech. Rep. AIAA-2009-1499, 2009.

[18] P. Cavallo and N. Sinha, "An error transport equation with practical applications," Tech. Rep. AIAA-2007-4092, 2007.

[19] P. Cavallo, N. Sinha, and M. R. O'Gara, "Viscous error transport equation for error quantification of turbulent flows," *AIAA-2008-3851*, 2008.

[20] I. Babuska and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal of Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[21] I. Babuska and W. C. Rheinboldt, "A-posteriori error estimates for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[22] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[23] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from pde approximations," *SIAM Review*, vol. 42, no. 2, pp. 247–264, 2000.

[24] T. S. Phillps and C. J. Roy, "Residual methods for discretization error estimation," *AIAA-2011-3870*, 2011.

[25] J. W. B. J. A. F. Hittinger, J. M. Connors, and C. S. Woodward, "Numerical error estimation for nonlinear hyperbolic pdes via nonlinear error transport," *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, pp. 1–15, 2012.

[26] T. S. Phillips and C. J. Roy, "A new extrapolation-based uncertainty estimator for computational fluid dynamics," *AIAA-2013-0260*, 2013.

[27] T. S. Phillips and C. J. Roy, "Richardson extrapolation-based discretization uncertainty estimation for computational fluid dynamics," *J of Fluids Eng, Accepted for publication*, 2014.

[28] C. J. Roy, "Strategies for driving mesh adaptation in cfd (invited)," Tech. Rep. AIAA-2009-1302, 2009.

[29] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[30] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, pp. 233–260, 1988.

[31] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, no. 1, pp. 40 – 69, 2002.

[32] T. J. Barth, "Higher order solution of the euler equations on unstructured grids using quadratic reconstruction," Tech. Rep. AIAA-90-0013, 1990. Initial development of K-exact method.

[33] T. J. Barth, "Recent developments in higher order k-exact reconstruction on unstructured meshes," Tech. Rep. AIAA-93-0668, 1993. Improvements to the K-exact method with work towards unstructured grids and comparison to other competing technologies.

[34] A. G. Godfrey, C. R. Mitchell, and R. W. Walters, "Practical aspects of spatially high-order accurate methods," *AIAA Journal*, vol. 31, no. 9, pp. 1634–1642, 1993. More efficient ENO scheme compared to k-exact and standard 2D standard ENO scheme – DS-ENO.

[35] A. Harten and S. Osher, "Uniformly high-order accurate nonoscillatory schemes i," *SIAM Journal of Numerical Analysis*, vol. 24, pp. 279–309, 1987. First development of ENO scheme-1D.

[36] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Finite volume solution reconstruction methods for truncation error estimation," *In Preparation*, 2014.

[37] S. R. Fulton, "On the accuracy of multigrid truncation error estimates," *Electronic Transactions on Numerical Analysis*, vol. 15, pp. 29–37, 2003.

[38] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[39] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Residual-based discretization error estimators for cfd part 1: Numerical benchmarks," *In Preparation*, 2014.

Chapter 7

Discussion and Conclusions

Residual-based discretization error estimation methods offer significant advantages over the more commonly used Richardson extrapolation for discretization error estimation. The critical component for discretization error estimation accuracy was identified to be truncation error estimation accuracy. Various truncation error estimation methods were identified or developed based on past work and the minimum requirements for asymptotically accurate estimates were identified. Asymptotically accurate means that the estimated truncation error approaches the exact trunction error as the computational grid is refined. If a truncation error estimation method is not asymptotically accurate, the discretization error estimate will also not be asymptotically accurate. Identification of asymptotically accurate methods is missing from past work and is a very important concept required for the success of residual-based discretization error estimation. The most successful method is the k-exact method. This method was develoed based on past solution reconstruction work by Barth[26,27] and modified to reduce the stencil size. The k-exact method requires only one grid level and the smallest stencil size of any of the single grid truncation error estimation methods.

The accuracy of the truncation error estimate was also extended to boundary conditions. Two different implementations were investigated, scheme consistent and scheme inconsistent, and the affects on truncation error were evaluated. The scheme consistent methods resulted in smooth truncation error and better truncation error estimation accuracy near the boundary than the scheme inconsistent methods. The scheme inconsistent boundary conditions resulted in lower discretization error because the applications studied resulted in larger truncation error of a different sign which resulted in more error cancelation through the domain. However, the scheme inconsistent boundary condition truncation error reduced at only a first-order rate and was not asymptotic. The lack of asymptoticness resulted in significantly reduced Richardson extrapolation and residual-based discretization error estimation accuracy despite the marginally smaller (and possibly application specific) discretization error. As the goal of this research is improved discretization error estimation accuracy, the scheme consistent boundary conditions are recommened.

The slip wall boundary conditions has a different governing equation for the flux at the boundary. For the single grid truncation error estimation methods, this modification to the governing

equations must be taken into consideration for accurate truncation error estimation. The resulting accuracy of the discretization error estimate is significantly reduced by sign and magnitude.

The residual-based discretization error estimation methods were evaluated for a set of compressible Euler flow applications ranging from subsonic to hypersonic. The residual-based methods were compared to Richardson extrapolation. Richardson extrapolation tended to underestimate the magnitude of the error, the location of the peak error, and in some cases the sign of the error. The largest peaks in the discretization error were significantly smoothed for all applications. For the NACA 0012 airfoils, the discretization error along the leading edge of the airfoil was significant and Richardson extrapolation consistently estimated the incorrect sign and magnitude of the error.

The residual-based discretization error methods had varied results and tended to have the opposite affect on the discretization error (i.e., tended to magnify the peak discretization error). The single grid methods, specifically the k-exact method, performed the best overall. For stronger discontinuities; however, the accuracy of the discretization error estimates degraded due non-smooth reconstructions. For the $M = 8$ hypersonic cone, the single grid methods did poorly significantly overestimating the error near the shock and the leading edge of the nose. The fine grid method with the correction factor performed consistently well in that it did not fail terribly but it was often not the best estimator. The overall shape of the discretization error compared well but the peaks tended to be magnified. The coarse grid with or without the correction term performed poorly and should not be used. The coarse grid methods tended to smooth out the truncation error near peaks due to the coarser computational grid and resulted in generally inaccurate estimates of the magnitude and location of the discretization error with tendancies to magnify the magnitude of the error.

Defect correction and the ETE discretization error estimates had similar accuracy. The ETEs tended to slightly under estimate the peak errors due to non-linearities in the implementation. The ETEs approach the defect correction estimate at the rate of ε_h^2. The accuracy of the ETE estimates compared well with descrepancies showing up where the local discretization error was large (i.e. near shocks and large gradients). The cost of solving the ETEs is essential free because of the linearity and is significantly cheaper than the cost of the defect correction estimate. It was found that the first-order Jacobian was more than sufficient for solving ETEs for second-order MUSCL and was also moderately more stable than the second-order Jacobian.

The recommended truncation error estimation method is the k-exact method with the corrected fine grid method as a second recommendation. It is important to note that for the single grid method requires correct implementation of the continuous equations including adjustments for boundary conditions or source terms. Likewise, the fine grid method implicitily incorporates these modifications but requires an assumption of order of accuracy. The ETEs are recommended with a first-order Jacobian for general discretization error estimation at the cost of a more code intrusive procedure and slightly reduced accuracy near areas of large discretization error. Defect correction is easier to implement and has improved accuracy but the cost is more significant (although moderately cheaper than the primal problem due to a better initial condition).

7.1 Extensions and Recommended Future Work

The applications of interest were steady-state Euler and Navier-stokes equations for structured grids. All procedures investigated are extendable to unsteady flow and unstructured grids. The exact same procedures can be used for unstructured grids. The truncation error for unstructured grids tends to be more noisy; therefore, the same conclusions regarding truncation error estimation accuracy can't be made. Extensions to steady-flow requires modifications to the reconstruction method. A temporal stencil must also be included; however, the temporal terms are stored at discrete time intervals similar to a finite difference method. For example, an unsteady one-dimensional, first-order polynomial would take the form $p(x,t) = a(t) + b(t)x$ where $a(t) = a_1 + a_2 t$, $b(t) = b_1 + b_2 t$, which, expanded, takes the general form $p(x,t) = a_1 + a_2 t + b_1 x + b_2 xt$ with the constraints

$$u_i^n = \frac{1}{V}\int_V p(x,t)dx \tag{7.1}$$

where i is the cell index and n is the temporal index. The only additional extension would require including the temporal term in the continuous equations.

The recommended truncation error estimation method is the k-exact method. A few applications had locally better results for the DS-ENO method and the adaptive stencil offers some advantages near discontinuities. A hybrid scheme between the k-exact and DS-ENO method is proposed but not tested for future work. The minimum stencil size required for the k-exact method for the upwinded MUSCL extrapolation is three. The minmod flux limiter is used to identify areas of the flow which are not smooth and requires a stencil size of three. For cell i, the minmod limiter is

$$\psi_i = min\left(max\left(\frac{u_{i+1}-u_i}{u_i-u_{i-1}}, 0\right), 1\right). \tag{7.2}$$

where $0 \leq \psi \leq 1$ and u is a solution variable. Two smoothness indicators are proposed. The smoothness indicator is based on the limiter values for all cells in the k-exact stencil. The first is the average limiter value

$$\bar{\psi}_i = \sum_{j=1}^{N_{st}} \frac{\psi_j}{N_{st}} \tag{7.3}$$

and the second is the the minimum value of ψ_i for all cells in the k-exact stencil resulting in a more conservative weighting

$$\bar{\psi}_i = min(\psi_j),\ j = 1\ to\ j = N_{st}. \tag{7.4}$$

The final truncation error estimation method proposed combines the k-exact estimate of order $k_1 = 2$ and the DS-ENO estimate of order $k_2 = 2$ or $k_2 = 3$. The final truncation error estimate is a weighted combination between the two truncation error estimation methods control by C_1

$$\tau_h(I_h^k u_h) = I^h L(I_h^{k_1} u_h) C_1 + I^h L(I_h^{k_2} u_h)(1 - C_1). \tag{7.5}$$

The composite truncation error estimate is estimated using a linear weighting where $C_1 = \bar{\psi}_i$. The reasons for the recommendation are itemized below.

- minmod flux limiter, same stencil size as $k = 2$ reconstruction, simple to implement, and range ($0 \leq \psi \leq 1$) is suitable for linear scaling between two quantities
- k-exact method, is the most accurate truncation error estimation method and considerably outperforms other methods especially for smooth problems
- $k_1 = 2$, the minimum order of polynomial reconstruction is two for both Euler and Navier-Stokes equations
- DS-ENO method, adaptive stencil useful for reconstruction near discontinuities and is accurate for convection dominated areas of flow
- $k_2 = 2$, if minmod limiter is activated then order of accuracy is limited towards first-order accuracy where $k_2 = 2$ is accurate
- $k_2 = 3$, is accurate for second-order MUSCL but activates adaptive stencil and would maintain full truncation error estimate accuracy regardless of minmod value
- linear $\tau_h(\cdot)$ weighting, simple implementation and switching

Bibliography

[1] C. J. Roy, "Review of discretization error estimators in scientific computing," AIAA Paper 2010-126, 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 4-7, 2010, 2010.

[2] C. J. Roy, "Strategies for driving mesh adaptation in cfd (invited)," Tech. Rep. AIAA-2009-1302, 2009.

[3] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, New York, 2000.

[4] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[5] H. J. Stetter, "The defect correction principle and discretization methods," *Numerische Mathematik*, vol. 29, pp. 425–443, 1978.

[6] M. Kurzen, T. S. Phillips, and C. J. Roy, "Method of nearby problems for generating exact solutions to 1d unsteady and 2d steady problems," Tech. Rep. AIAA-2009-3652, 2009.

[7] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[8] A. Naumovich, M. Foerster, and R. Dwight, "Algebraic multigrid within defect correction for the linearized euler equations," *Numerical Linear Algebra with Applications*, vol. 17, pp. 307–324, 2009.

[9] X. D. Zhang, J. Y. Trepanier, and R. Camarero, "A posteriori error estimation for nite-volume solutions of hyperbolic conservation laws," *Computer Methods in Applied Mechanics and Engineering*, vol. 185, pp. 1–19, 2000.

[10] X. D. Zhang, D. Pelletier, and J.-Y. Trepanier, "Verification of error estimators for the euler equations," Tech. Rep. AIAA-2000-1001, 2000.

[11] I. Celik and G. Hu, "Single grid error estimation using error transport equations," *J of Fluid Eng*, vol. 126, pp. 778–790, 2004.

[12] T. S. Phillps and C. J. Roy, "Residual methods for discretization error estimation," *AIAA-2011-3870*, 2011.

[13] Y. Qin and T. I.-P. Shih, "A method for estimating grid-induced errors in finite-difference and finite-volume methods," Tech. Rep. AIAA-2003-845, 2003.

[14] Y. Qin, P. S. Keller, R. L. Sun, E. C. Hernandez, C.-Y. Perng, N. Trigui, Z. Han, F. Z. Shen, T. Shieh, and T. I.-P. Shih, "Estimating grid-induced errors in cfd by discrete-error-transport equations," Tech. Rep. AIAA-2004-656, 2004.

[15] Y. Qin, K. Chi, and T. I.-P. Shih, "Modeling the residual in error-transport equations for estimating grid-induced errors in cfd solutions," Tech. Rep. AIAA-2006-892, 2006.

[16] P. Cavallo and N. Sinha, "An error transport equation with practical applications," Tech. Rep. AIAA-2007-4092, 2007.

[17] P. Cavallo and N. Sinha, "A posteriori error estimation for nite-volume solutions of hyperbolic conservation laws," *Journal of Aircraft*, vol. 44, no. 6, 2007.

[18] P. Cavallo, N. Sinha, and M. R. O'Gara, "Viscous error transport equation for error quantification of turbulent flows," *AIAA-2008-3851*, 2008.

[19] J. W. B. J. A. F. Hittinger, J. M. Connors, and C. S. Woodward, "Numerical error estimation for nonlinear hyperbolic pdes via nonlinear error transport," *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, pp. 1–15, 2012.

[20] Y. Qin and T. I.-P. Shih, "A discrete transport equation for error estimation in cfd," Tech. Rep. AIAA-2002-0906, 2002.

[21] T. I.-P. Shih and B. R. Williams, "Development and evaluation of an a posteriori method for estimating and correcting grid-indeuced errors in solutions of the navier-stokes equations," Tech. Rep. AIAA-2009-1499, 2009.

[22] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, no. 1, pp. 40 – 69, 2002.

[23] S. R. Fulton, "On the accuracy of multigrid truncation error estimates," *Electronic Transactions on Numerical Analysis*, vol. 15, pp. 29–37, 2003.

[24] F. Fraysse, J. de Vicente, and E. Valero, "The estimation of truncation error by τ-estimation revisited," *Journal of Computational Physics*, vol. 231, pp. 3457–3482, 2012.

[25] C. Hirsch, *Numerical Computation of Internal and External Flows, Volume 2: COmputational Methods for Inviscid and Viscous FLows*. John Wiley and Sons, 1984.

[26] T. J. Barth, "Higher order solution of the euler equations on unstructured grids using quadratic reconstruction," Tech. Rep. AIAA-90-0013, 1990. Initial development of K-exact method.

[27] T. J. Barth, "Recent developments in higher order k-exact reconstruction on unstructured meshes," Tech. Rep. AIAA-93-0668, 1993. Improvements to the K-exact method with work towards unstructured grids and comparison to other competing technologies.

[28] C. R. Mitchell and R. W. Walters, "K-exact reconstruction for the navier stokes equations on arbitrary grids," Tech. Rep. AIAA-93-0536, 1993. Apply k-exact to structured and unstructred grids for Euler and Navier-Stokes equations.

[29] A. Harten and S. Osher, "Uniformly high-order accurate nonoscillatory schemes i," *SIAM Journal of Numerical Analysis*, vol. 24, pp. 279–309, 1987. First development of ENO scheme-1D.

[30] C.-W. Shu, "Numerical experiments on the accuracy of eno and modified eno schemes," *Journal of Scientific Computing*, vol. 5, no. 2, pp. 127–147, 1990. Modification to the ENO scheme-1D.

[31] A. G. Godfrey, C. R. Mitchell, and R. W. Walters, "Practical aspects of spatially high-order accurate methods," *AIAA Journal*, vol. 31, no. 9, pp. 1634–1642, 1993. More efficient ENO scheme compared to k-exact and standard 2D standard ENO scheme – DS-ENO.

[32] X. D. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *Journal of Computational Physics*, vol. 115, 1994. Initial development of WENO scheme.

[33] Y. Shen and G. Zha, "Improved seventh-order weno scheme," Tech. Rep. AIAA-2010-1451, 2010. 7th order accurate WENO scheme.

[34] W. Li and Y.-X. Ren, "High-order k-exact weno finite volume schemes for solving gas dynamic euler equations on unstructured grids," *International Journal for Numerical Methods in Fluids*, vol. 70, pp. 742–763, 2012. Combination of k-exact and WENO for simplicity and stabiltiy.

[35] V. Pereyra, "On improving an approximate solution of a functional by deferred corrections," *Numerische Mathematik*, vol. 8, pp. 376–391, 1965.

[36] R. D. Skeel, "Thirteen ways to estimate global error," *Numerische Mathematik*, vol. 48, pp. 1–20, 1986.

[37] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, pp. 233–260, 1988.

[38] T. S. Phillips and C. J. Roy, "A new extrapolation-based uncertainty estimator for computational fluid dynamics," *AIAA-2013-0260*, 2013.

[39] C. W. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, pp. 197–205, 1960. http://www.digizeitschriften.de/dms/img/?PPN=GDZPPN001163442.

[40] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Dok. Akad. Nauk SSSR*, vol. 4, pp. 240–243, 1963. Found reference: Gerstner and Briebel, Numerical Algorithms 18 (1998) 209-232.

[41] J. Burkardt, "Clenshaw curtis quadrature rules." `http://people.sc.fsu.edu/~jburkardt/m_src/ccn_rule/ccn_rule.html`, 2010.

[42] J. M. Derlaga, T. S. Phillips, and C. J. Roy, "Sensei computational fluid dynamics code: A case study in modern fortran software development," Tech. Rep. AIAA-2013-2450, 2013.

[43] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 2002.

[44] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[45] P. J. Roache, *Fundamentals of Verification and Validation*. Hermosa Publishers, Albuquerque, NM., 2009.

[46] J. W. Banks, T. Aslam, and W. J. Rider, "On sub-linear convergence for linearly degenerate waves in capturing schemes," *Journal of Computational Physics*, 2008.

[47] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 210, pp. pp. 307–357, 1911.

[48] P. J. Roache and P. M. Knupp, "Completed richardson extrapolation," *Communications in Numerical Methods in Engineering*, vol. 9, no. 5, pp. 365–374, 1993.

[49] S. A. Richards, "Completed richardson extrapolation in space and time," *Communications in Numerical Methods in Engineering*, vol. 13, pp. 573–582, 1997.

[50] E. Fehlberg, "Low-order classical runge-kutta formulas with step size control and their application to some heat transfer problems," *NACA Technical Report 315*, 1969.

[51] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 337–357, 1987.

[52] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates, part 2: Error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1365–1382, 1992.

[53] I. Babuska and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal of Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[54] I. Babuska and W. C. Rheinboldt, "A-posteriori error estimates for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[55] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[56] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from pde approximations," *SIAM Review*, vol. 42, no. 2, pp. 247–264, 2000.

[57] A. Choudhary and C. J. Roy, "Structured mesh r-refinement using truncation error equidistribution for 1d and 2d euler problems," *AIAA-2013-2444*, 2013.

[58] B. van Leer, "Flux vector splitting for the euler equations," in *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, 1982.

[59] C. W. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, pp. 1–15, 1960.

[60] C. Ollivier-Gooch, A. Nejat, and K. Michalak, "Obtaining and verifying higher-order unstructured finite volume solutions to the euler equations," *AIAA Journal*, vol. 47, no. 9, pp. 2105–2120, 2009.

[61] T. S. Phillips, J. M. Derlaga, and C. J. Roy, "Numerical benchmark solutions for laminar and turbulent flows," *AIAA-2012-3074*, 2012.

[62] W. L. Oberkampf and T. G. Trucano, "Verification and validation benchmarks," *Nuclear Engineering and Design*, vol. 238, no. 3, pp. 716–743.

[63] C. J. Roy, "Review of code and solution verification procedures for computational simulation," *Journal of Computational Physics*, vol. 205, no. 1, pp. 131 – 156, 2005.

[64] Roy, "Grid convergence error analysis for mixed-order numerical schemes," *AIAA Journal*, vol. 41, pp. 595–604, April 2003.

[65] J. C. Vassberg and A. Jameson, "In pursuit of grid convergence for two-dimensional euler solutions," *Journal of Aircraft*, vol. 47, no. 4, pp. 1152–1166, 2010.

[66] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Residual-based discretization error estimators for cfd part 2: Error estimates," *In Preparation*, 2014.

[67] T. S. Phillips and C. J. Roy, "Richardson extrapolation-based discretization uncertainty estimation for computational fluid dynamics," *J of Fluids Eng, Accepted for publication*, 2014.

[68] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Finite volume solution reconstruction methods for truncation error estimation," *In Preparation*, 2014.

[69] T. S. Phillips, J. M. Derlaga, C. J. Roy, and J. Borggaard, "Residual-based discretization error estimators for cfd part 1: Numerical benchmarks," *In Preparation*, 2014.