SANDIA REPORT
SAND2011-7118
Unlimited Release
Printed October, 2011

Discretization Error Estimation and Exact Solution Generation using the Method of Nearby Problems

Christopher J. Roy, Tyrone S. Phillips, Matthew J. Kurzen
Virginia Tech

Andrew J. Sinclair and Anil Raju
Auburn University

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-7118
Unlimited Release
Printed October 2011

Discretization Error Estimation and Exact Solution Generation using the Method of Nearby Problems

Christopher J. Roy
Tyrone S. Phillips
Matthew J. Kurzen

Aerospace and Ocean Engineering Department
Virginia Tech
Blacksburg, Virginia 24061

Andrew J. Sinclair, Anil Raju

Aerospace Engineering Department
Auburn University
Auburn, Alabama 36849

Abstract

The Method of Nearby Problems (MNP), a form of defect correction, is examined as a method for generating exact solutions to partial differential equations and as a discretization error estimator. For generating exact solutions, four-dimensional spline fitting procedures were developed and implemented into a MATLAB code for generating spline fits on structured domains with arbitrary levels of continuity between spline zones. For discretization error estimation, MNP/defect correction only requires a single additional numerical solution on the same grid (as compared to Richardson extrapolation which requires additional numerical solutions on systematically-refined grids). When used for error estimation, it was found that continuity between spline zones was not required. A number of cases were examined including 1D and 2D Burgers' equation, the 2D compressible Euler equations, and the 2D incompressible Navier-Stokes equations. The discretization error estimation results compared favorably to Richardson extrapolation and had the advantage of only requiring a single grid to be generated.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

CFD	Computational Fluid Dynamics
GTEE	Generalized Truncation Error Expression
MNP	Method of Nearby Problems
PDE	Partial Differential Equations
Re	Reynolds Number

1. INTRODUCTION

High-fidelity computational simulations are playing an ever-increasing role in the design and development of engineering systems. It is thus critical to be able to quantify the uncertainty in the simulation predictions. Numerical error is an important factor in the overall uncertainty in the simulation prediction. A key component of the uncertainty is the explicit numerical error in the prediction itself. In addition, numerical error plays a more subtle role when significant numerical errors are present during the model validation phase. If too large, these numerical errors can yield "false positives" for the validation of the model. Furthermore, if large numerical errors are present during a model calibration step, then these errors will have propagated into the model.

There are three components of the numerical error: round-off error, iterative error, and discretization error.[1,2] Round-off error can be mitigated by simply using more significant digits in the computation. Iterative error can often be reduced by simply running additional iterations or monitoring convergence of the solution residuals. Discretization error is the most difficult aspect of numerical error to analyze, and is defined as the difference between the exact solution to the discretized equations and the exact solution to the original partial differential or integral equations (herein referred to collectively as the PDEs). Assuming iterative and round-off errors are negligible relative to the discretization error, the numerical solution can be used as a surrogate for the exact solution to the discrete equations when estimating the discretization error.

The main difficulty in estimating the discretization error is finding a way to estimate the exact solution to the PDEs. Exact solutions exist for only the simplest equations or simplified versions of nonlinear, coupled partial differential equations. The most common way of estimating the exact solution to the PDEs is Richardson extrapolation,[2,3] which uses numerical solutions on two or more grids to estimate the exact solution. Richardson extrapolation requires systematic refinement over the entire domain,[2] and that both of the mesh levels be in the asymptotic grid convergence regime where errors are reduced at the rate dictated by the formal order of accuracy of the method (e.g., the rate found by evaluating the truncation error). A third grid is needed to confirm that the asymptotic regime has been reached. The requirement for three grid levels (all in the asymptotic regime) is difficult to achieve both because of the total number of grid points required and the burden on the grid generation tool to generate systematically refined grids. Another promising approach for estimating discretization errors is the adjoint method.[4] While this approach can provide estimates of the discretization error in any quantity of interest, there is a great deal of overhead associated with the coding and solving of the adjoint system. This approach is thus very difficult from a code development point of view. Furthermore, adjoint methods have not yet been demonstrated for realistic engineering problems on complex geometries.

The standard methods for evaluating the efficacy of error estimators involve the use of either exact solutions or benchmark numerical solutions. For complex PDEs (e.g., the Navier-Stokes equations), there are generally only a limited number of exact solutions available. Furthermore, these exact solutions often involve significant simplifications and do not exercise the general governing equations. For example, consider the flow between moving parallel plates separated by a small gap (Couette flow). Here the velocity gradient is linear and thus the diffusion term, a second derivative of velocity, is identically zero. The use of a benchmark numerical solution or

highly-refined "truth" mesh is also problematic since the accuracy of the benchmark solution is generally unknown. In addition, assessing the rate of convergence of the numerical method is difficult without a true exact solution. A method for generating exact solutions to complex partial differential equations in realistic parameter regimes and geometries would provide a useful framework for evaluating discretization error estimators. Furthermore, such exact solutions would be extremely useful in evaluating the efficacy of numerical schemes and solution adaptation techniques.

In 2003, Roy and Hopkins[5] proposed an approach for estimating discretization error and generating exact solutions called the Method of Nearby Problems (MNP). MNP requires two numerical solutions on the same grid, thereby eliminating the problems associated with generating multiple grids that are all within the asymptotic grid convergence regime. The steps associated with applying MNP as a single-grid discretization error estimator are:

1. compute the original numerical solution on the chosen grid,
2. generate an accurate curve fit to this numerical solution, thereby providing an analytic representation of the numerical solution,
3. operate the governing partial differential equations on the curve fit from step 2 to generate small analytic source terms,
4. compute the nearby problem (original problem plus analytic source terms) on the chosen grid,
5. evaluate the exact discretization error (i.e., numerical solution minus the analytic curve fit) on the nearby problem, and
6. assume that the discretization error in the nearby problem can be used to estimate the discretization error in the original problem of interest.

The key point to this approach is that, by definition, the curve fit generated in step 2 is the exact solution to the nearby problem. If the source terms are sufficiently small, then the nearby problem is said to be sufficiently near the original problem. This technique is loosely related to the Method of Manufactured Solution for code verification[2,3]: however, in the present case, the solution must be a realistic solution (which is not a requirement for Manufactured Solutions). MNP is closely related to a type of defect correction technique known as differential correction (for more information, see the review by Skeel[6]).

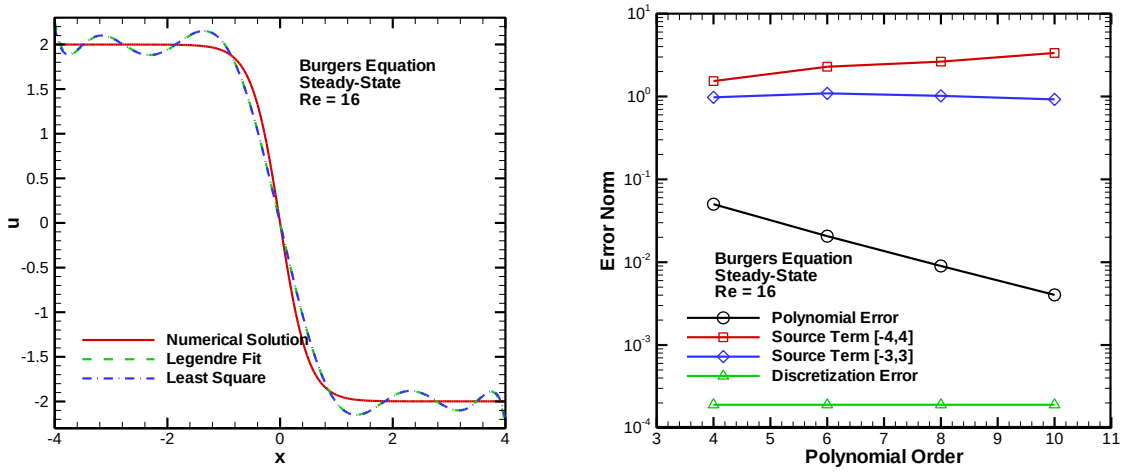
The purpose of this report is to detail the work that was performed under a five-year effort as part of the Principle Investigator's Presidential Early Career Award for Scientists and Engineers (PECASE). The initial goal of the effort was to generate exact solutions to the three-dimensional, unsteady Navier-Stokes equations in order to assess numerical methods and subgrid turbulence closure models for turbulent flows. While this goal was not achieved in its entirety, several important developments were obtained. First, a methodology and MATLAB tool were developed which allowed for the spline fitting of up to four-dimensional (3D + time) data sets on curvilinear structured grids with arbitrary levels of solution continuity. Second, a framework was developed for investigating residual based methods (defect correction, error transport equations, and adjoint methods) for discretization error estimation. Finally, the MNP methodology, a form

of defect correction, was applied to a number of 1D and 2D fluid mechanics problems in order to provide discretization error estimates using a single grid. The range of problems examined includes Burgers' equation (steady and unsteady), the Euler equations, and the incompressible Navier-Stokes equations.

2. CURVE FITTING METHODS

2.1. One-Dimensional Fits

In our previous work[7] we attempted to achieve good fits to the underlying numerical solution using global polynomials. These global polynomials tended to provide poor approximations of the numerical solution both at the boundaries and in the vicinity of sharp gradients. The poor agreement for global polynomials is demonstrated for both standard and Legendre polynomials[8] in Figure 1a for steady-state Burgers equation at a Reynolds number of 16 (see Ref. 7 for details). This lack of agreement is also evident in Figure 1b which gives L^2 norms of the MNP source term and the curve fitting error as a function of the polynomial order. While the error in the curve fit does drop slowly with increasing polynomial order, the norms of the source term increase when evaluated over the whole domain (from -4 to 4), or stay nearly constant when the boundaries are removed from the norm calculation (from -3 to 3). Ideally, the norm of the source term would drop until the polynomial error is the same order as the discretization error in the original numerical solution (also shown in Figure 1b). Lee and Junkins[9] also observed an increase in polynomial error near the boundaries using global Chebyshev polynomials. In the current work, we instead focus on spline fits, where piecewise-polynomial approximations are made by dividing the domain into a sequence of equally spaced zones with continuity constraints enforced at the zone boundaries. For curvilinear structured meshes in higher dimensions, the fits are performed in a Cartesian computational space.



a) b)

Figure 1. Global polynomial fit for Burgers' equation at a Reynolds number of 16: a) comparison of global 10th order polynomial curve fits with the underlying numerical solution and b) norm of the polynomial error and source term versus polynomial order.

2.1.1. Cubic Spline Fits

A one-dimensional cubic spline is constructed of piecewise third-order polynomials.[10] A cubic spline is twice continuously differentiable (i.e., is C^2 continuous) and depends on four parameters. It can be written as

$$S_i(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{1}$$

for

$$x \in [x_i, x_{i+1}], i = 0,..., n-1$$

and a schematic of this spline system is shown in Figure 2. The domain is broken up into *n+1* nodes (0 to *n*) and therefore *n* spline zones (S_0, S_1, …, S_{n-1}). Equation (1) is used to represent the solution over each spline zone, subject to certain constraints. The conditions that are used to construct the cubic spline polynomials are:

$$S_i(x_i) = u_i,\ i = 0,..., n$$
$$S_i(x_i) = S_{i-1}(x_i),\ i = 1,..., n$$
$$S_i'(x_i) = S_{i-1}'(x_i),\ i = 1,..., n-1$$
$$S_i''(x_i) = S_{i-1}''(x_i),\ i = 1,..., n$$

where the first condition matches the given solution values at each node. The remaining conditions provide continuity of the solution and its derivatives (up to the second derivative) at the spline boundaries. Here we find it convenient to set $S_{n-1}(x_n) = a_{n-1}$ and $S_{n-1}''(x_n) = 2c_{n-1}$. The first derivatives at end points are also specified which provides two additional conditions and thus closes the system.

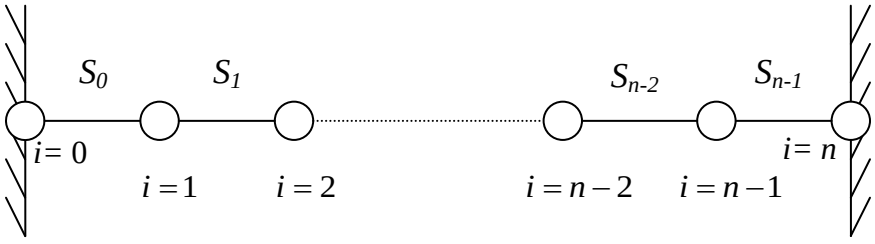


Figure 2. Schematic of the spline fitting system.

2.1.2. Hermite Spline Fits

A fifth degree Hermite spline is constructed of piecewise fifth-order polynomials.[10] This spline polynomial is given by

$$S_i(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 + e_i(x - x_i)^4 + f_i(x - x_i)^5 \tag{2}$$

for

$$x \in [x_i, x_{i+1}], i = 0,...,n-1$$

where the same spline system given in Figure 2 is used. The conditions used to construct the fifth degree Hermite spline are

$$\begin{aligned}
S_i(x_i) &= u_i,\ i = 0,...,n \\
S_i'(x_i) &= u_i',\ i = 0,...,n \\
S_i(x_i) &= S_{i-1}(x_i),\ i = 1,...,n \\
S_i'(x_i) &= S_{i-1}'(x_i),\ i = 1,...,n-1 \\
S_i''(x_i) &= S_{i-1}''(x_i),\ i = 1,...,n-1 \\
S_i'''(x_i) &= S_{i-1}'''(x_i),\ i = 1,...,n-1
\end{aligned}$$

where the first two conditions match values and derivatives of the given solution. The last four conditions match the solution value and derivatives up to the third derivative at the spline boundaries. Here we find it convenient to set $S_{n-1}(x_n) = a_{n-1}$ and $S_{n-1}'(x_n) = b_{n-1}$.

Fifth degree Hermite spline fits were used to approximate the numerical solution to Burgers equation.[11] The results for steady-state Burgers equation with a Reynolds number of 64 are presented below in Figure 3. The numerical solution appears to be approximated quite well since the fifth-order Hermite spline fit with 65 spline points is visually indistinguishable from the underlying numerical solution. Note that even though the Reynolds number for this case is larger by a factor of four than the global polynomial fit shown in Figure 1 (thus resulting in a sharper gradient), the results are significantly better with the Hermite spline fit.

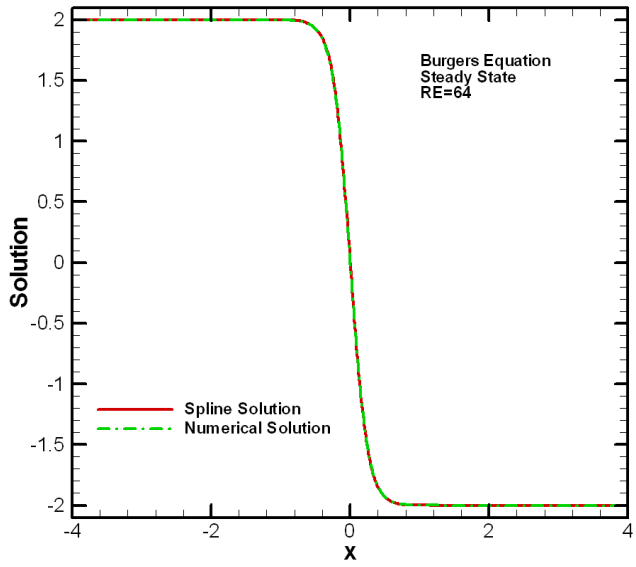


Figure 3. Fifth-order Hermite spline fit and underlying numerical solution for Burgers equation at a Reynolds number of 64 using 65 spline points.

2.1.3. Weighting Function Approach

A weighting function approach is also examined following the work of Jancaitis and Junkins.[12] The basic idea is to employ a series of overlapping local fits over the domain which are then joined together in a continuous manner using weighting functions. This approach has the advantages that one can specify arbitrary levels of continuity in the spline fit (via the weighting functions) and that it can be readily extended to arbitrary dimensions.

A simple example of this 1D weighting function approach is presented in Figure 4, where the original data used to generate the spline fit are simply 17 points sampled at equal intervals from the function $\sin(2\pi x)$. The goal of this example is to create a spline fit made up of four spline regions which exhibits C^2 continuity at the spline zone interfaces. The first step is to generate five overlapping local fits Z_1 through Z_5, with each of the interior fits spanning two spline regions (see top of Figure 4). A least squares method is used to find a best fit quadratic function in each of the five regions:

$$Z_n(\bar{x}) = a_n + b_n\bar{x} + c_n\bar{x}^2$$

Since each spline zone now has two different local fits, one from the left and the other from the right, these two local fits are combined together with the left and right weighting functions shown in Figure 4 (middle). The form of the 1D weighting function used here for C^2 continuity is

$$W_{RIGHT}(\bar{x}) = \bar{x}^3\left(10 - 15\bar{x} + 6\bar{x}^2\right)$$

and from symmetry we have $W_{LEFT}(\bar{x}) = W_{RIGHT}(1-\bar{x})$. Thus the final fit in each region can be written as

$$F(x,y) = W_{LEFT}Z_{LEFT} + W_{RIGHT}Z_{RIGHT}$$

For example, for region 2, one would have $Z_{LEFT} = Z_2$ and $Z_{RIGHT} = Z_3$. Note that in addition to providing the desired level of continuity at spline boundaries, the weighting functions are also useful in reducing the dependence on the extreme ends of the local fits where they often exhibit the poorest agreement with the original data. When these final fits are plotted (bottom of Figure 4), we see that they are indeed C^2 continuous, maintaining continuity of the function value, slope, and curvature at all three interior spline boundaries. Outer domain boundary conditions (Dirichlet and Neumann) can be satisfied exactly by using constrained local fits in the boundary regions.

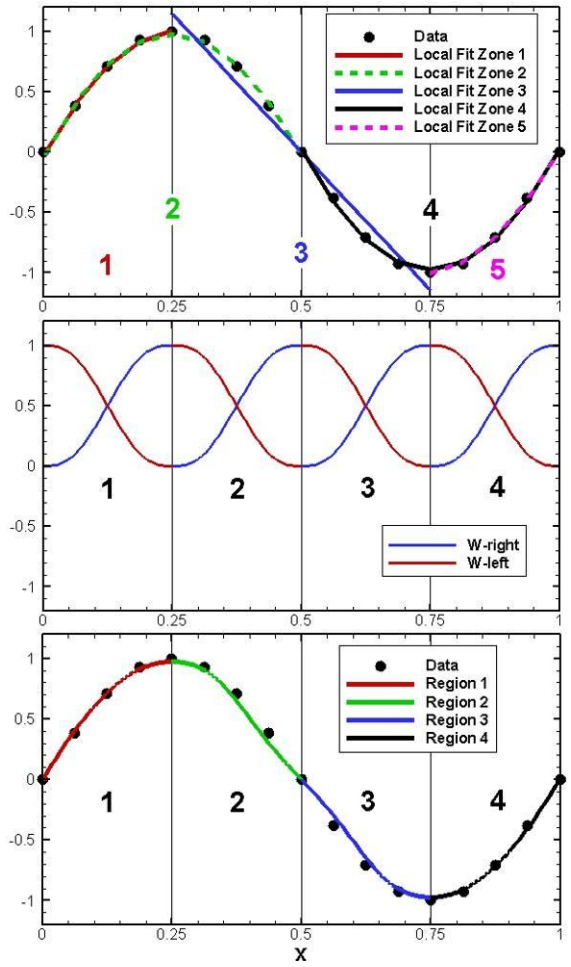


Figure 4. Simple one-dimensional example of the weighting function approach for combining local quadratic least squares fits to generate a C^2 continuous spline fit: local fits (top), weighting functions (middle), and resulting C^2 continuous spline fit (bottom).

The right 1D weight functions for varying levels of continuity (C^0 through C^6) are given in Table 1, and these weight functions are shown graphically in Figure 5. Note that the left weighting function can be found by simply inserting $x_{LEFT} = 1 - x_{RIGHT}$.

Table 1. One-dimensional weight functions

Continuity Order	Right Weight Function
0	x
1	$x^2(-2x+3)$

2	$x^3\left(6x^2-15x+10\right)$
3	$x^4\left(-20x^3+70x^2-84x+35\right)$
4	$x^5\left(-70x^4+315x^3-540x^2+420x-126\right)$
5	$x^6\left(-252x^5+1386x^4-3080x^3+3465x^2-1980x+462\right)$
6	$x^7\left(924x^6-6006x^5+16380x^4-24024x^3+20020x^2-9009x+1716\right)$

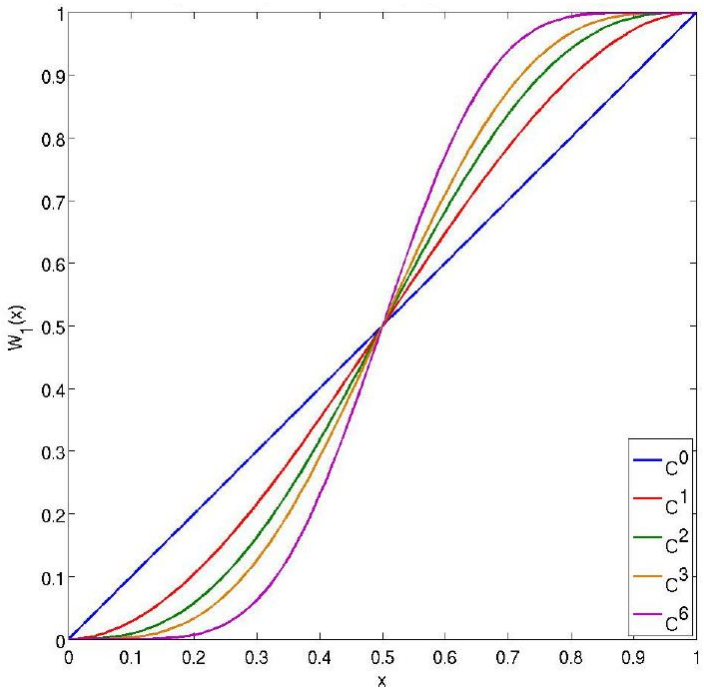


Figure 5. Right 1D weight functions: varying orders of continuity.

2.2. Higher-Dimensional Fits

The one-dimensional weighting function-based spline fitting procedure can be extended to higher dimensions using the approach of Junkins et al.[13] For example, in 2D a series of overlapping 2D local curve fits $Z_n(x,y)$ are joined together with higher-order weighting functions $W_n(x,y)$. In this case, the fitting function for a local region can be written as:

$$Z(x,y) = \sum_{n=1}^{4} Z_n(x,y) \cdot W_n(\bar{x},\bar{y})$$

where the *n* index represents different fitting regions for the local curve fits and the overbars indicate that the independent variables in the weighting functions are normalized to go from zero to one on each local region, i.e.,

$$0 \le \bar{x} \le 1, \quad \text{and} \quad 0 \le \bar{y} \le 1$$

The basis functions for these local curve fits Z_n can be chosen as needed, and thus can be modified to handle regions containing strong gradients or singularities. The weighting functions W_n are chosen such that an arbitrary level of continuity can be enforced between the fitting regions (function value, first derivative, second derivatives, etc.). In addition, these weighting functions ensure that the contribution from the local curve fits goes to zero at the boundaries of their local regions of applicability. For two-dimensional problems, these weighting functions are chosen such that the functions W_2, W_3, and W_4 are simple coordinate transformations of W_1:

$$W_2(\bar{x},\bar{y}) = W_1(1-\bar{x},\bar{y})$$
$$W_3(\bar{x},\bar{y}) = W_1(1-\bar{x},1-\bar{y})$$
$$W_4(\bar{x},\bar{y}) = W_1(\bar{x},1-\bar{y})$$

Additional constraints used to determine the form of the weighting functions in 2D are that the weighting function W_1 and its *k*-derivatives must equal zero along the two lines $\bar{x} = 0$ and $\bar{y} = 0$, it must equal unity at ($\bar{x} = 1$, $\bar{y} = 1$), and the sum of the weighting functions must be equal to unity

$$\sum_{n=1}^{4} W_n(\bar{x},\bar{y}) = 1$$

for all $0 \le \bar{x} \le 1$ and $0 \le \bar{y} \le 1$. For example, for C^1 continuity (i.e., continuity of the function value and its first derivative) at local fitting boundaries, the weighting function W_1 is

$$W_1(\bar{x},\bar{y}) = \bar{x}^2\bar{y}^2\left(9 - 6\bar{x} - 6\bar{y} + 4\bar{x}\bar{y}\right)$$

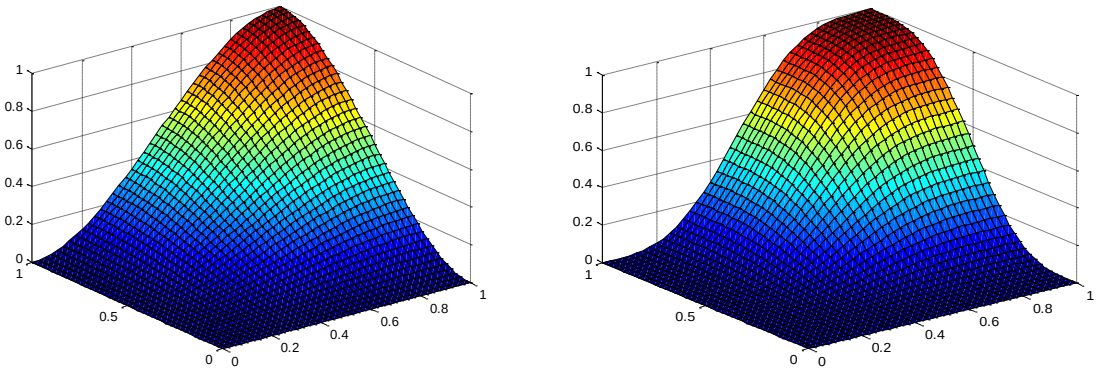
and the form for C^3 continuity (used for most of the current results) is

$$W_1(\bar{x},\bar{y}) = \bar{x}^4\bar{y}^4(1225 - 2940\bar{y} + 2450\bar{y}^2 - 700\bar{y}^3 - 2940\bar{x} + 7056\bar{x}\bar{y} - 5880\bar{x}\bar{y}^2 + 1680\bar{x}\bar{y}^3$$
$$+ 2450\bar{x}^2 - 5880\bar{x}^2\bar{y} + 4900\bar{x}^2\bar{y}^2 - 1400\bar{x}^2\bar{y}^3 - 700\bar{x}^3 + 1680\bar{x}^3\bar{y} - 1400\bar{x}^3\bar{y}^2 + 400\bar{x}^3\bar{y}^3)$$

where again $\bar{x}$ and $\bar{y}$ are linearly scaled to vary between zero and one in each region. These two weighting functions are shown graphically in Figure 6 and are essentially one quadrant of a three-dimensional bell shape with a square base. Although possibly not unique, the general form of the 2D weighting function for C^k continuity is:

$$W_1^k(\bar{x},\bar{y}) = \bar{x}^{k+1}\bar{y}^{k+1}\sum_{i=0}^{k}\sum_{j=0}^{k} a_{i,j}\bar{x}^i\bar{y}^j$$

where the $a_{i,j}$ coefficients are found using the constraints given above. It should be noted that once the series of local fits has been generated, they can be joined together with weighting functions of varying degree of continuity in order to provide the best overall fit on the domain.



a) b)

Figure 6. Weighting functions $W_1(\bar{x},\bar{y})$ for a) C^1 continuity and b) C^3 continuity.

Higher-dimensional weight function can be constructed by multiplying 1D weight functions together (where the 1D functions have different independent variables), such as shown in Equation (3) for C^2 continuous weight functions in 2D (see Ref. 14).

$$\begin{aligned} W_1(x,y)\big|_{2D} &= W_1(x)\big|_{1D} * W_1(y)\big|_{1D} \\ &= \left\{x^3\left(6x^2-15x+10\right)\right\} * \left\{y^3\left(6y^2-15y+10\right)\right\} \\ &= x^3y^3\left(36x^2y^2-90x^2y+60x^2-90xy^2+225xy-150x+60y^2-150y+100\right) \end{aligned} \tag{3}$$

Three-dimensional weight functions follow the same general form as the one- and two-dimensional weight function. Figure 7 shows six slices of a three-dimensional weight function. The slices are presented due to the difficulty of representing data that is dependent on three independent variables. Note that these weight functions could be used for a steady-state problem with three spatial dimensions or for a time-varying problem with two spatial and one time dimension.

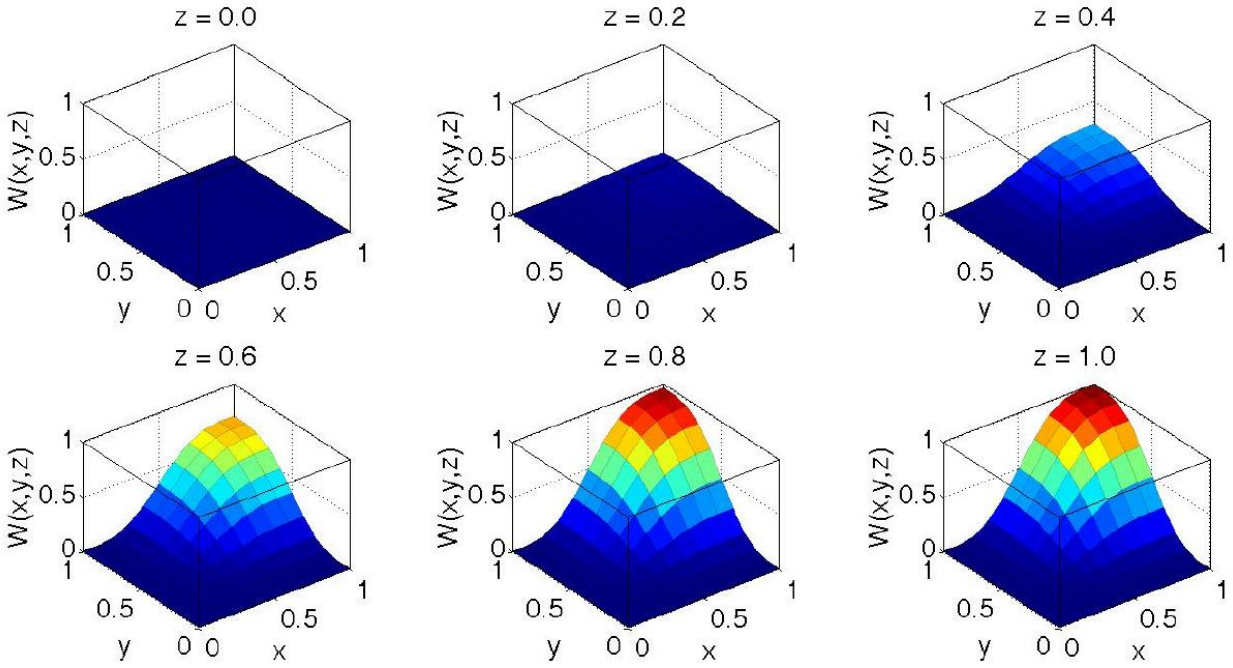


Figure 7. Six slices of a three-dimensional weight function.

2.3. Four-Dimensional Spline Fitting Tool

A MATLAB spline fitting package was written to generate up to four dimensional, arbitrary continuity global spline fits. The package is built around a MATLAB class designed to store and manipulate polynomials up to four dimensions. The polynomial class was designed so that common mathematical operators can be used to manipulate polynomials where the primary operations include polynomial multiplication and addition required for MNP procedures. Other important operations include coordinate transformation routines to transform a polynomial to different coordinate system. In addition to specifying arbitrary dimensions, data points per spline zone, continuity, and other MNP specific variables, the spline fitting package also allows for constrained fits to a constant value along a boundary. The constrained fits can be useful to ensure that the resulting spline fit is consistent with physical boundary conditions (e.g., zero normal velocity for a slip-wall boundary condition). Figure 8 shows a simplified flowchart of the spline fitting package, where the procedures outlined in red are part of the general package. There are two primary routines. Given the data to be fitted and a configuration variable, the first function performs the local least squares fits and returns a matrix storing the polynomials. This matrix is then assembled with the weight functions in the next routine returning a final matrix of continuous polynomials where each polynomials domain is [0,1] in all coordinate directions. All other procedures are part of a wrapper required to setup the particular problem of interest by specifying problem dimension, continuity order, number of points per spline zone, etc.

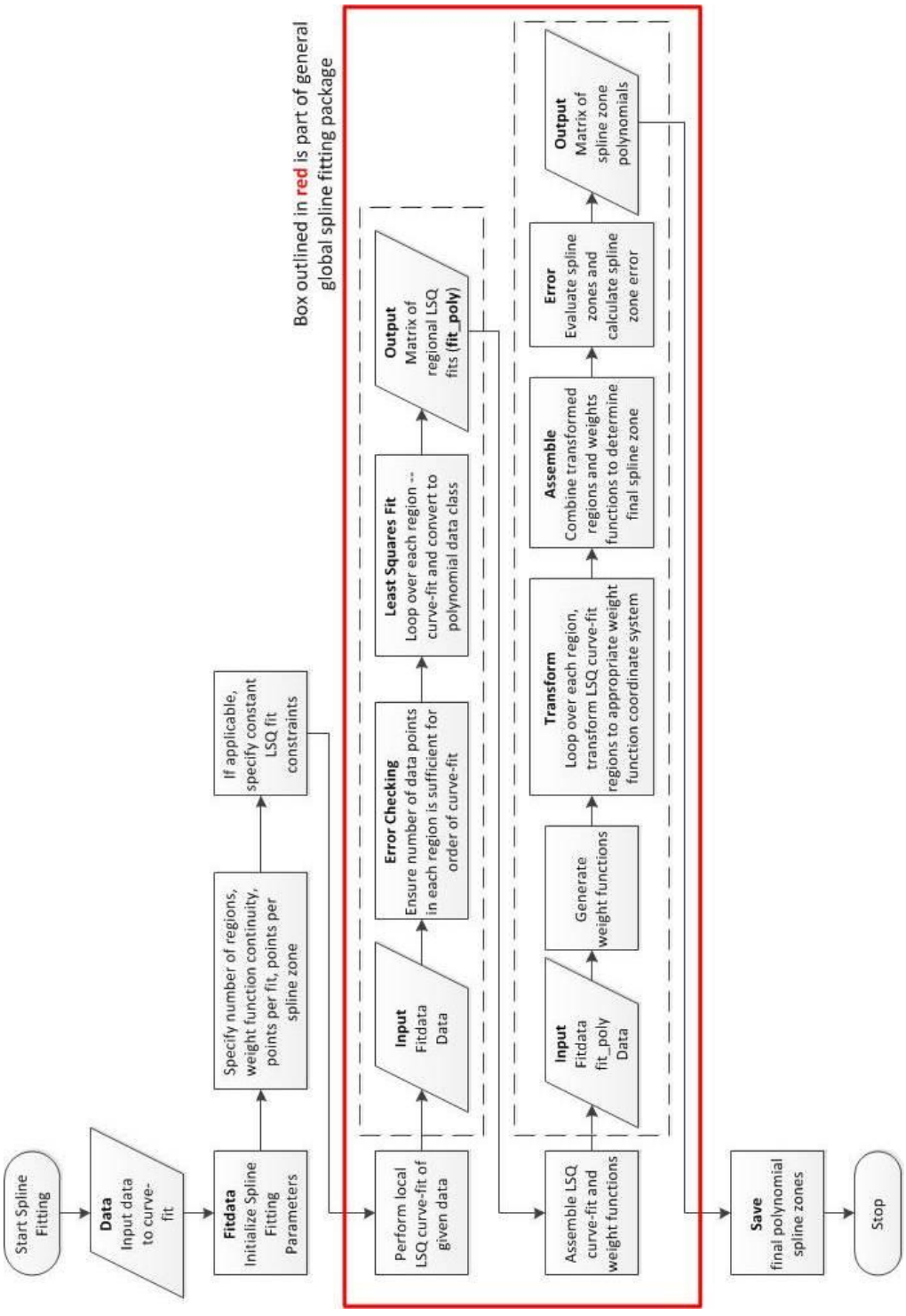


Figure 8. Flowchart of the Four-Dimensional Spline Fitting Tool.

3. DISCRETIZATION ERROR ESTIMATION THEORY

3.1. Residual-Based Error Estimation Framework

A residual-based error estimation and adaptivity framework was developed by Roy[16] where the Generalized Truncation Error Expression (GTEE) was formulated to relate the discrete equations (e.g., the finite volume method) to the partial differential or integral equations through the truncation error. However, the GTEE requires appropriate operators to prolong discrete quantities (e.g., finite volume solutions at cell centers) to continuous functions and to restrict continuous functions to cell center or nodal locations. We therefore introduce the interpolation function I which can perform both prolongation and restriction operations. This interpolation function is designed to be read from bottom to top. Consider the following examples where u_h is a discrete numerical solution on a fine mesh with spacing h, u_{2h} is a discrete numerical solution on a coarse mesh with spacing $2h$, and $\tilde{u}$ is the continuous exact solution to the governing partial differential or integral equations:

- prolongation of u_h to a continuous space: $I_h u_h$
- prolongation of u_{2h} to mesh h: $I_{2h}^h u_{2h}$
- restriction of u_h to mesh $2h$: $I_h^{2h} u_h$
- restriction of $\tilde{u}$ to mesh h: $I^h \tilde{u}$

When no subscript or superscript is present, a continuous function is implied. Using this interpolation operator, the GTEE can be rigorously recast for finite difference and finite volume schemes as

$$L_h(I^h \tilde{w}) = I^h L(\tilde{w}) + TE_h(\tilde{w}) \tag{4}$$

where $L_h(\cdot)$ represents the discrete equations on mesh h, $L(\cdot)$ the partial differential (or integral) equations, $TE_h(\cdot)$ is the truncation error on mesh h, and $\tilde{w}$ is any general smooth function. The three terms in Equation (4) refer to nodal values for finite difference methods and cell center values for finite volume methods. In addition, the discretization error can be defined either continuously or discretely as

$$\varepsilon = I_h u_h - \tilde{u} \qquad \text{or} \qquad \varepsilon_h = u_h - I^h \tilde{u}, \tag{5}$$

respectively.

3.2. Continuous Defect Correction

Defect correction methods were originally developed nearly 50 years ago in order to improve the accuracy of numerical solutions to ordinary differential equations.[17,18] More recently, defect correction has been used in CFD to increase the accuracy of first order spatial discretization schemes.[19,20] We are interested in using defect correction methods to estimate the discretization error in CFD solutions. There are two main types of defect correction[6]: differential correction

(referred to herein as continuous defect correction) and difference correction (referred to herein as discrete defect correction).

The original problem that we start with is the PDE ($L(u)=0$) which is exactly solved by $\tilde{u}$. In general, we do not know $\tilde{u}$, but would like to estimate it. We use a numerical method to come up with an approximate problem ($L_h(u_h)=0$) which is the numerical approximation to the PDE (i.e., the discretization scheme) that we can solve. We now formulate a differential equation which is "nearby" our original equation by prolonging the numerical solution to a continuous space, operating the PDE onto it, and then adding it as a source term to the right-hand side of the PDE to give

$$L(u) = L(I_h u_h) \tag{6}$$

with dependent variable u. We note that this differential equation will be exactly satisfied by the prolongation of u_h onto a continuous space, i.e., by the continuous function $I_h u_h$. We can also form an approximation of this "nearby" problem by discretizing the left-hand side of Equation (6) as

$$L_h(u_h') = I^h L(I_h u_h) \tag{7}$$

which can be solved numerically to give our approximate solution to the nearby problem u_h'. The key aspect of defect correction is that the error that arises by solving our approximation of the original problem should be very close to the error that arises by solving our approximation to the nearby problem since both simply involve the discretization of the PDE operator. Since the discretization error in the approximate solution to the nearby problem does not need to be estimated, but can be evaluated directly as

$$\varepsilon_h' = u_h' - u_h \tag{8}$$

(where we have again assumed that $I^h I_h u_h = u_h$), we can estimate the discretization error that occurs when approximately solving (i.e., discretizing) the original PDE as

$$\varepsilon_h \cong \varepsilon_h' = u_h' - u_h . \tag{9}$$

Thus the continuous defect correction method requires an additional numerical solution, on the same mesh, to the approximate nearby problem given by Equation (7). This approximate nearby problem has the continuous residual (i.e., the PDE operating on the numerical solution which is an approximation of the truncation error) as a source term.

A key issue in the continuous form of defect correction is related to the technique used to prolong the numerical solutions to the continuous space (i.e., the I_h operator). The weighting function-based MNP approach described earlier used sophisticated spline fits which allowed for arbitrary levels of continuity across spline zone boundaries and could be readily extended to multiple dimensions.[21] However, the generation of such global functions is expensive and can lead to oscillations in regions with insufficient resolution. When using MNP/defect correction for

discretization error estimation purposes, global fits are likely not necessary, so the use of local fits will also be explored.

3.3. Discrete Defect Correction

For the discrete form of defect correction, the PDE in the original problem from Section 3.2 is simply replaced by a discrete operator which has a higher order of accuracy than the discrete operator that we would like to solve. The nearby problem thus becomes

$$L_h(u_h') = \bar{L}_h(u_h) \tag{10}$$

where $\bar{L}_h(\cdot)$ is a higher-order discrete operator, u_h is the lower order numerical solution, and u_h' is the numerical solution to the nearby problem. For example, if the underlying numerical method is second-order accurate, then this higher-order accurate operator should typically be at least fourth-order accurate. The numerical stability of this higher-order discretization is unimportant since it is not used to obtain numerical solutions, but only to compute a residual by operating it onto the solution from the lower order discretization.

4. GOVERNING EQUATIONS AND EXACT SOLUTIONS

4.1. One-Dimensional Burgers' Equation

Burgers' equation is a quasi-linear, parabolic partial differential equation of the form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{11}$$

where *u(x,t)* is a scalar field. Here the position is given by *x*, the time by *t*, and *ν* is the viscosity. We have selected Burgers equation because it is a scalar equation with a number of known exact solutions.[22] Of these, we have chosen the steady-state viscous shock wave solution for our initial testing. This solution is chosen because it is smooth, non-trivial, and is in the real plane. Dirichlet boundary conditions are $u \to 2$ as $x \to -\infty$ and $u \to -2$ as $x \to \infty$. This solution is given by

$$u'(x) = \frac{-2\sinh(x')}{\cosh(x')} \tag{12}$$

where the prime denotes a dimensionless variable. The Reynolds number for Burgers equation can be defined as

$$\mathrm{Re} = \frac{u_{ref} L_{ref}}{\nu} \tag{13}$$

where u_{ref} is taken as the maximum value for *u(x,t)* in the domain (here u_{ref} = 2 m/s), L_{ref} is the domain width (generally L_{ref} = 8 m), and the choice for ν specifies the Reynolds number.

This solution is related to dimensional quantities by the following transformations:

$$x' = x / L_{ref} \quad \text{and} \quad u' = uL_{ref} / \nu \tag{14}$$

Furthermore, the solutions are invariant to scaling by a constant α:

$$\bar{x} = x / \alpha \quad \text{and} \quad \bar{u} = \alpha u$$

Using the above transformations, the exact solution to 1D steady Burgers' equation for a viscous shock can be rewritten in terms of the Reynolds number as

$$\tilde{u}(x) = -2\tanh\left(\frac{\mathrm{Re}}{2}\frac{x}{L}\right) \tag{15}$$

This exact solution is represented by the curves in Figure 9 where *x* is on the ordinate and *u* the abscissa. Two different values for ν are shown, corresponding to Reynolds numbers of 8 and 64.

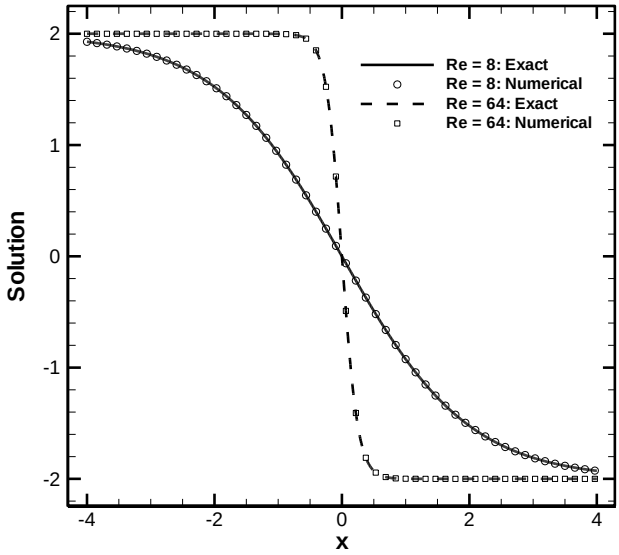


Figure 9. Exact and numerical solutions to Burgers equation for a steady, viscous shock wave at Reynolds numbers of 8 and 64.

4.2. Two-Dimensional Problems

4.2.1. Unsteady Burgers' Equation

The time-varying (unsteady) Burgers' equation is shown in Equation (16)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{16}$$

where t is time, x is spatial location, u is velocity, and ν is kinematic viscosity. One exact solution models viscous shock coalescence.[22] Two shocks travel from opposite ends of the x-axis and meet at the center. The two viscous shocks merge to form the steady-state shock as time increases. This solution is given in Equation (17)

$$u' = -\frac{2\sinh(x')}{\cosh(x') + e^{-t'}} \tag{17}$$

where the prime coordinates represent non-dimensionalized coordinates. Another exact solution is pulse decay[22] where two equal but opposite pulses are initiated at the origin and decay in time. This solution is given in Equation (18).

$$u' = -\frac{x'/t'}{1 + t'^{1/2} e^{x'^2/4t'}} \tag{18}$$

4.2.2. Compressible Euler Equations

The Euler equations describe inviscid, compressible fluid flow. The set of four equations governing two-dimensional flow is given in differential, conservation form in Equation (19)

$$\begin{aligned}
&\frac{\partial(\rho)}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0\\
&\frac{\partial(\rho u)}{\partial t}+\frac{\partial\left(\rho u^2+p\right)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}=0\\
&\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho vu)}{\partial x}+\frac{\partial\left(\rho v^2+p\right)}{\partial y}=0\\
&\frac{\partial(\rho e_t)}{\partial t}+\frac{\partial(\rho u e_t+pu)}{\partial x}+\frac{\partial(\rho v e_t+pv)}{\partial y}=0
\end{aligned} \tag{19}$$

where ρ is density, u is velocity in the x-direction, v is velocity in the y-direction, and p is pressure. These four variables are referred to as the primitive variables. The e_t term represents the total energy (internal plus kinetic) and is expressed in primitive variables as shown in Equation (20)

$$e_t=\frac{1}{\gamma-1}\frac{p}{\rho}+\frac{1}{2}\left(u^2+v^2\right) \tag{20}$$

where γ is the ratio of specific heats. The gas is assumed to be calorically perfect.

4.2.2.1. Supersonic Vortex

Supersonic vortex flow is an exact solution to the Euler equations and consists of flow around a 90 degrees annulus.

$$\begin{aligned}
&\rho(\mathrm{r})=\rho_\mathrm{i}\left(1+\frac{\gamma-1}{2}\mathrm{M_i^2}\left(1-\frac{\mathrm{R_i^2}}{\mathrm{r^2}}\right)\right)^{\frac{1}{\gamma-1}}\\
&u(y,r)=\frac{\mathrm{yU}}{\mathrm{r}},\qquad v(x,r)=-\frac{\mathrm{xU}}{\mathrm{r}},\qquad P=\frac{\rho^\gamma}{\gamma}\\
&U_i=M_i\rho_i^{\frac{\gamma-1}{2}},\qquad U=\frac{U_iR_i}{r}
\end{aligned} \tag{21}$$

The flow field is defined as a function of variables at the inner radius of the annulus denoted by the subscript i. For the problems examined herein, 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. See Ref. 23 for more information regarding supersonic vortex flow.

4.2.2.2. Ringleb's Flow

Ringleb's flow is an exact solution to the Euler equations for flow around a 180 degree turn.[24] The flow solution consists of both supersonic and subsonic regions; however, the domain was chosen so that only supersonic flow exists. The stream function for this flow is

$$\psi = \frac{1}{q}\sin(\theta) \tag{22}$$

where q is the normalized velocity of the flow and

$$k = \frac{1}{\psi} \tag{23}$$

Equations (22) and (23) are used to solve for the flow angle

$$\theta = \sin^{-1}\left(\frac{q}{k}\right) \tag{24}$$

The spatial locations are functions of q and k

$$x = \frac{1}{2h}\left(\frac{1}{q^2} - \frac{1}{k^2}\right) + \frac{J}{2} \tag{25}$$

$$y = \pm\frac{1}{qhk}\sqrt{1 - \left(\frac{q^2}{k^2}\right)} \tag{26}$$

where

$$h = c^{\frac{2}{\gamma-1}} \tag{27}$$

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\ln\left(\frac{1+c}{1-c}\right) \tag{28}$$

$$c = \sqrt{1 - \frac{\gamma-1}{2}q^2} \tag{29}$$

The Mach number is defined as

$$M = \sqrt{\frac{q^2}{1 - \frac{\gamma-1}{2}q^2}} \tag{30}$$

The maximum Mach number is a function of the ratio of specific heats γ. The flow field is defined in terms of k and q and the corresponding node locations are calculated. From the Mach

number and the flow angle, the primitive variables are calculated using isentropic relations for air at a pressure of p = 100,000 Pa and a temperature of T = 300 K. For the analysis presented herein, $1.1 < k < 1.6$ and $q_{min} = 1.05$, which gives a supersonic Mach number range of $1.19 < M < 2.29$.

4.2.3. Incompressible Navier-Stokes Equations

The viscous, incompressible flow in a lid-driven cavity is also examined at a Reynolds number of 100. This flow can be described by a modified form of the incompressible Navier-Stokes equations, which for constant viscosity is given by

$$\frac{1}{\rho\beta^2}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = S_2 + S_4$$

$$\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} - \frac{\partial p}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2} + \mu\frac{\partial^2 u}{\partial y^2}$$

$$\rho\frac{\partial v}{\partial t} + \rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} - \frac{\partial p}{\partial y} = \mu\frac{\partial^2 v}{\partial x^2} + \mu\frac{\partial^2 v}{\partial y^2}$$

These equations are solved in finite-difference form on a co-located (i.e., non-staggered) Cartesian mesh by integrating in pseudo-time using Chorin's artificial compressibility method.[25] In order to suppress odd-even decoupling (a common problem when solving the incompressible Navier-Stokes equations on non-staggered grids[26]), second- and fourth-derivative pressure damping (S_2 and S_4, respectively) are added to the mass conservation equation

$$S_2 = C_2\Delta x\frac{\left|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}\right|}{\left|p_{i+1,j} + 2p_{i,j} + p_{i-1,j}\right| + \varepsilon}\frac{\partial^2 p}{\partial x^2} + C_2\Delta y\frac{\left|p_{i,j+1} - 2p_{i,j} + p_{i,j-1}\right|}{\left|p_{i,j+1} + 2p_{i,j} + p_{i,j-1}\right| + \varepsilon}\frac{\partial^2 p}{\partial y^2}$$

$$S_4 = \frac{1}{4}\rho C_4\Delta x^2\frac{\partial^4 p}{\partial x^4} + \frac{1}{4}\rho C_4\Delta y^2\frac{\partial^4 p}{\partial y^4}$$

where C_2 = 1.0, $C_4 = -1\times10^{-4}$, and $\varepsilon = 1\times10^{-10}$ unless otherwise noted. This fourth derivative pressure damping is similar to that employed by Sotiropoulos and Abdallah.[27] This modified form of the equations reduces to the steady-state incompressible Navier-Stokes equations in the limit as the mesh is refined and as a steady-state solution is obtained. Dirichlet boundary conditions are used for velocity (all velocities are zero except for the u-velocity is set to unity at the top wall) and the boundary pressure is found from the interior solution.

5. RESULTS

5.1. One-Dimensional Burgers' Equation

5.1.1. Hermite Spline Fit

In our previous work[7] we used global Legendre polynomial fits to the underlying numerical solution. An example for a Reynolds number of 16 was shown in Figure 1a. The solutions using global polynomials exhibit large oscillations, especially at the boundaries. The MNP source term can then be found by inserting the curve fit of the numerical solution into the PDE. The poor representation of the underlying numerical solution leads to large source terms near the boundaries with magnitudes on the order of 10, as shown in Figure 10. Furthermore, the source terms become larger at the boundaries as the polynomial order is increased.

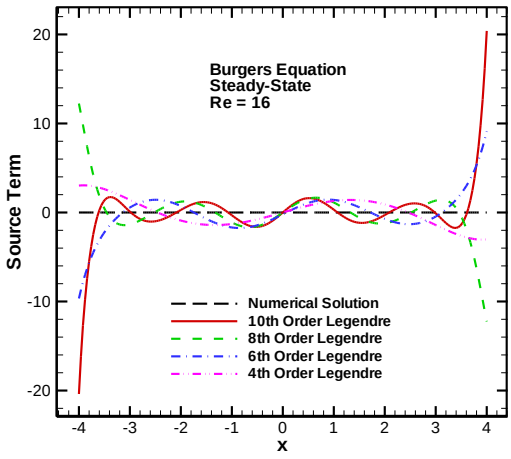


Figure 10. Source term distribution using different order Legendre polynomial fits for Burgers' equation at a Reynolds number of 16.

When the fifth-order Hermite splines are implemented,[11] the magnitude of the source term is significantly smaller. The source term for the Reynolds number 8 case using 17 spline points is shown in Figure 11 along the entire domain. The source term is much smaller than that seen in the global polynomial fits (Figure 10), with maximum magnitudes over the entire domain approximately 0.0015. Similarly, source terms were also calculated for the Reynolds number 64 case using Hermite spline fits. The distribution of the source term using 65 spline points is given in Figure 12. The maximum magnitude of the source term is on the order of 0.07 and occurs near the viscous shock wave (x = 0). The higher frequency seen in the Reynolds number 64 source term arises from the stronger gradients and increased number of spline points used as compared with the Reynolds number 8 case.

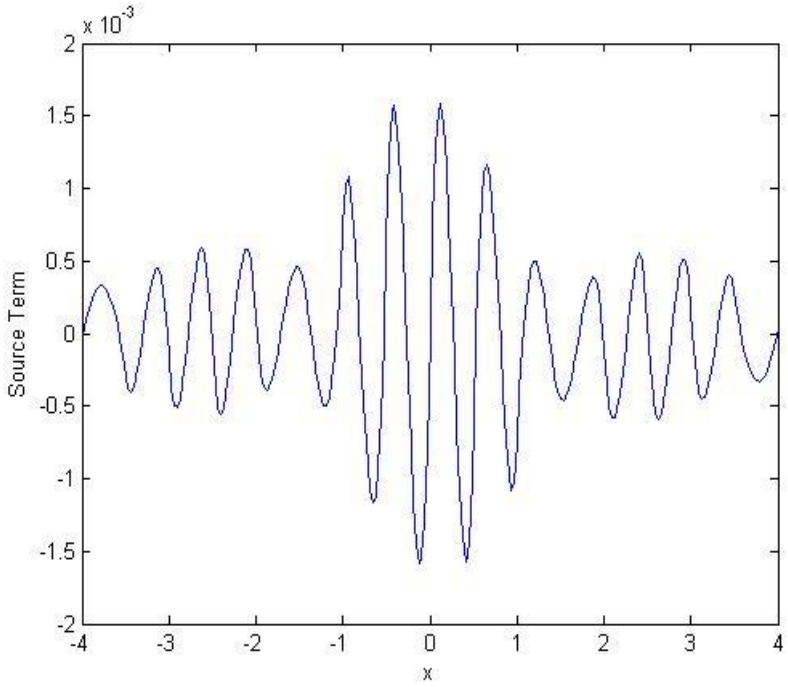


Figure 11. Source term for the nearby problem with 17 Hermite spline points for a Reynolds number of 8.

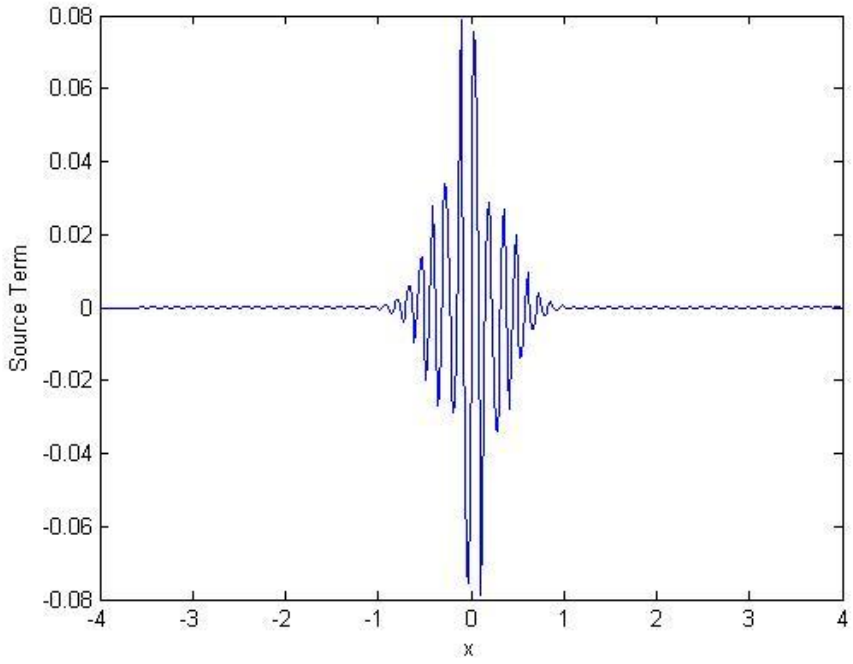


Figure 12. Source term for the nearby problem with 65 Hermite spline points for a Reynolds number of 64.

The Reynolds number 8 case was also approximated using standard cubic splines. Figure 13 shows the distribution of the source term along the domain using 17 spline points. While the magnitude of the source term is relatively small, the source term is no longer smooth. There are now slope discontinuities at each of the 15 boundaries between the spline zones, with the exception of the node at $x = 0$ due to the symmetry of the solution about this point.

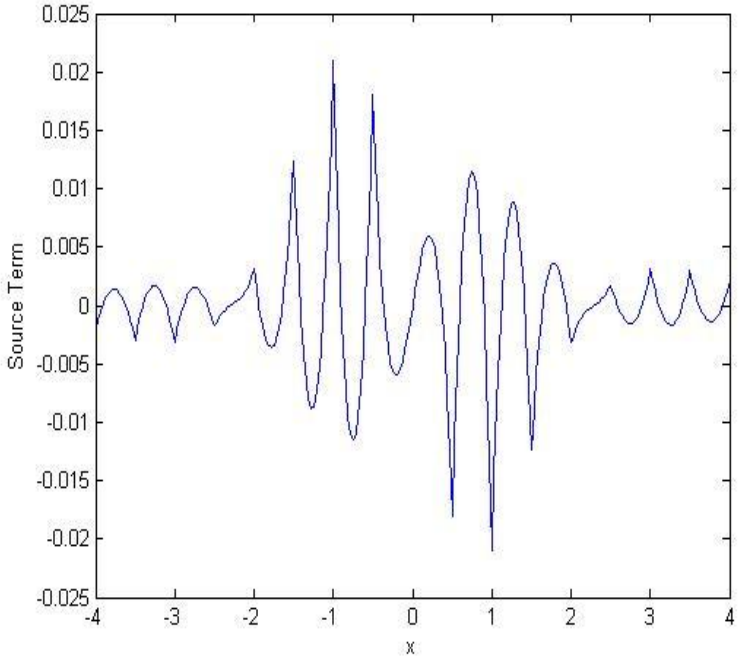
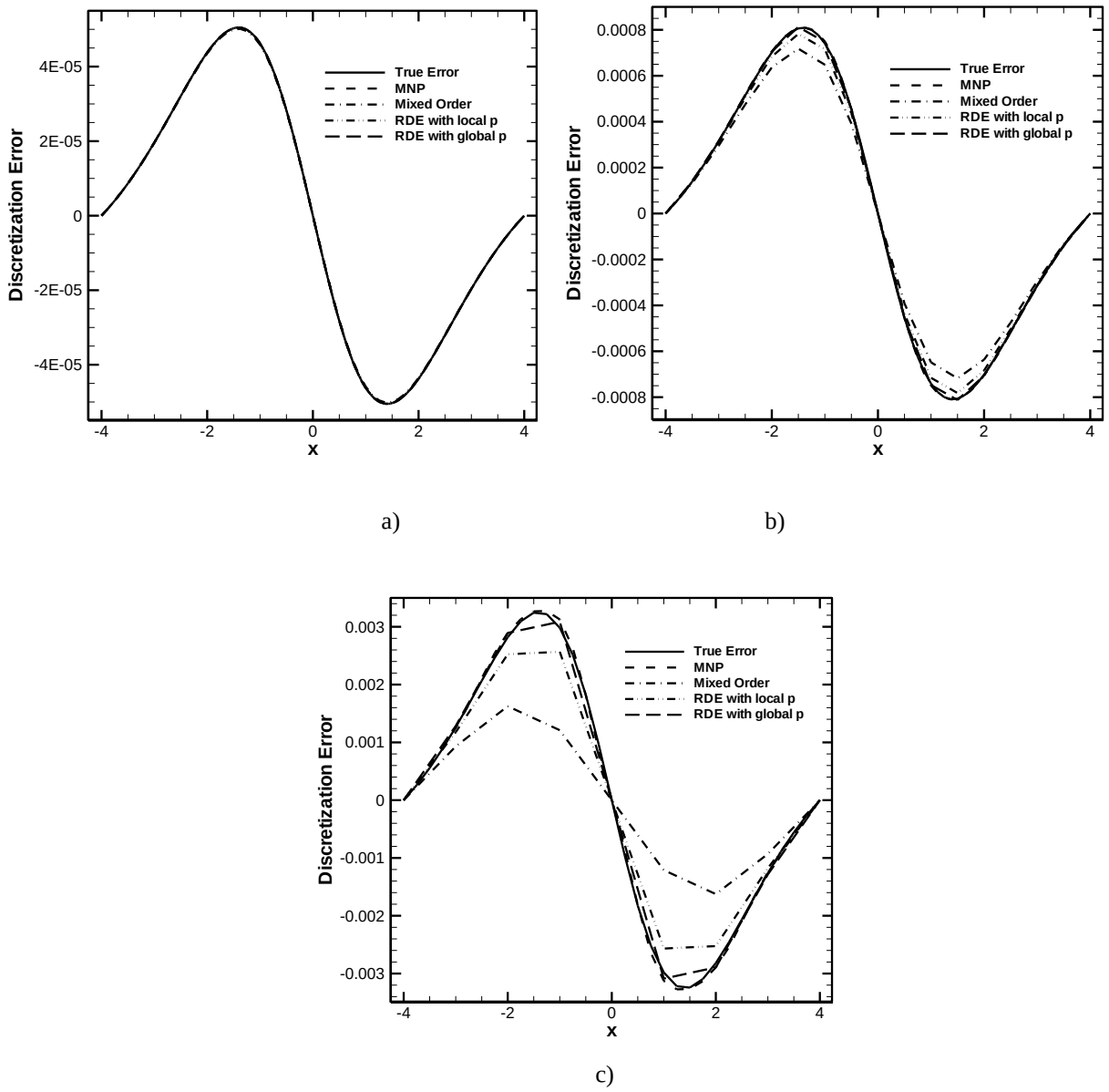


Figure 13. Source term for the nearby problem with 17 cubic spline points for Burgers' equation at a Reynolds number of 8.

The discretization error for Burgers' equation is estimated for the Reynolds number of 8 case using MNP as well as three extrapolation-based discretization error estimators[2,3] and compared to the true error. The distribution of the various discretization error estimates are shown in Figure 14 for a) a very fine mesh, b) a medium mesh, and c) a coarse mesh. For the fine mesh, all the error estimators agree well with the true error. As the mesh is coarsened, the mixed-order error estimator performs poorly, eventually under predicting the true error by a factor of two or more on the coarsest mesh. The same is true, but to a lesser extent, for Richardson extrapolation with the local order of accuracy, with the coarsest mesh under predicting the error by 10-15%. Both MNP and Richardson extrapolation using the formal order of accuracy provide good error estimates for this case.



a) b)

c)

Figure 14. Discretization error estimates for Burgers' equation with Reynolds number 8 using a finest mesh of a) 257 nodes, b) 65 nodes, and c) 33 nodes.

For the Reynolds number 64 case, the discretization error estimates are shown in Figure 15 for a) a 1025 node mesh, b) a 257 node mesh and c) a 65 node mesh. All of the approaches provide good error estimates for the fine grid case. For the 257 node mesh, the mixed-order error estimator greatly under predicts the error, Richardson extrapolation with the local order of accuracy slightly under predicts the error, and MNP and Richardson extrapolation using the formal order of accuracy provide good error estimates. The coarsest mesh that could be run without going unstable was 33 nodes. For the 65 node mesh, since only one coarser mesh solution is available, error estimates can be computed only by MNP and Richardson extrapolation with formal order of accuracy. As shown in Figure 15c, Richardson extrapolation gives error estimates that are nearly twice as large as the true error, while MNP gives estimates

that are within 20% of the true error. These results suggest that MNP can provide reasonable error estimates, even when the underlying mesh is not sufficiently refined enough to produce well-behaved asymptotic convergence.

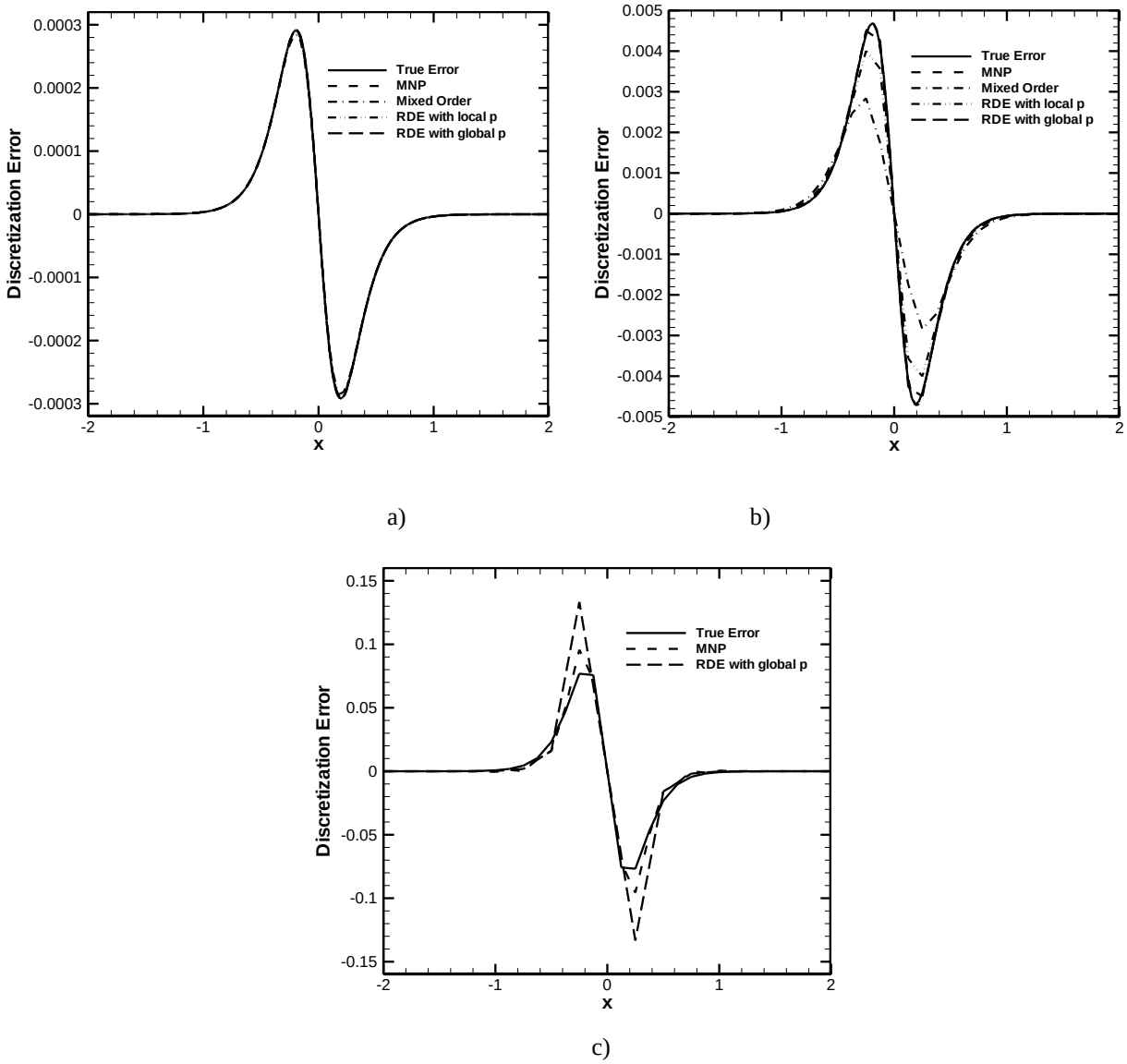
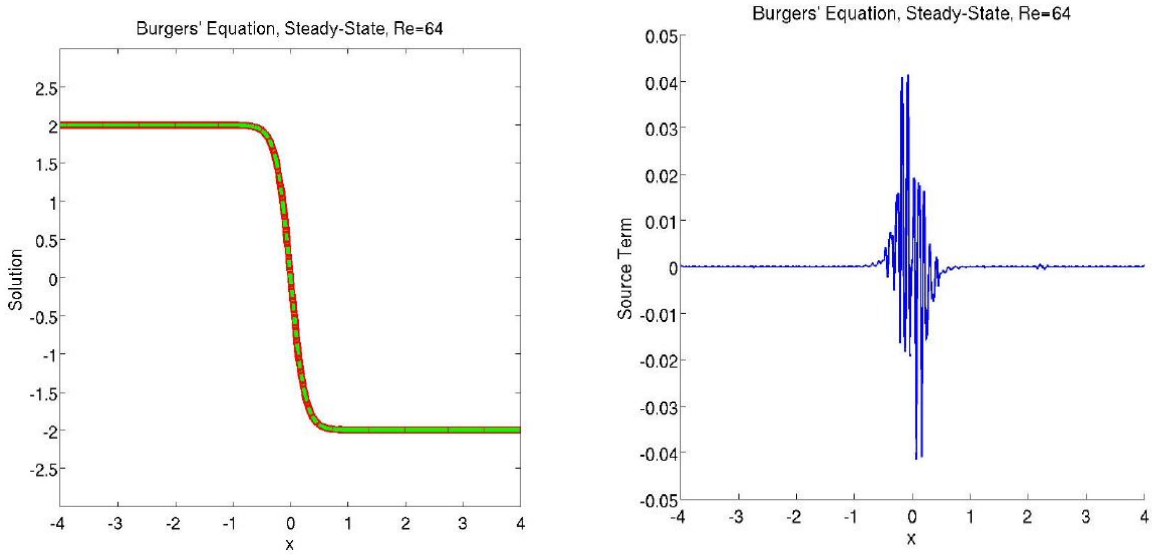


Figure 15. Discretization error estimates for Burgers' equation with Reynolds number 64 using a finest mesh of a) 1025, b) 257, and c) 65 nodes.

5.1.2. Weighting Function Approach

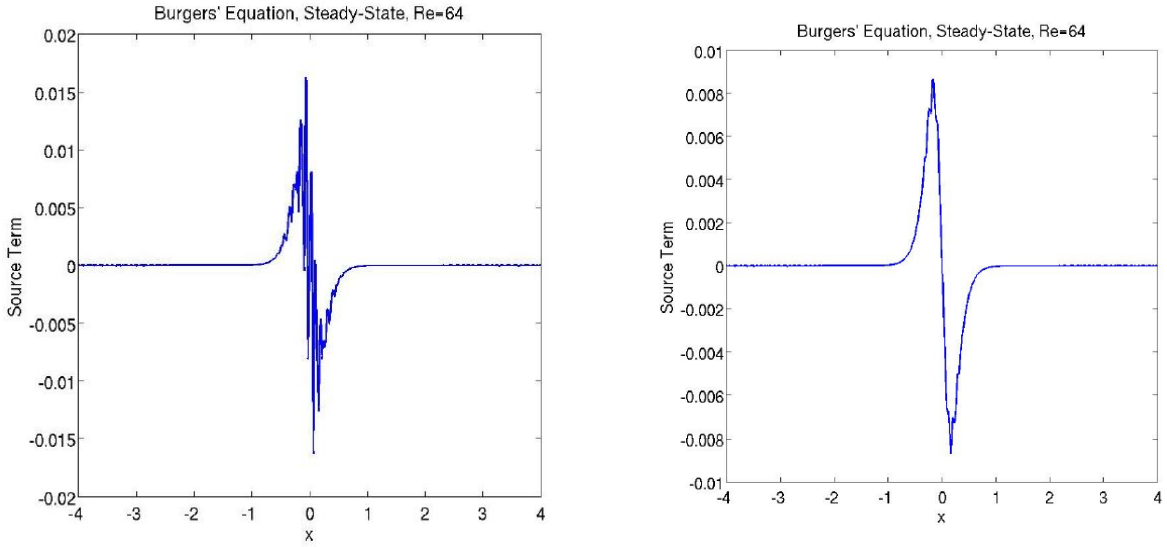
Weighting function-based fits are determined as described previously; the local least squares fits are merged with the weight functions to generate the piecewise final fit. This is done in MATLAB with symbolic expressions. This case is for a Reynolds number of 64, with 513 points in the numerical solution. 64 fit zones are used with fifth order local least squares fits and C^3 weight functions.[14] Figure 16 shows the numerical solution and curve fit, along with the source term.



a) b)

Figure 16. Burgers' equation: a) numerical solution and curve fit and b) source term.

The source terms often have a high frequency oscillation present. The magnitude of this oscillation seems to decrease when the order of the local fits is increased. As an example, Figure 17 shows the difference between the source terms resulting from sixth order local fits and seventh order local fits. The examples shown are for a Reynolds number of 64 and use 64 zones and C^3 weight functions.



a) b)

Figure 17. Effect of local fit order on source terms: a) 6th order and b) 7th order.

5.1.3. Local Fitting Approach

We have also examined a form of MNP/defect correction for the case when only local fits are used (i.e., without any weighing functions).[28] While this approach precludes the generation of a continuous exact solution, it can be used to provide discretization error estimates. For 1D steady Burgers' equation on a uniform mesh continuous defect correction is employed using fourth order accurate local polynomial fits of the numerical solution which is equivalent to discrete defect correction using fourth-order accurate finite differences. Richardson extrapolation and discretization error transport equations using two different linearization approaches are also applied (see Ref. 28 for more details). As shown in Figure 18, Richardson extrapolation overpredicts the error by almost a factor of two due to the coarse grid being outside the asymptotic range. The error transport equations with the simple linearization underpredict the error by ~30%. Defect correction and the advanced linearization for error transport equations provide accurate estimates of the true discretization error.

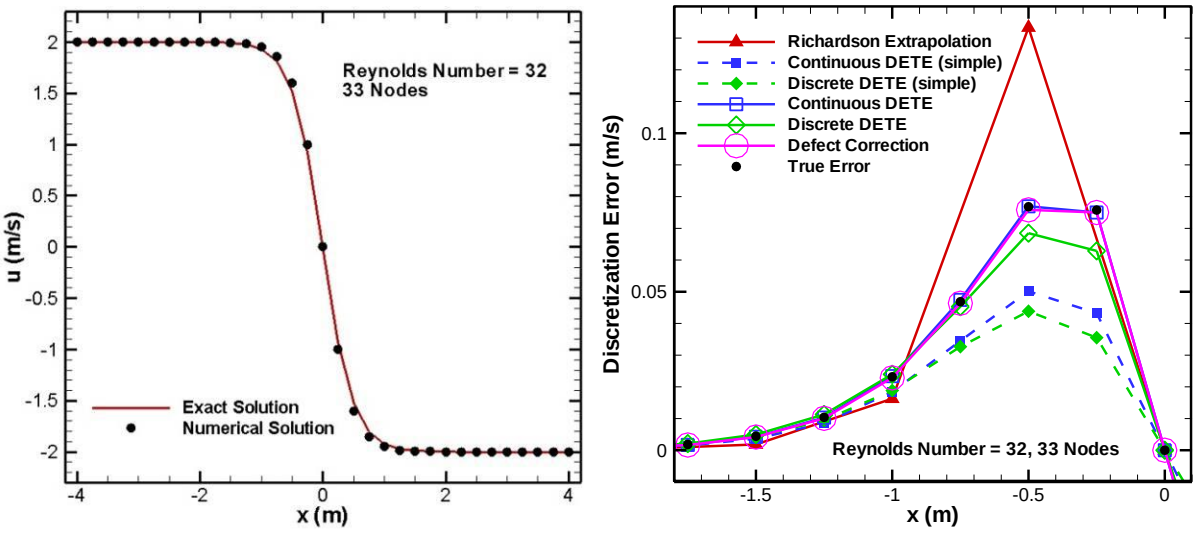


Figure 18. Burgers' equation for a viscous shock: numerical and exact solution (left) and residual-based discretization error estimates (right).

5.2. Two-Dimensional Cases

5.2.1. Unsteady Burgers' Equation

5.2.1.1. Shock Coalescence

Results for the shock coalescence case are shown in Figure 19 (see Refs. 14 and 15). The *x*-axis is the independent spatial axis varying from -4 m to 4 m and the *t*-axis represents time over a two second period. The *u*-axis measures the velocity, which varies from -2 m/s to 2 m/s. Note that two different grid levels are used in this section: a coarse grid with 65 points in each direction and a fine grid with 257 points in each direction.

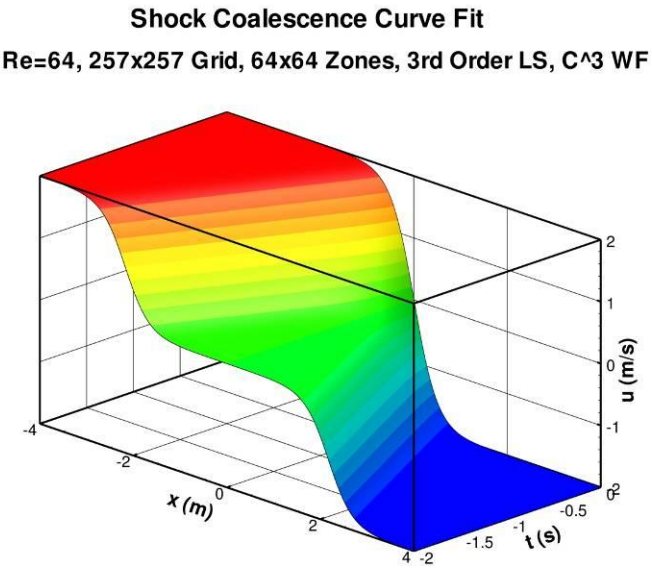


Figure 19. Shock coalescence curve fit surface plot.

Two-dimensional contour plots comparing the numerical solution and the curve fit on the fine grid are shown in Figure 20. The local fits for all of these two-dimensional results are done with a weighted least squares approach. The weights come from the same weighting functions that are used to patch the local fits together. Using the weighted least squares approach in this manner is intended to improve the local fits at the critical regions. A comparison of the curve fit error on the different grid levels is shown in Figure 21, while a comparison of the source term is shown in Figure 22. Both of these quantities have the largest values along the sharp gradients in the curve fit, where there are oscillations present. By refining the grid, these oscillations can be reduced in magnitude. Increasing the number of grid points that cover the sharp gradient helps to improve the resolution of the numerical solution and also provides for a better curve fit. Equation (31) is used to compute percent curve fit error, with the maximum shown for both grids in Table 2. The maximum percent curve fit error drops over two magnitudes when the grid is refined by a factor of four.

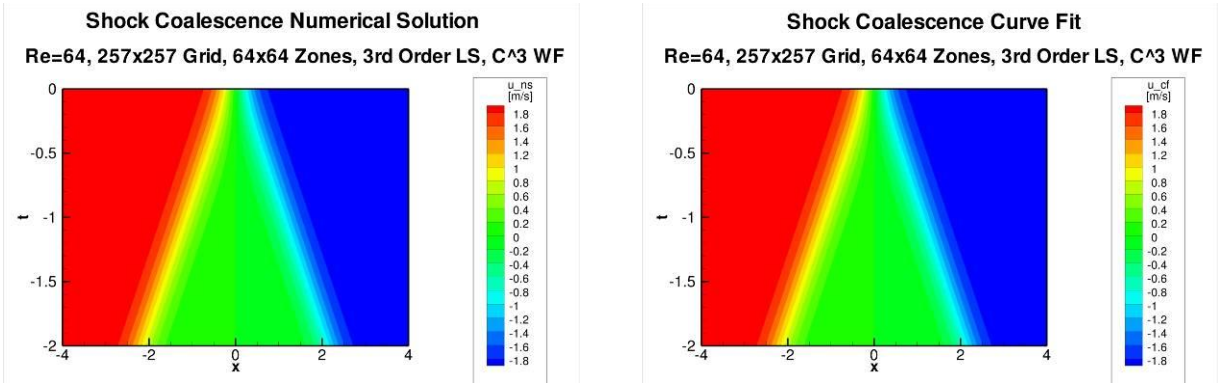


Figure 20. Shock coalescence numerical solution and curve fit.

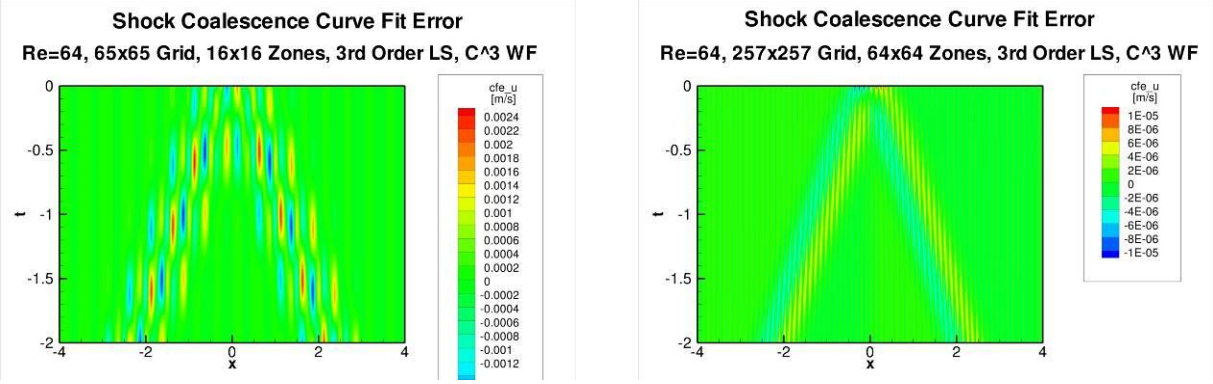


Figure 21. Shock coalescence curve fit error comparison.

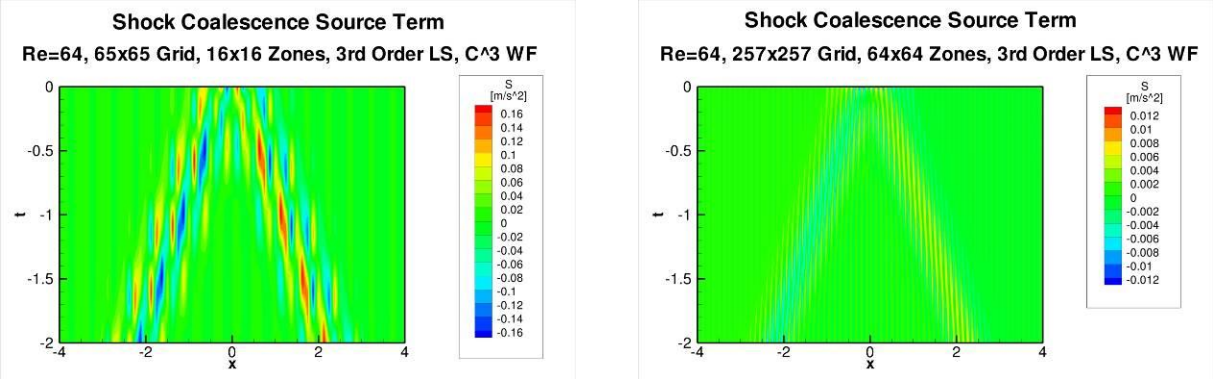


Figure 22. Shock coalescence source terms comparison.

$$\varepsilon_{CF,\%} = \left| \frac{u_{CF} - u_{NS}}{u_{ref}} \right| * 100 \tag{31}$$

In this equation $\varepsilon_{CF,\%}$ is percent curve fit error, u_{CF} is the curve fit, u_{NS} is the numerical solution, and u_{ref} is the reference velocity of 2 m/s. The percentage curve fit error is defined this way with a reference velocity in the denominator for both Burgers' equation cases because the velocities pass through zero.

Table 2. Shock coalescence maximum percentage curve fit errors

	% CF Error
Coarse Grid	0.13
Fine Grid	0.00058

A comparison of the original problem and nearby problem discretization errors is presented in Figure 23, for the coarse grid. The fine grid results are shown in Figure 24 and x-y plots showing a slice at $t = -1$ are given in Figure 25. The nearby problem discretization error matches the qualitative behavior of the original problem discretization error and the magnitudes are also close to one another. This holds true for both grid resolutions. However, the x-y plots show that the nearby problem discretization error exhibits noticeable oscillations near the shock region. These are minimized in the fine grid case.

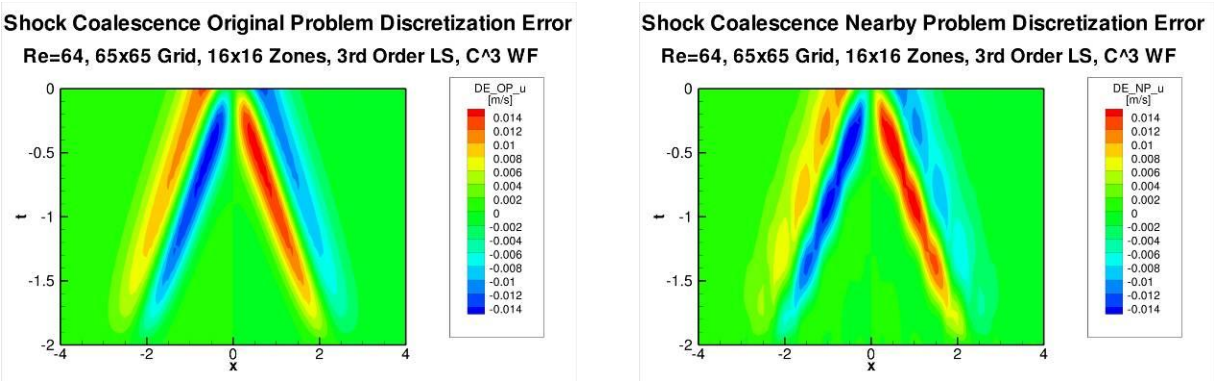


Figure 23. Shock coalescence discretization error comparison (coarse grid).

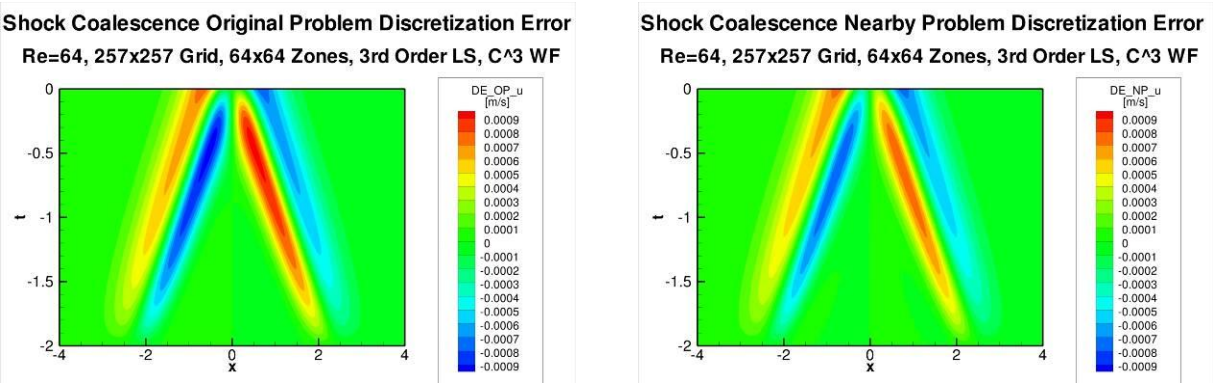


Figure 24. Shock coalescence discretization error comparison (fine grid).

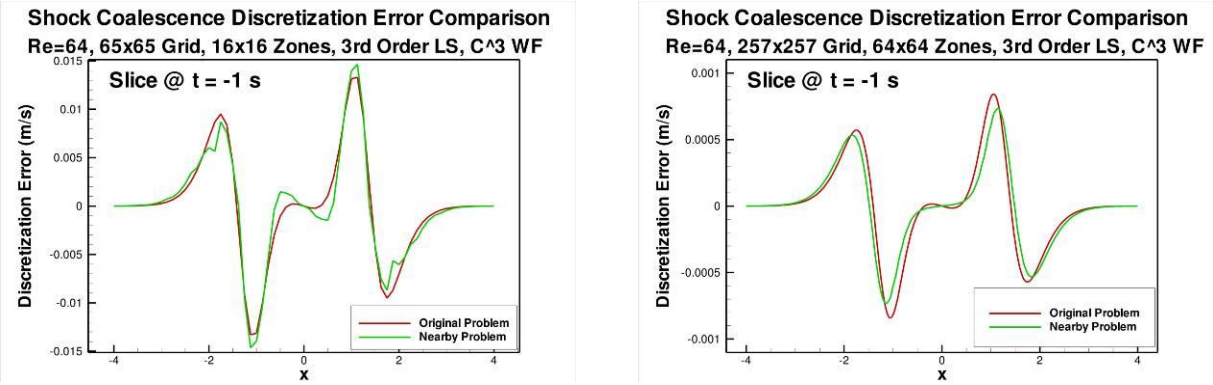


Figure 25. Shock coalescence discretization error comparisons (x-y plots).

5.2.1.2. Pulse Decay

The pulse decay case was also examined,[14] and a curve fit of this solution is shown in (x, t) space in Figure 26. The axes are the same as for the previous case. The pulse decay is shown from 0.2 seconds to 1 second. The numerical solution and curve fit are shown in Figure 27. Curve fit error on the coarse and fine grids are shown in Figure 28, while the source term is shown in Figure 29. The curve fit of the pulse decay is more difficult than that of the previous shock coalescence case. The pulse decay has sharp solution gradients where the initial pulses are formed. This represents a difficult region to fit accurately. In addition the pulses are strongest right at a boundary. The magnitude of the oscillations in the curve fit error and source term are decreased by refining the grid. The maximum percent curve fit errors are shown in Table 3, where grid refinement decreases the error by two orders of magnitude.

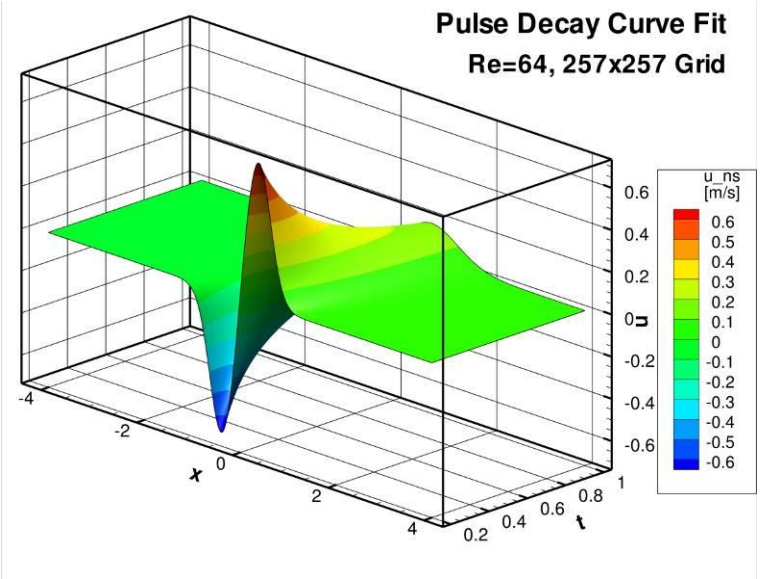


Figure 26. Pulse decay curve fit surface plot.

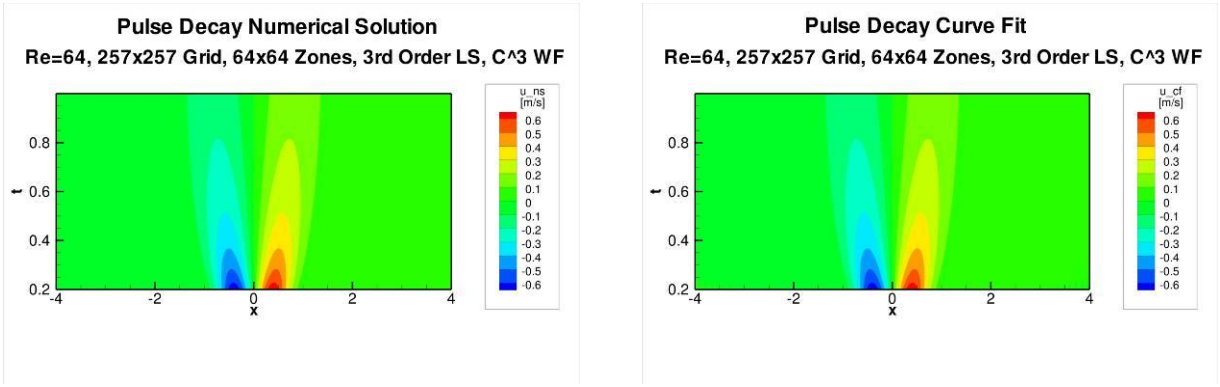


Figure 27. Pulse decay numerical solution and curve fit.

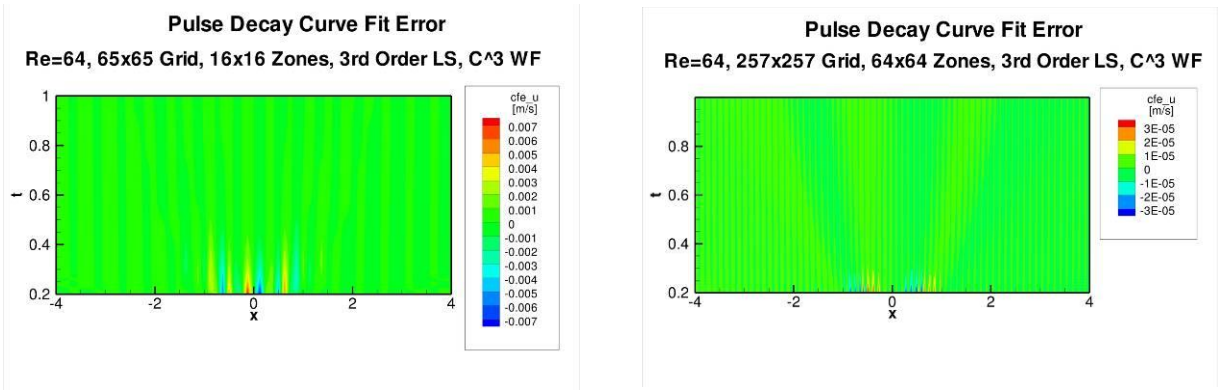


Figure 28. Pulse decay curve fit error comparison.

Table 3. Shock coalescence maximum percentage curve fit errors

	% CF Error
Coarse Grid	0.37
Fine Grid	0.0020

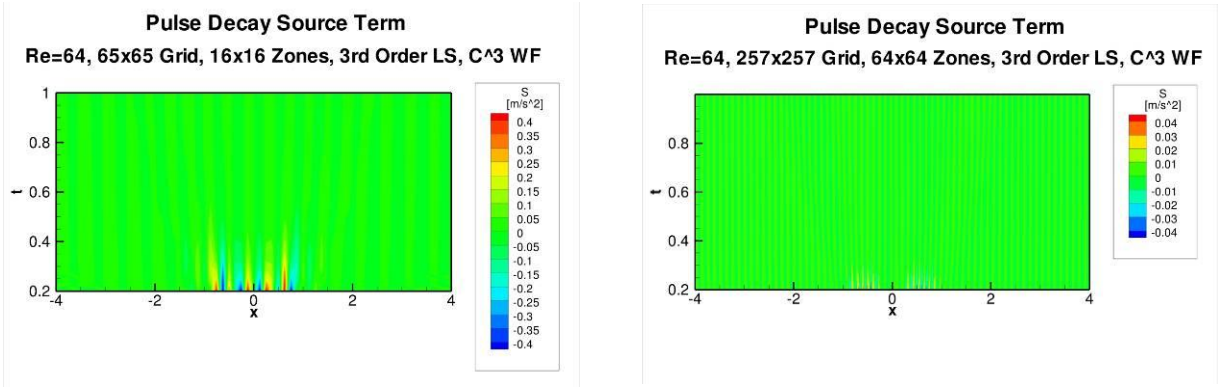


Figure 29. Pulse decay source term comparison.

The discretization errors in the original and nearby problems are presented in Figure 30 for the coarse grid. The nearby problem discretization error matches the qualitative behavior of the original problem discretization error. The discretization errors on the fine grid are shown in Figure 31. This refined grid actually does a worse job at estimating the error in the original problem. The two complete oscillations seen in the original problem discretization error are represented by only one oscillation in the nearby problem discretization error. This behavior can also be seen in Figure 32 with x-y plots comparing the two discretization errors for both coarse and fine grids. The reasons for such poor estimates of the discretization error are not known, but may be related to the presence of the large gradients at the $t = 0$ domain boundary.

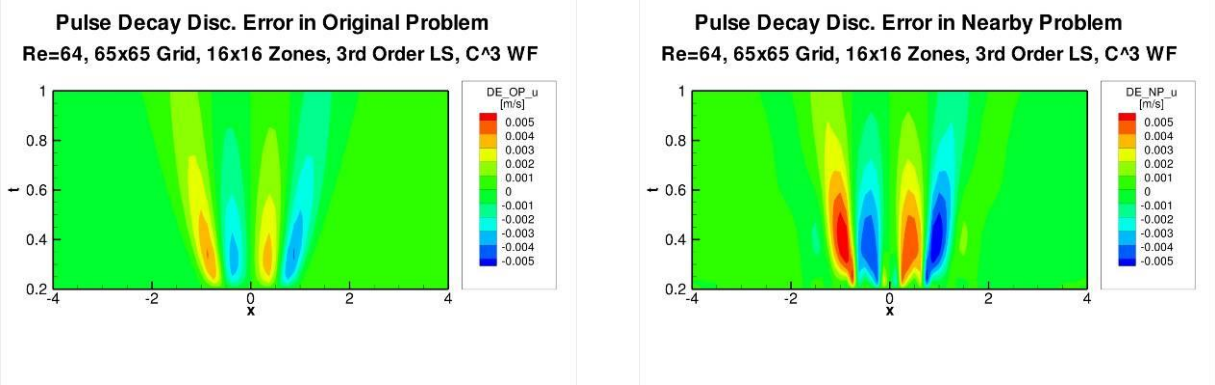


Figure 30. Pulse decay comparison of discretization error (coarse grid).

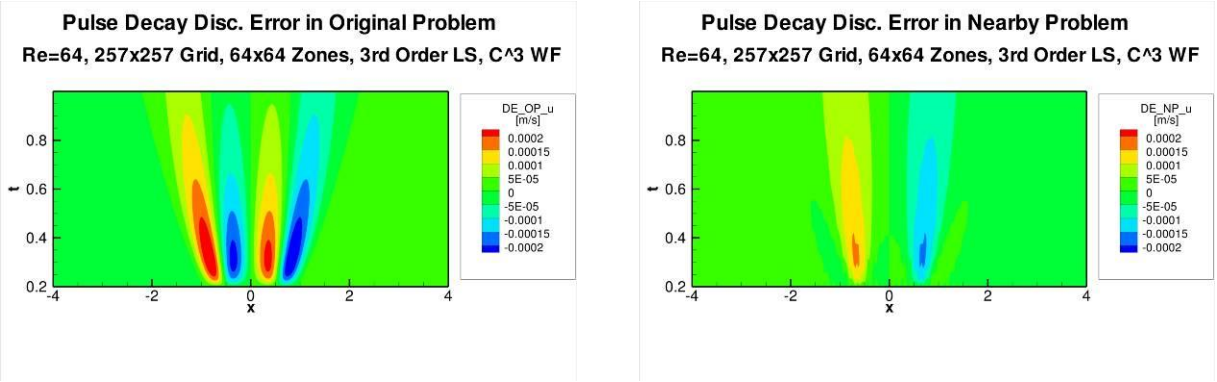


Figure 31. Pulse decay comparison of discretization error (fine grid).

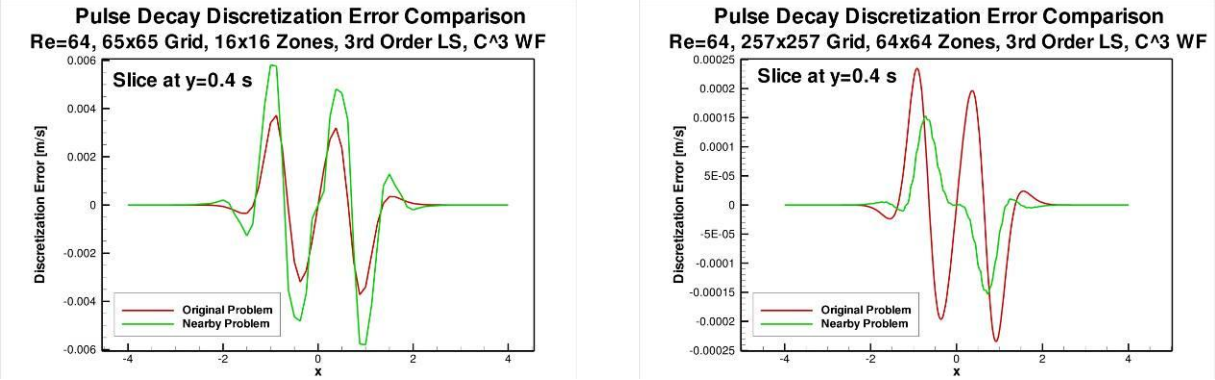
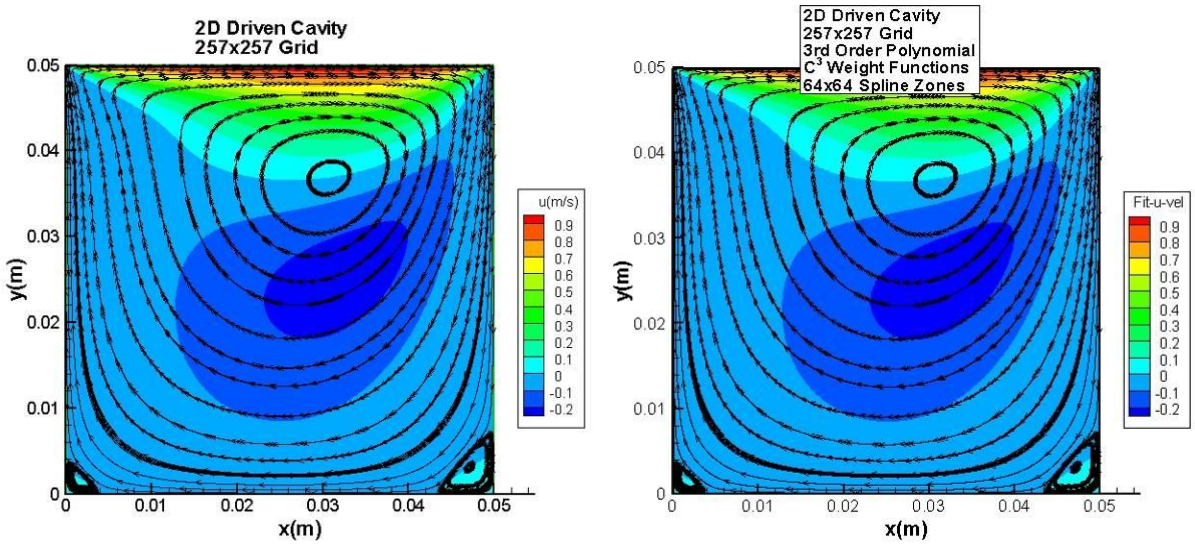


Figure 32. Pulse decay comparison of discretization error (x-y plots).

5.2.2. Incompressible Navier-Stokes

The incompressible Navier-Stokes equations are solved for the flow in a lid-driven cavity at a Reynolds number of 100 (see Ref. 21 for details). A contour plot of the *u*-velocity (i.e., the

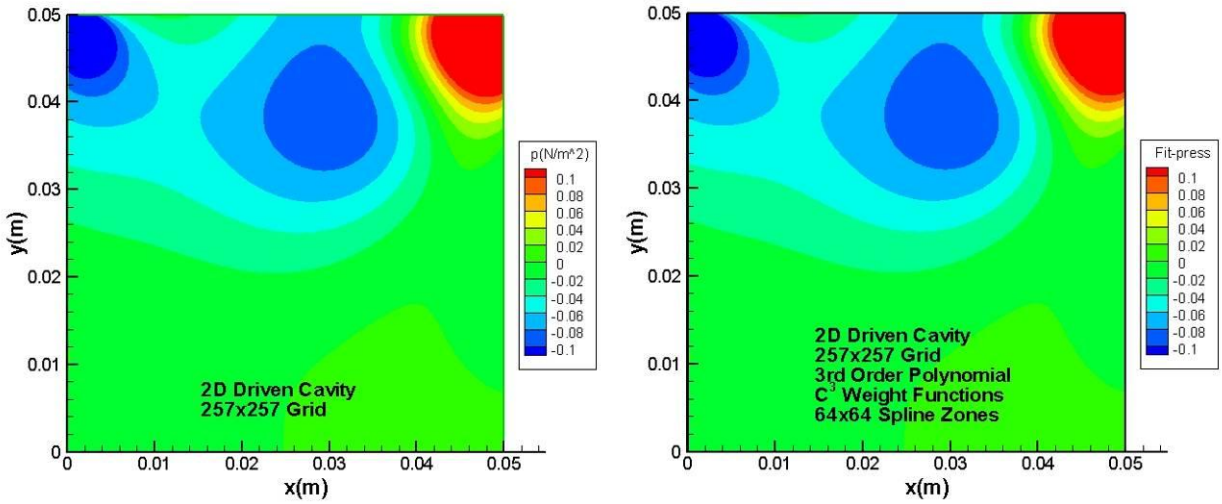
velocity in the *x*-direction) from a numerical solution on a 257×257 grid is given in Figure 33a. Also shown in the figure are streamlines which denote the overall clockwise circulation induced by the upper wall velocity (the upper wall moves from left to right), as well as the two counter-clockwise rotating vortices in the bottom corners. A spline fit was generated using 3rd order polynomials with C^3 continuous weighting functions and 64×64 spline zones. Note that while no additional boundary constraints are placed on the velocity components for the spline fit, the maximum deviations from the original boundary conditions are on the order of 1×10^{-7} m/s and are thus quite small. The *u*-velocity contours and streamlines for the spline fit are presented in Figure 33b. The fit solution is qualitatively the same as the underlying numerical solution. The streamlines were injected at exactly the same locations in both figures and are indistinguishable from each other. Furthermore, in both cases the streamlines near the center of the cavity follow the same path for multiple revolutions.



a) b)

Figure 33. Contours of *u*-velocity and streamlines for the driven cavity case at Reynolds number 100: a) 257x257 node numerical solution and b) C^3 continuous spline fit using 64x64 spline zones.

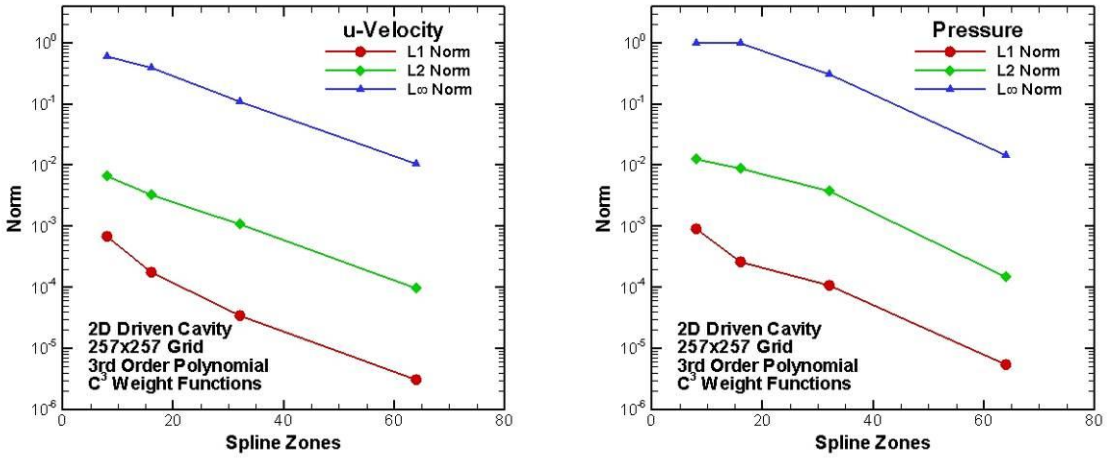
One of the difficulties that arises in both solving the driven cavity problem and in generating spline fits of the resulting solutions is the presence of strong singularities at the two top corners where the moving lid meets the stationary walls. The flow stagnation in the top right corner leads to a strong pressure rise, while the flow acceleration in the top left corner leads to a strong pressure drop. Contours of static (gauge) pressure for the 257×257 numerical solution are given in Figure 34a, while contours for the spline fit using 64×64 zones are shown in Figure 34b. Again, these contours are qualitatively identical.



a) b)

Figure 34. Contours of static gauge pressure for the driven cavity case at Reynolds number 100: a) 257x257 node numerical solution and b) C^3 continuous spline fit using 64x64 spline zones.

A more quantitative comparison between the underlying numerical solution and the spline fits is presented in Figure 35 which shows the spline fitting error relative to the numerical solution as a function of the number of spline zones in each direction. For the *u*-velocity (Figure 35a), the average error magnitude (L_1 norm) decreases from 1×10^{-3} m/s to 3×10^{-6} m/s with increasing number of spline zones from 8×8 to 64×64, while the maximum error (L_∞ norm) decreases from 0.7 m/s to 0.01 m/s. For the static pressure (Figure 35b), the average error magnitude decreases from 1×10^{-3} N/m^2 to 5×10^{-6} N/m^2 with increasing number of spline zones, while the maximum error decreases from 1 N/m^2 to nearly 0.01 N/m^2.



a) b)

Figure 35. Variation of the error between the spline fits and the underlying 257x257 numerical solution as a function of the number of spline zones in each direction for the driven cavity: a) *u*-velocity and b) pressure.

In addition to using the spline fitting approach to generate exact solutions, MNP has been employed to estimate the discretization error in the original driven cavity numerical solution. As discussed previously, if the nearby problem is "near enough" to the original problem of interest, then the discretization error in the nearby problem (which can be evaluated exactly) can serve as an estimate of the error in the original problem of interest. The estimated discretization error at the mid-height of the cavity using MNP for the 65×65 nodes solution is presented in Figure 36 along with error estimates from Richardson extrapolation (which also requires the solution on a coarser 33×33 node mesh). In order to judge which error estimation approach is more accurate, a numerical solution is also computed on a very fine 257×257 node mesh, with the resulting solution used to approximate the "true error" which is displayed as symbols in Figure 36. The discretization error estimates using MNP are generally not as good as those from Richardson extrapolation when the nominal value of the second-derivative damping coefficient (C_2 = 1.0) is used in the nearby problem; however, as this coefficient is reduced to 0.1 and then ultimately to zero (i.e., no second-derivative damping), the error estimates from MNP improve dramatically and are somewhat better than those found from Richardson extrapolation. In this case the nearby problem is likely smoother than the original problem, especially near the singular corners. Thus the nearby problem does not require the second derivative damping (whereas the original problem was unstable without some second derivative damping). In addition, these damping terms act as additional terms in the governing equations which are being neglected when computing the source term for the nearby problem. Table 4 gives the maximum difference between the different discretization error estimates and the true error as judged by the "truth" mesh, which can also be considered as the error in the error estimate. While Richardson extrapolation consistently out-performs the baseline MNP approach, as the second-derivative damping coefficient (C_2) is reduced, the error estimates from MNP are significantly improved relative to Richardson extrapolation.

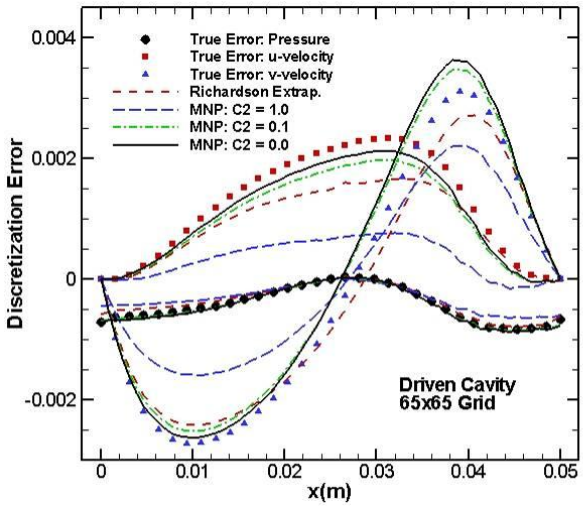
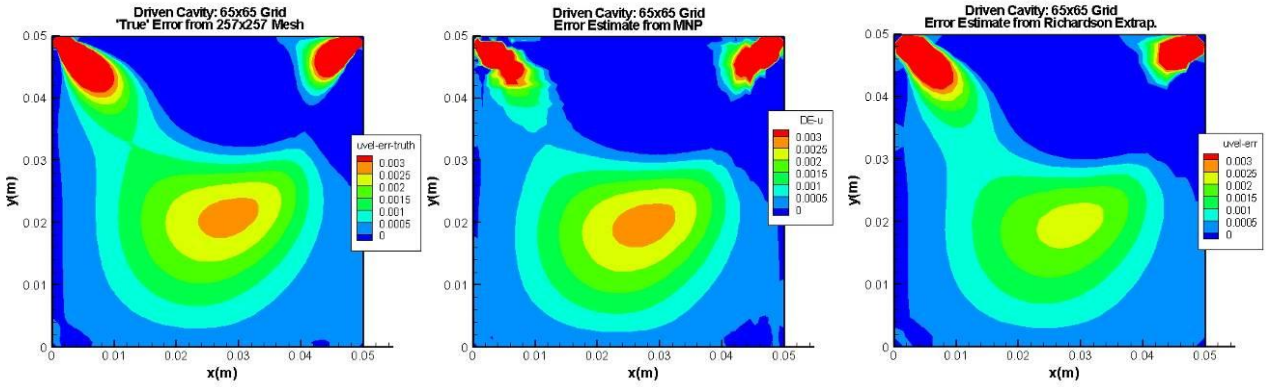


Figure 36. Discretization error for the standard driven cavity along the line *y* = 0.025 m (cavity centerline) showing the true error (estimated from a 257x257 grid), Richardson extrapolation (using grids of 65x65 and 33x33 nodes), and MNP using varying second-derivative damping constants.

Table 4. Maximum error magnitude in the discretization error estimate for the driven cavity case.

	RE	MNP	MNP (C2=0.1)	MNP (C2=0.0)
u-velocity	33%	66%	15%	8%
v-velocity	14%	32%	10%	16%
Pressure	20%	28%	6%	8%

Contour plots of the discretization error in the *v*-velocity are presented in Figure 37 for a) the "true" error (approximated using the 257×257 node mesh), b) MNP with C_2 = 0, and c) Richardson extrapolation. The MNP approach clearly provides a more accurate representation of the true error in the center of the cavity; however, Richardson extrapolation does appear to be more accurate near the two corner singularities.



a) b) c)

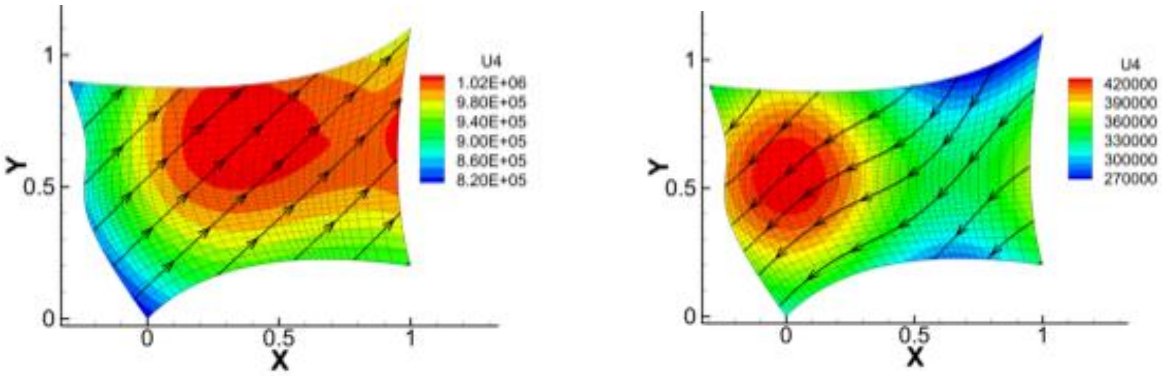
Figure 37. Contours of the discretization error in *u*-velocity for the standard driven cavity: a) true error estimated from a 257x257 grid, b) error estimate from the MNP procedure with C_2 = 0.0, and c) error estimate from Richardson extrapolation.

5.2.3. Compressible Euler Equations

The compressible Euler equations are discretized using a cell-centered finite volume method. Roe's method[29] is used for calculating the interface fluxes. MUSCL extrapolation[30] is used for a second-order accurate, fully upwinded scheme. A simple explicit scheme is used with local time stepping. All simulations were done in double precision and the iterative residuals were converged to machine zero. The code has been fully verified using the method of manufactured solutions.[2,3] Additional details on the numerical method can be found in Refs. 14 and 28.

5.2.3.1. Manufactured Solution: Curvilinear Grid A

Two different manufactured solutions are studied on a general curvilinear mesh (called Curvilinear Grid A).[28] The first is a supersonic solution and the second is a subsonic solution. The problems are identical with the exception of the exact solution, the corresponding source term, and the boundary conditions. Both solutions are shown in Figure 38a and Figure 38b, respectively. The finest mesh is 129×129 and corresponds to a mesh size parameter of $h = 1$. A family of grids is created by systematically coarsening the original grid by a factor of two (i.e. removing every other point) where the coarsest mesh is 9×9 (corresponding to $h = 16$).



a) b)

Figure 38. Manufactured solution showing the conserved variable ρe_t and streamlines on the 33x33 mesh: a) supersonic solution and b) subsonic solution.

To compare the local estimates, a slice extracted from the subsonic manufactured solution is shown in Figure 39. The defect correction methods and Richardson extrapolation compare qualitatively well with the exact error with neither showing clearly better results. All three defect corrections methods have the same trends. A similar cross-section for the supersonic manufactured solution is given in Figure 40 and shows very accurate estimates for all methods.

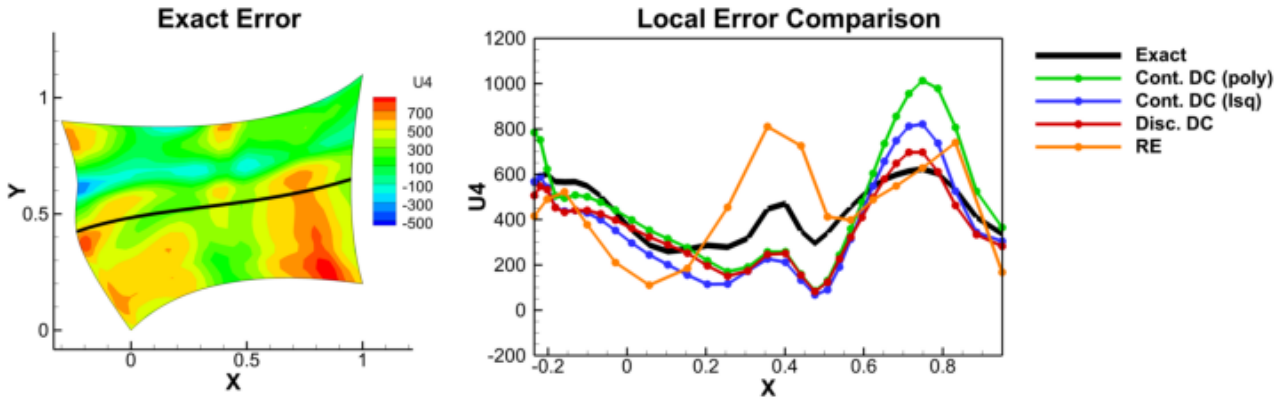


Figure 39. Slice through the subsonic manufactured solution along the j=17 node location for the 33x33 grid.

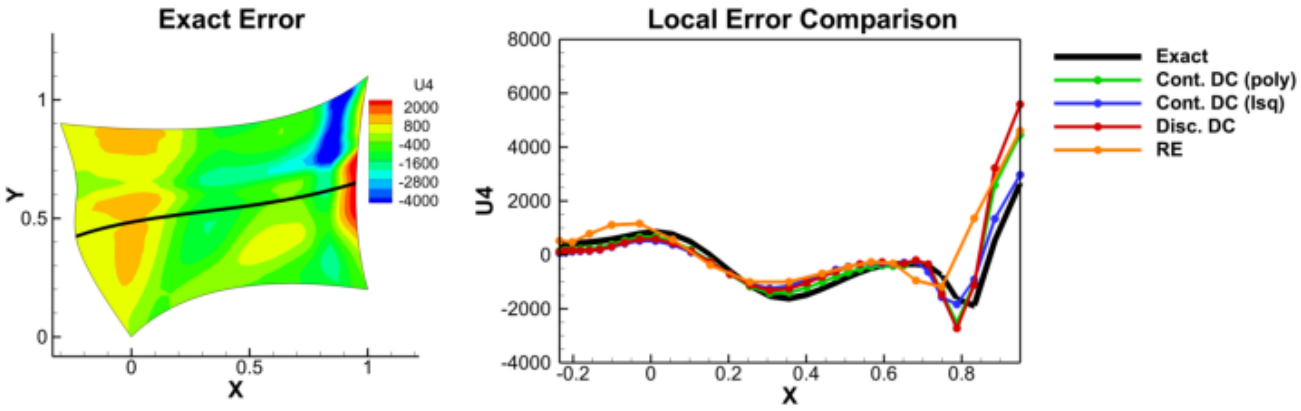


Figure 40. Slice through the supersonic manufactured solution along the j=17 node location for 33x33 grid.

5.2.3.2. Manufactured Solution: Curvilinear Grid B

Manufactured solutions[2,3] were examined on two different grids in Refs. 14 and 15 to investigate the effect of grid quality on the MNP process: a highly skewed grid (Curvilinear Grid A) and a mildly skewed grid (Curvilinear Grid B). The second grid, Curvilinear Grid B, covers a similar domain in physical space, but has less stretching, skewing and curvature than Curvilinear Grid A. Curvilinear Grid B is pictured in Figure 41.

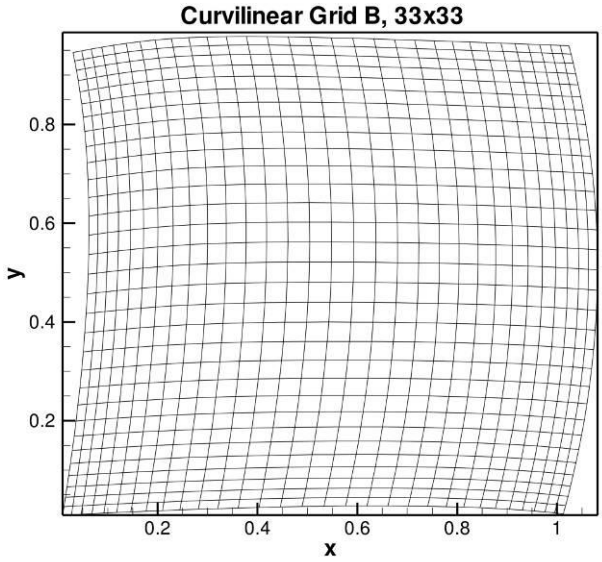


Figure 41. Curvilinear Grid B.

Figure 42 shows a comparison of the numerical solution and curve fit for pressure. The curve fits for all primitive variables are shown in Figure 43. The results are shown for a 129×129 grid, with 32 zones in each direction, 3rd order least squares fits, and C^3 weight functions. The curve fits do

a good job of matching the underlying numerical solution. The curve fit errors are presented in Figure 44 with percent error shown in Table 5. The curve fit errors are smooth which is expected as this is the nature of the numerical solution and curve fits. The percent curve fit errors are less than those for Curvilinear Grid A, which is evidence that grid quality affects the curve fit accuracy. See Ref. 14 for more details.

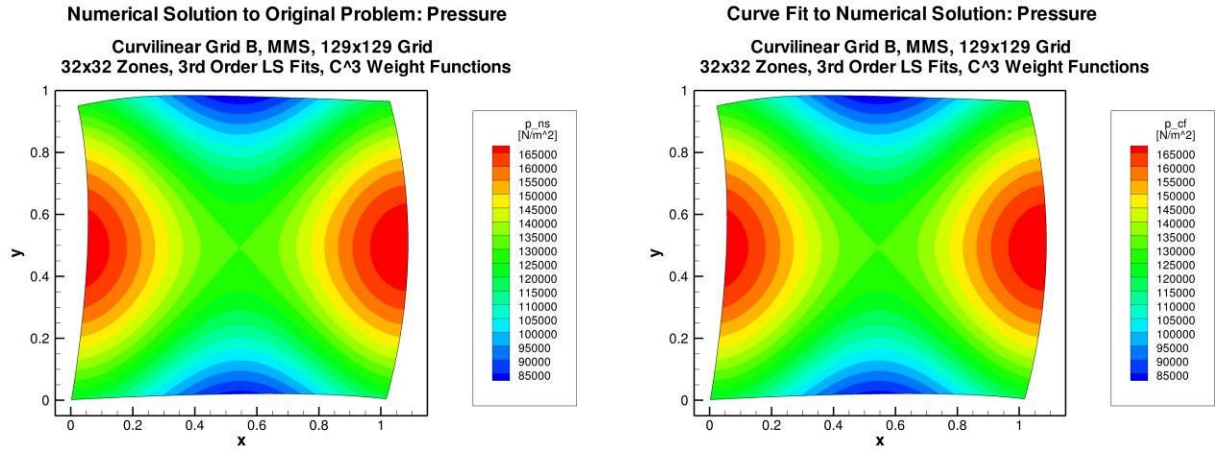


Figure 42. Curvilinear Grid B numerical solution and curve fit.

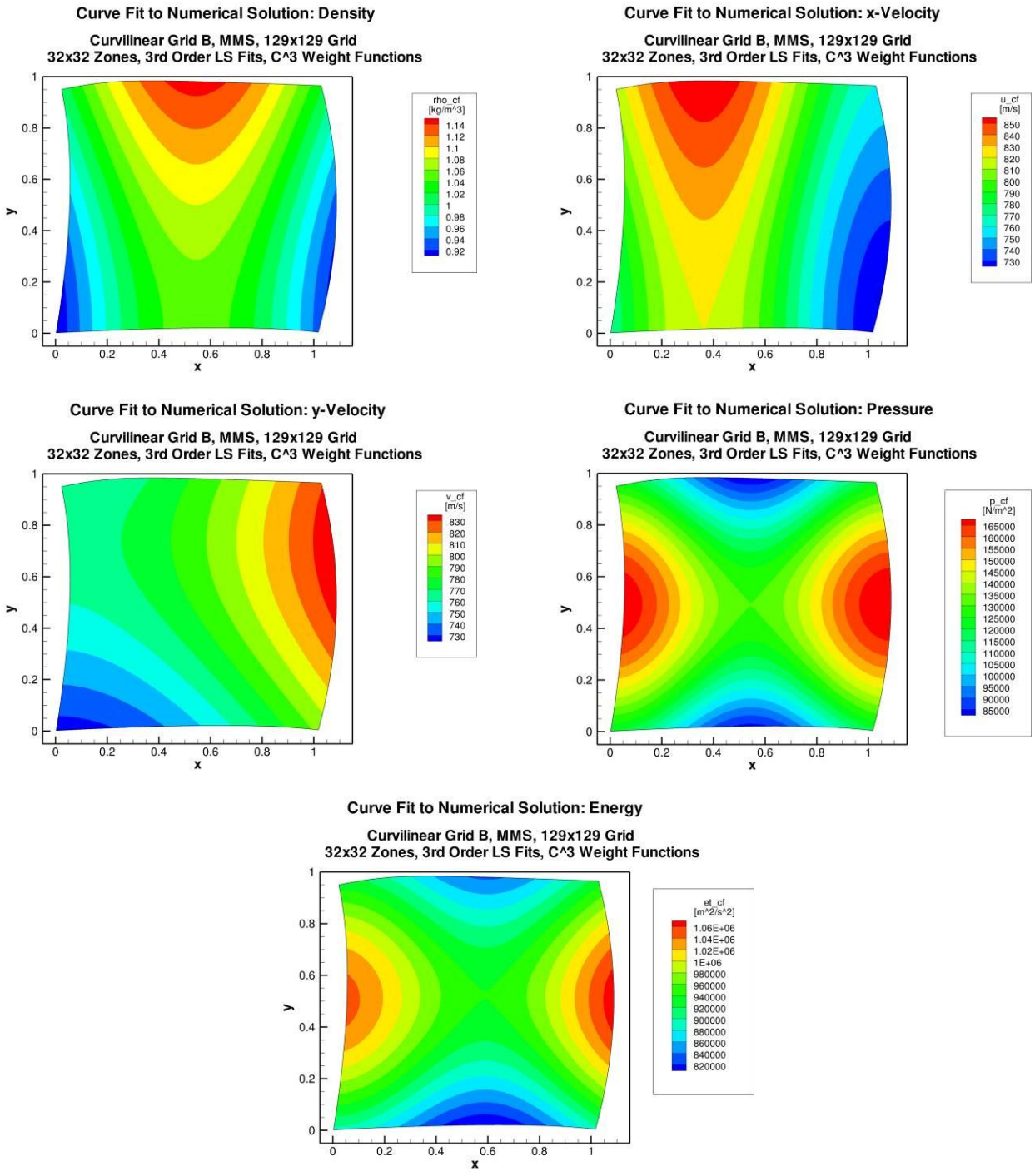


Figure 43. Curvilinear Grid B curve fits.

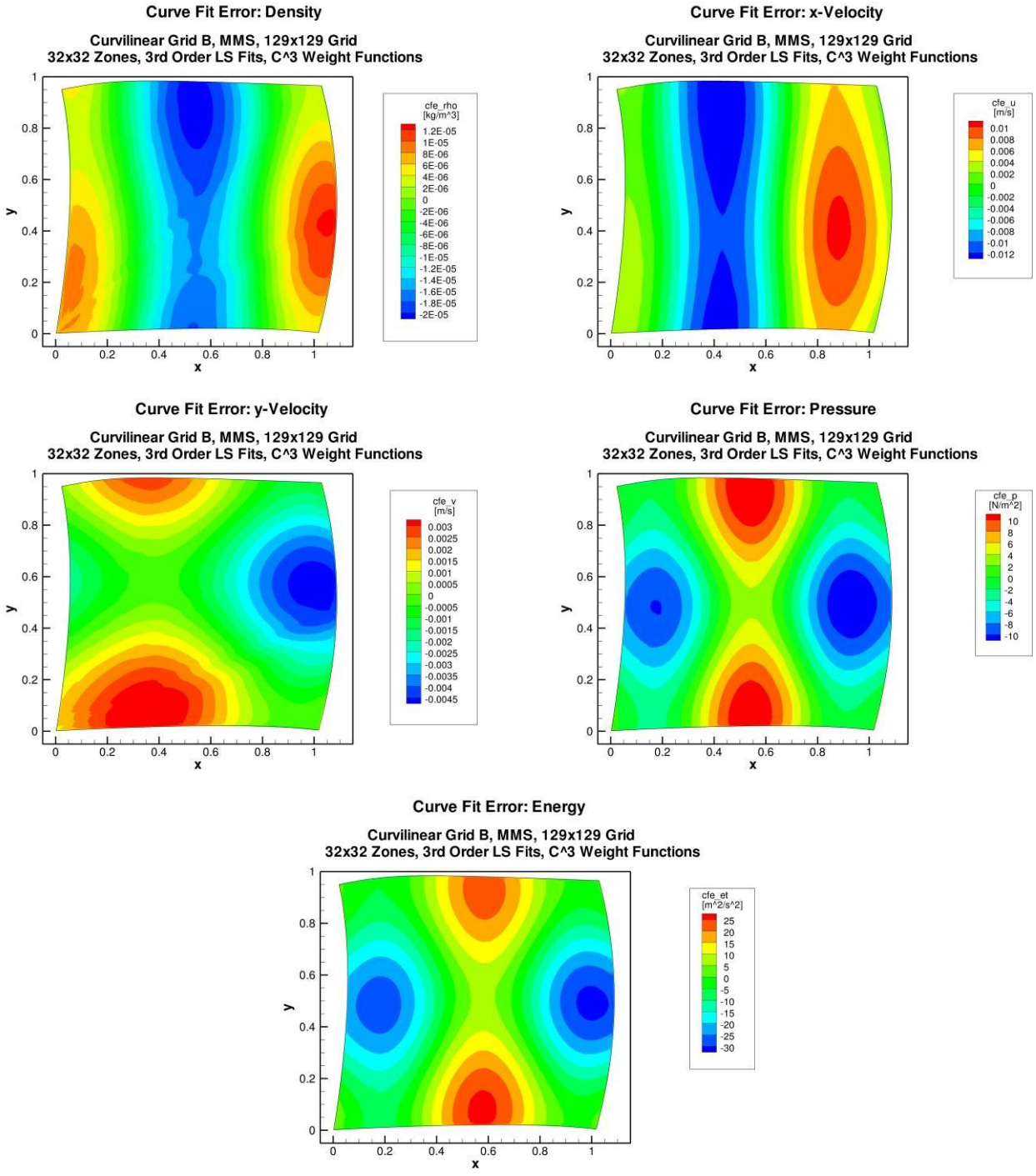


Figure 44. Curvilinear Grid B curve fit error.

Table 5. Curvilinear Grid B maximum percentage curve fit errors.

	ρ	u	v	p	e_t
% CF Error	0.00187	0.00167	0.000586	0.0152	0.00325

Extracted distributions of the curve fit and curve fit error for density are shown in Figure 45 for a slice at $y = 0.5$ m. There is a spike in the curve fit error at both boundaries, which indicates that even for a smooth solution, the boundaries of the problem are a more difficult region to fit.

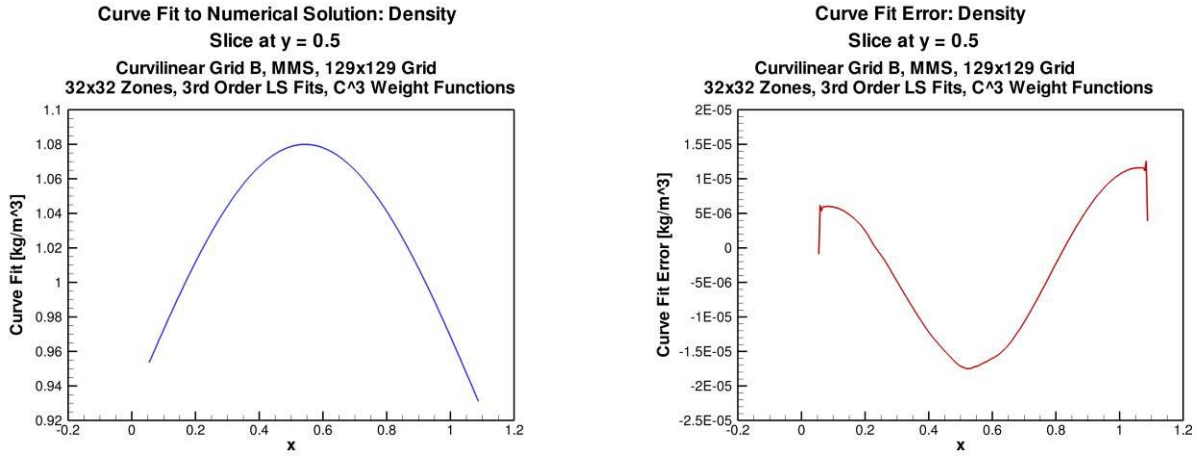


Figure 45. Curvilinear Grid B slices of curve fit and error.

The source terms in Figure 46 exhibit the greatest magnitude near the boundaries. Figure 47 is an x-y plot of the mass source term, at a slice through the domain. The spike in source term magnitude at the boundaries can be seen clearly. This is related to the curve fit error at the boundaries.

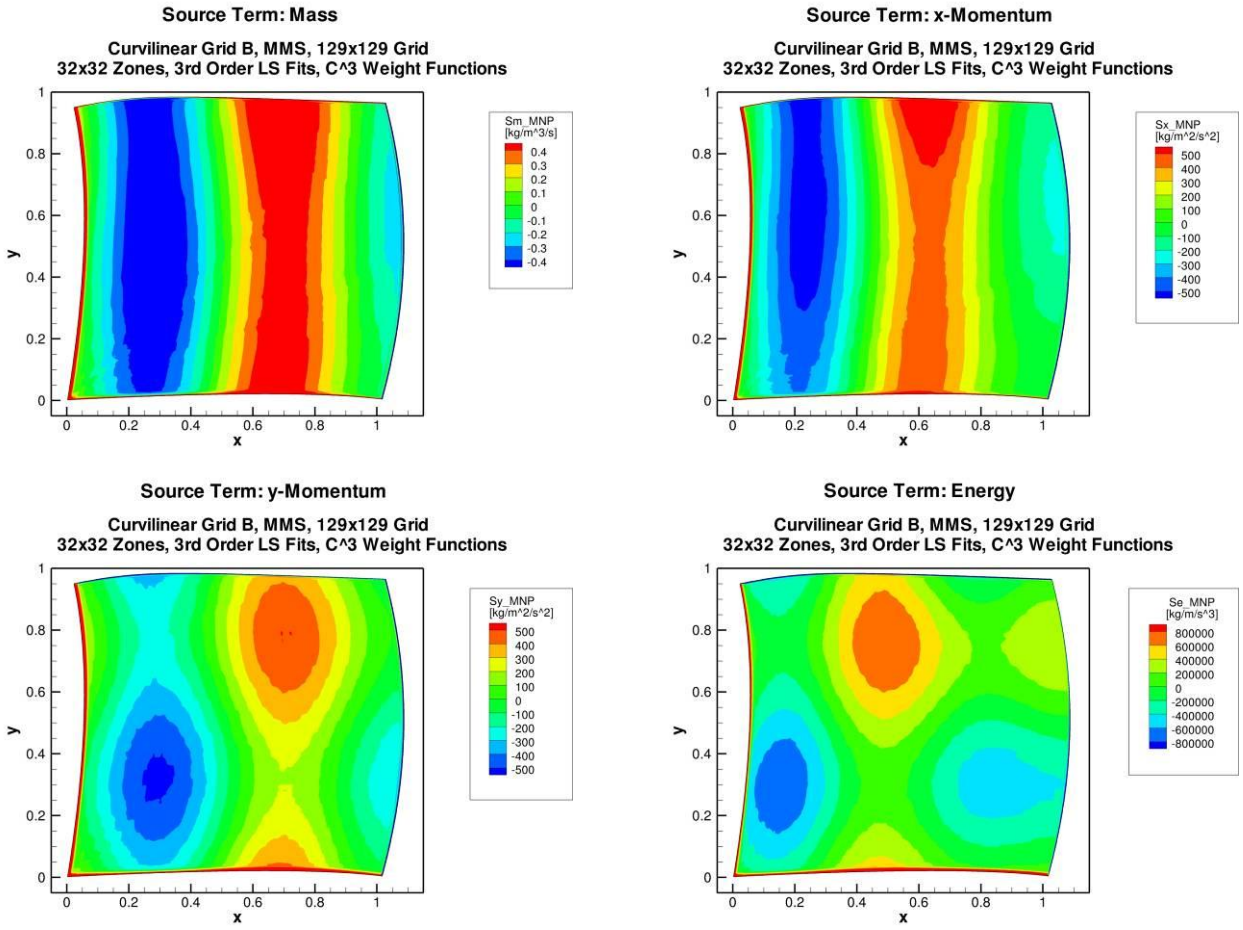


Figure 46. Curvilinear Grid B MNP source terms.

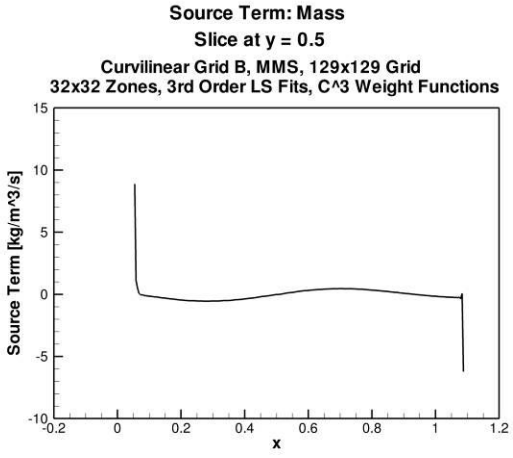
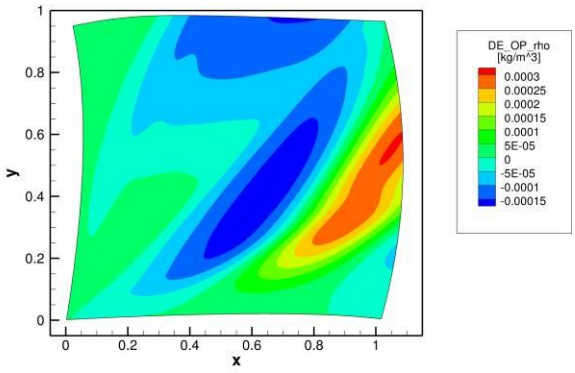


Figure 47. Curvilinear Grid B slice of mass source term.

A comparison between original problem discretization error and nearby problem discretization error is given in Figure 48. The nearby problem discretization error provides a good estimate of the error in the original problem. A slice comparison of the discretization error is provided in Figure 49.

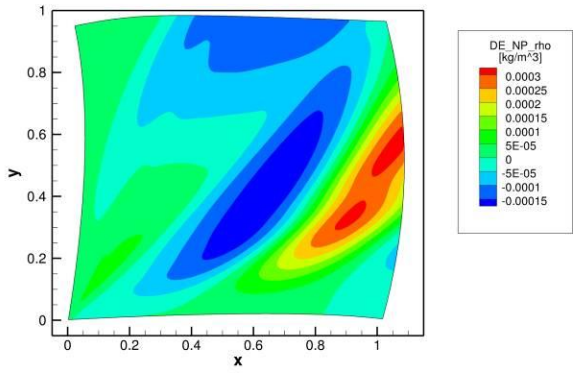
Discretization Error in Original Problem: Density

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



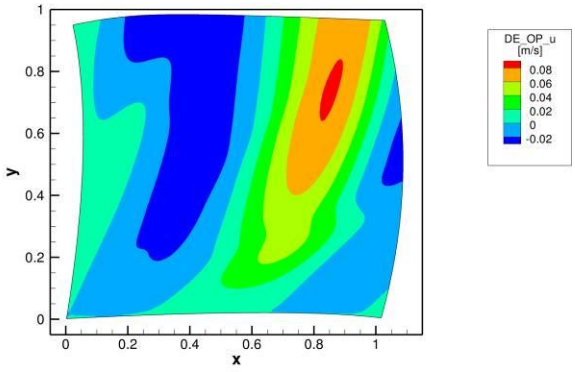
Discretization Error in Nearby Problem: Density

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



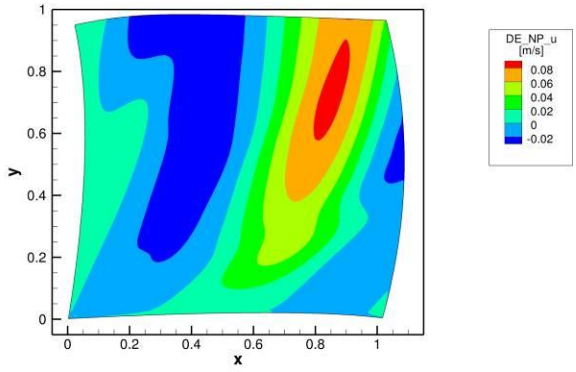
Discretization Error in Original Problem: x-Velocity

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



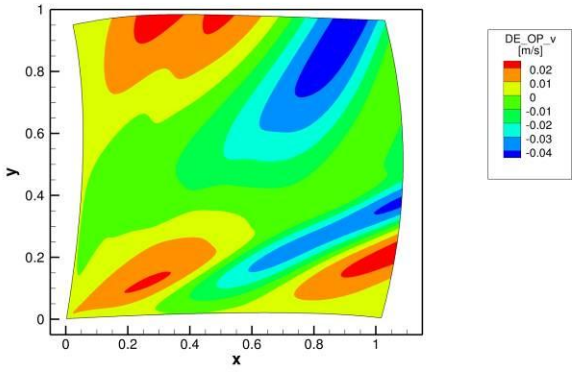
Discretization Error in Nearby Problem: x-Velocity

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



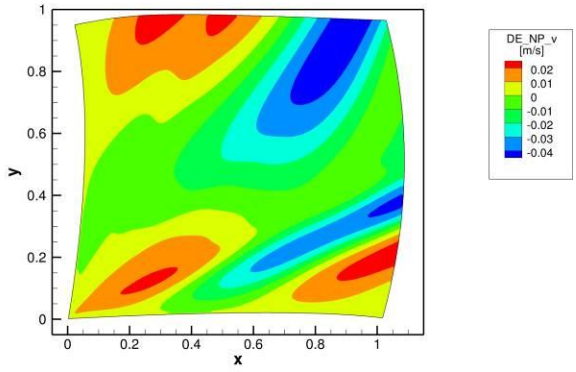
Discretization Error in Original Problem: y-Velocity

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



Discretization Error in Nearby Problem: y-Velocity

Curvilinear Grid B, MMS, 129x129 Grid
32x32 Zones, 3rd Order LS Fits, C^3 Weight Functions



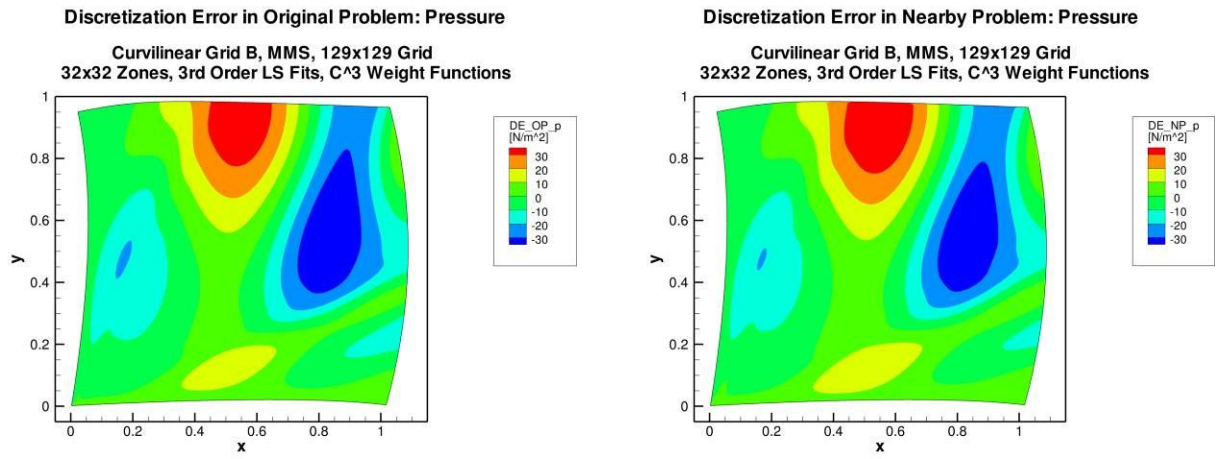


Figure 48. Curvilinear Grid B discretization error comparison: original problem (left) and nearby problem (right).

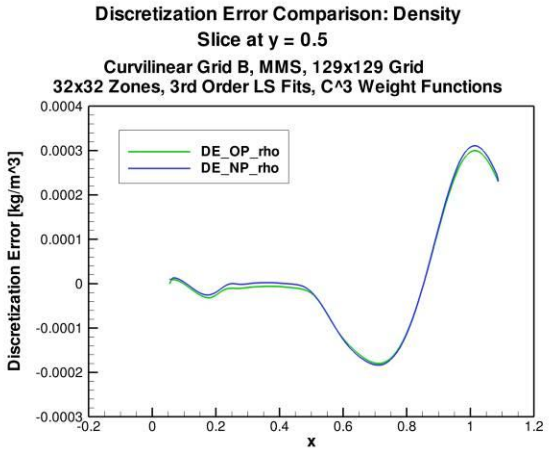


Figure 49. Curvilinear Grid B slice discretization error comparison.

The effect of refining the number of grid points and zones is shown in Figure 50. Both the previous results for Curvilinear Grid A (see Ref. 14) and the above results for Curvilinear Grid B are shown for comparison. As both grid points and zones are refined, the curve fit error and source term both drop in magnitude at a constant rate. The effect of changing only the number of zones while holding grid points constant is given in Figure 51. The curve fit error drops as the number of zones is increased, but shows the biggest drop between 8 zones and 16 zones. The source term norms give a conflicting picture of the effect of zone refinement. It appears that increasing the number of zones does not significantly decrease the source term magnitude. Even with a small number of zones, the problem may be smooth enough that using more zones simply is not necessary. Figure 52 shows the effects of changing the order of the local least squares fits. Increasing the order of the least squares fits causes the curve fit error norms and source term norms to drop, with the greatest reduction between first and second order fits. Again, with

sufficiently small zones the numerical solution will approach a linear behavior over each zone, where a higher order least-squares fit may not improve the accuracy of the fit. One important observation to note is that for most of the trends observed, the norms for Curvilinear Grid B are smaller than for Curvilinear Grid A. This indicates that grid quality has an effect on the curve fits and source terms for the MNP.

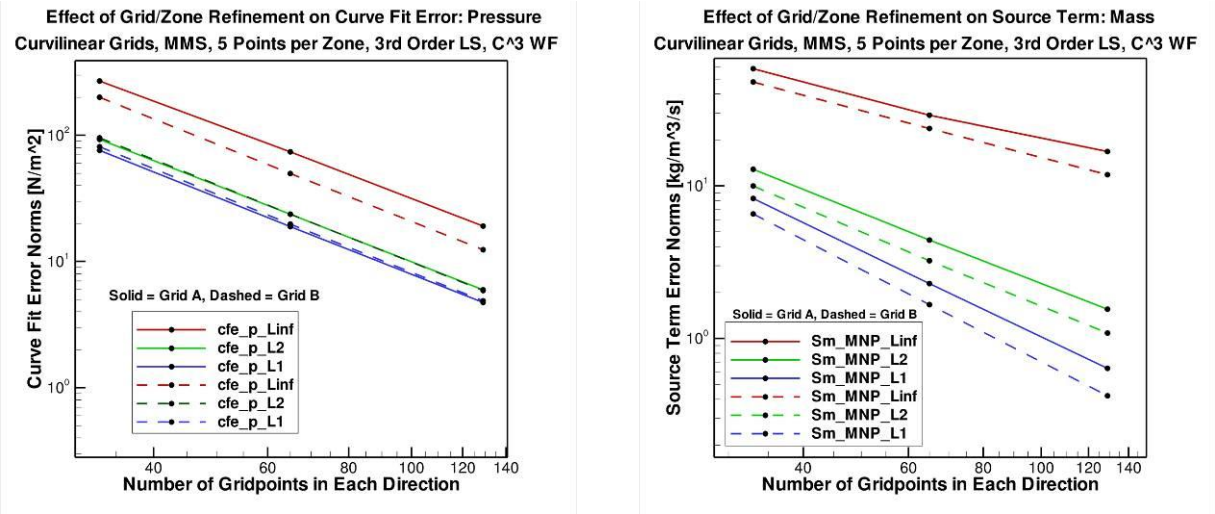


Figure 50. Curvilinear Grid B effect of grid/zone refinement.

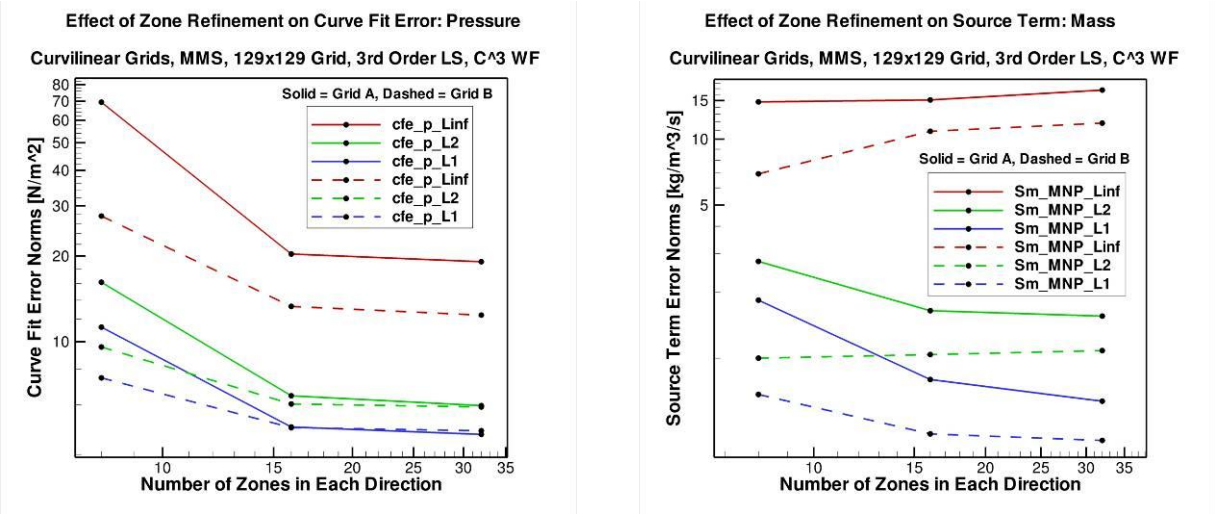


Figure 51. Curvilinear Grid B effect of zone refinement only.

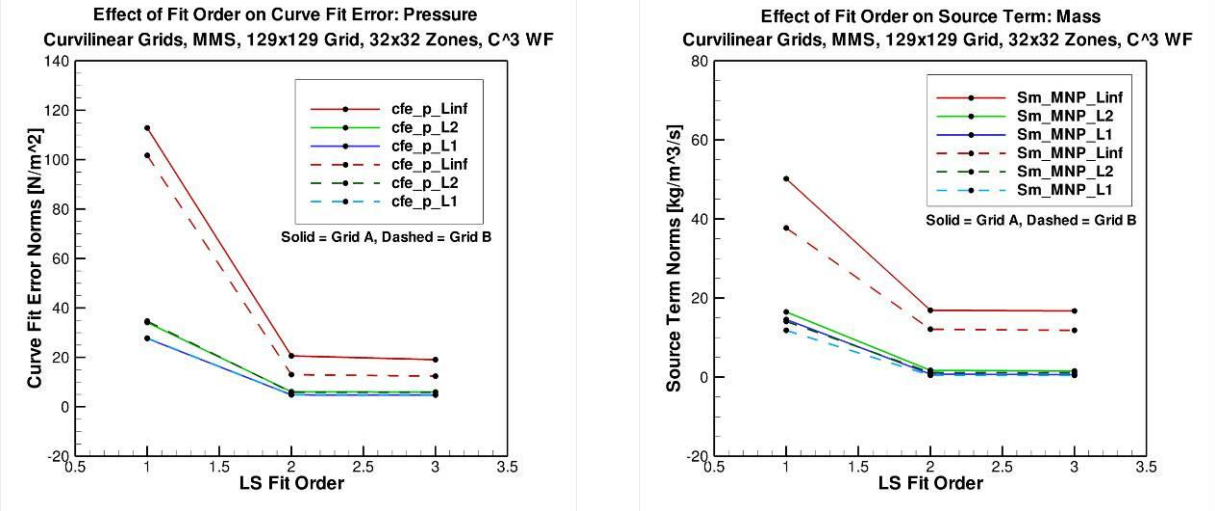


Figure 52. Curvilinear Grid B effect of least squares fit order.

5.2.3.3. Supersonic Vortex

The supersonic vortex problem is solved on a series of five systematically-refined meshes.[28] The finest grid is 257×129 and is refined by a factor of 2 to the coarsest grid of 17×9. The conserved energy variable, streamlines, and a coarse grid are shown in Figure 53.

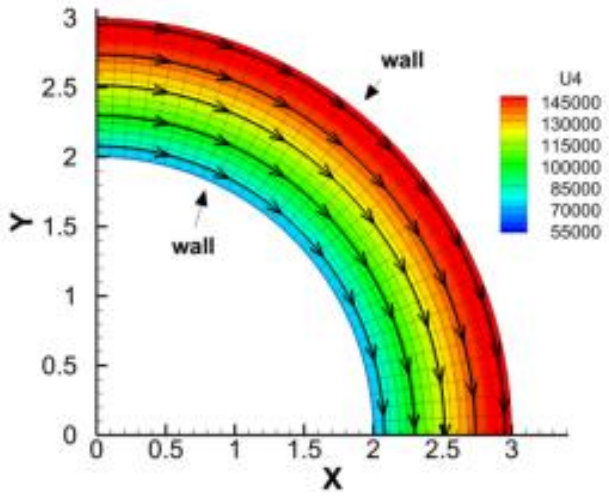


Figure 53. Supersonic vortex flow showing the conserved variable ρe_t and streamlines on the 33x17 mesh.

The true discretization error for this problem is shown in Figure 54 as both a contour plot and as an extracted x-y plot along the outer wall. The defect correction methods do not provide accurate representations of the discretization error for this case, possibly due to deficiencies in the boundary conditions. All three defect correction methods show almost identical result with large oscillations in the error estimate compared to the true error. Contours of the discretization error

for the defect correction methods, Richardson extrapolation, and the true error are shown in Figure 55. Oscillations are also evident along the two inviscid walls.

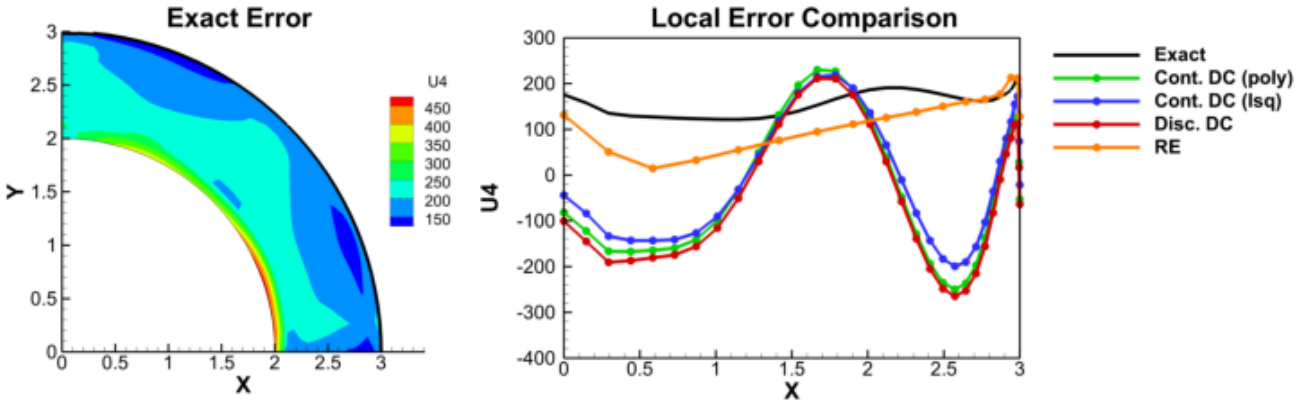


Figure 54. Slice along the outer wall for supersonic vortex flow for 33x17 grid.

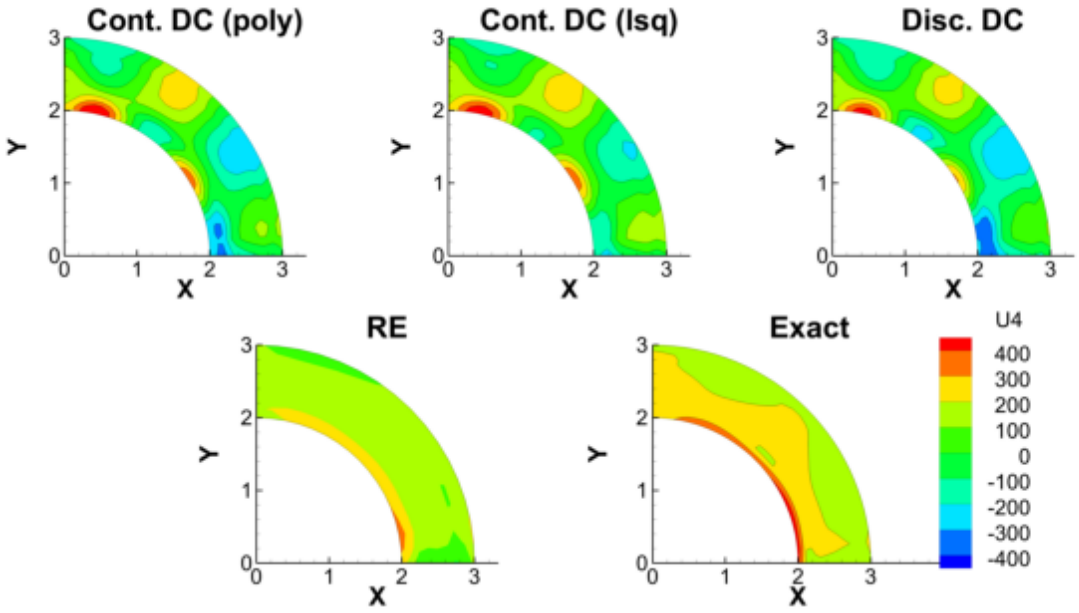


Figure 55. Contour plots of error for supersonic vortex flow on the 33x17 mesh.

5.2.3.4. Ringleb's Flow

Ringleb's flow (see Figure 56) is computed on three systematically-refined meshes.[28] The finest grid is 65×65 and is coarsened by a factor of 2 to the coarsest grid of 17×17. Similar to supersonic vortex flow, the bounding streamlines are modeled as a wall. While Ringleb's flow and supersonic vortex flow are both fully supersonic problems with non-penetration boundary conditions, the defect correction error estimates performed very differently as illustrated by the slice along the outer wall for supersonic vortex flow shown in Figure 54 compared to a slice

through the Ringleb's flow shown in Figure 57. Ringleb's flow matches the true error very well and performs significantly better than Richardson extrapolation. These results are further confirmed by examining contour plots of the discretization error given in Figure 58.

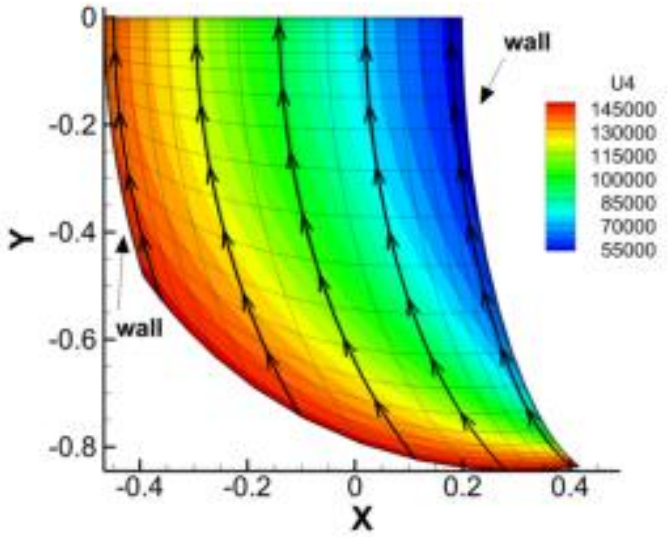


Figure 56. Ringlebs flow solution showing the conserved variable ρe_t and streamlines on the 33x33 mesh.

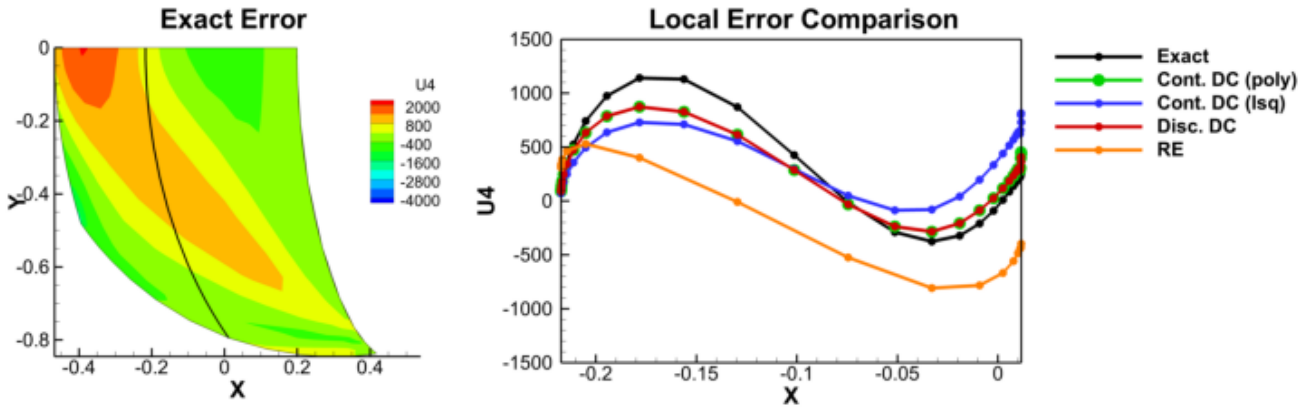


Figure 57. Slice through Ringleb's flow on the 33x33 grid.

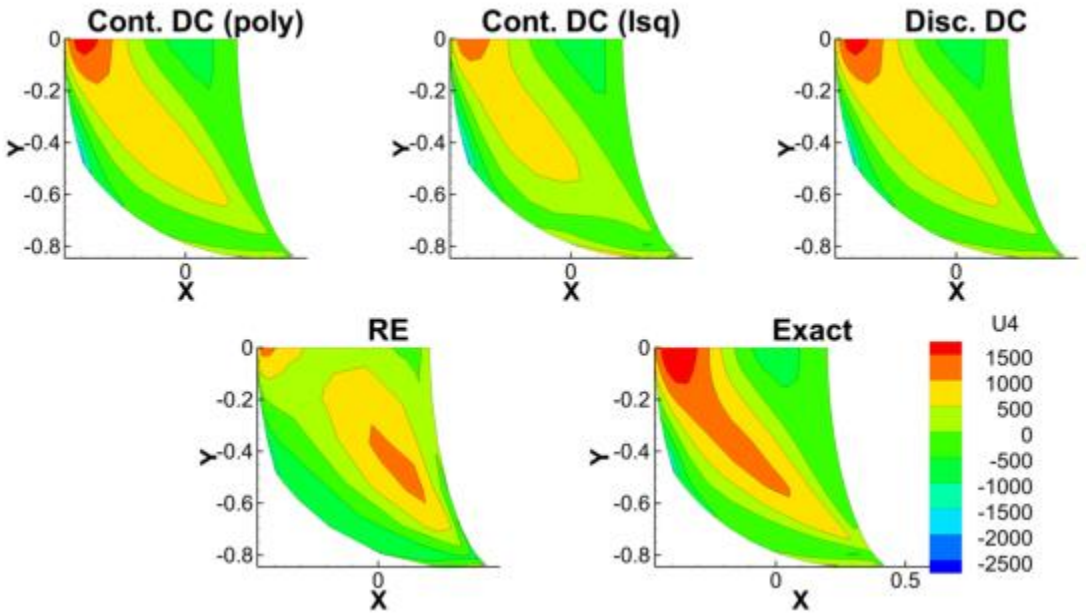


Figure 58. Contour plot of discretization error for Ringleb's flow on the 33x33 mesh.

5.2.3.5. Subsonic Airfoil

The Euler equations are used to solve for the inviscid flow over a subsonic airfoil[14,15] which, in this case, has no exact solution. The airfoil is run at sea level atmospheric conditions with a Mach number of 0.5 and 4 degrees angle of attack. The domain is decomposed using a structured "C" shaped grid, which wraps around the airfoil. The fine grid is 129×257 and is shown in Figure 59.

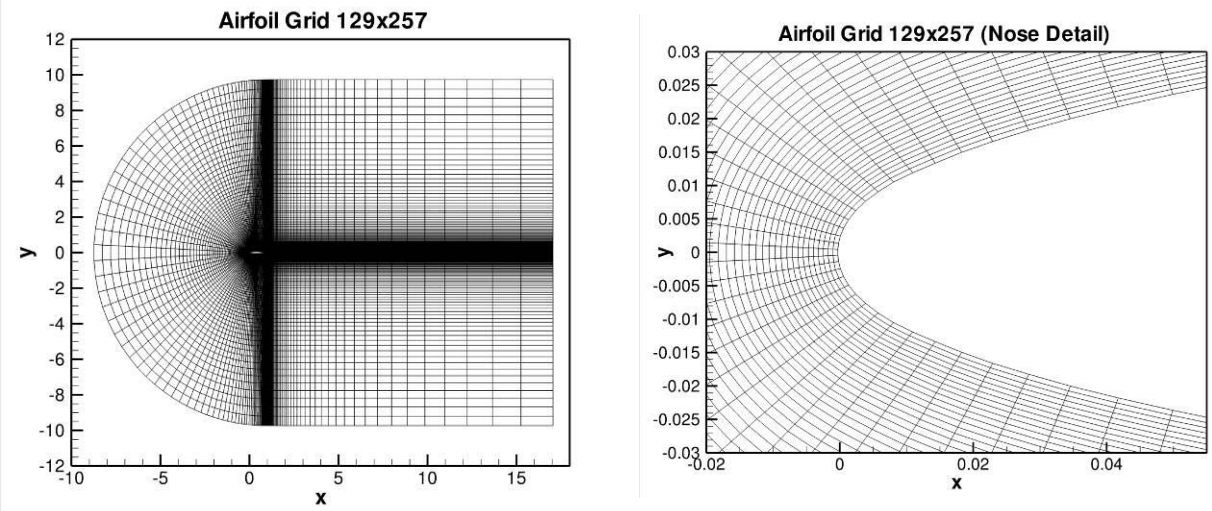
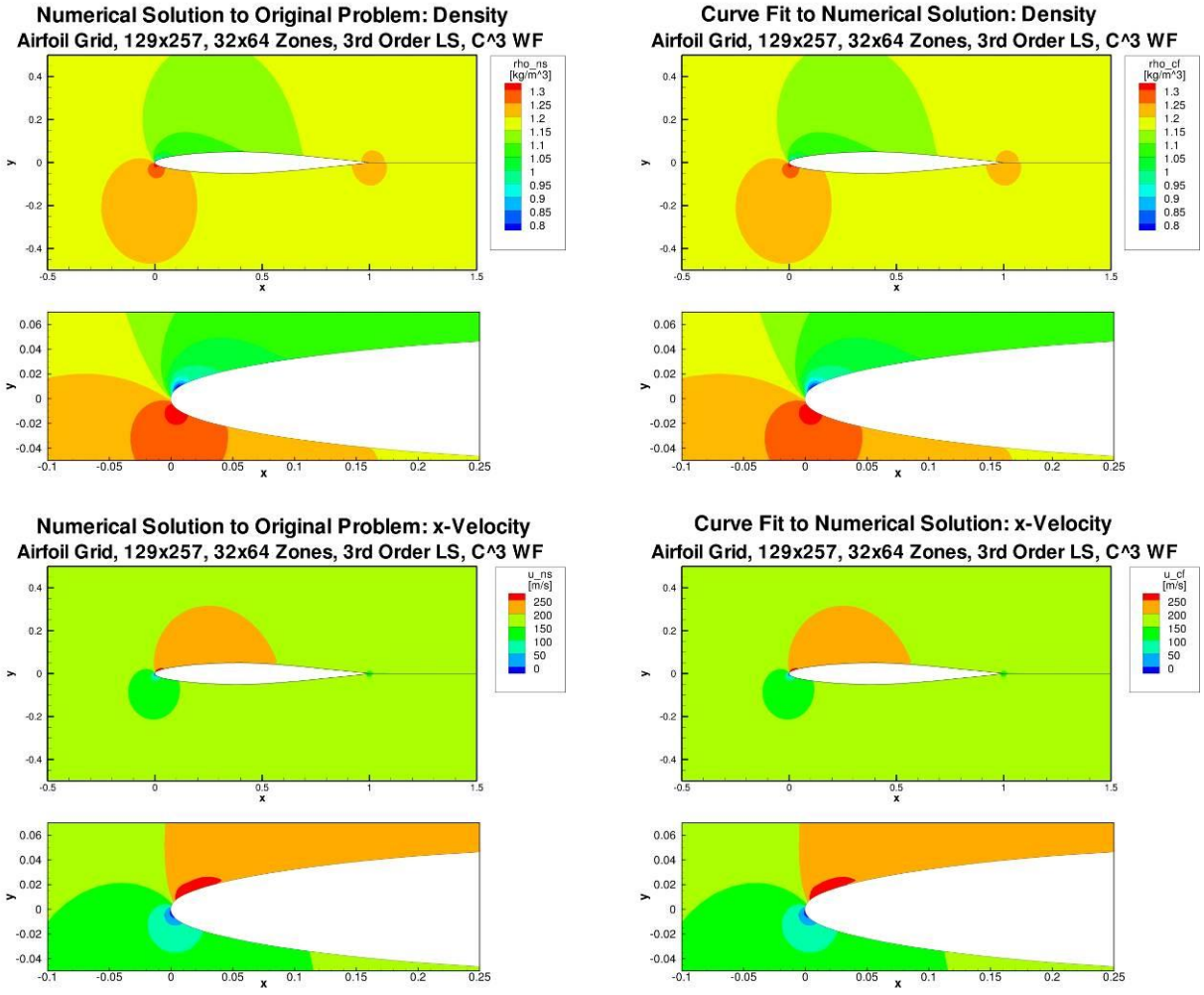


Figure 59. Airfoil grid.

The curve fits of the primitive variables and the energy term are presented in Figure 60. The curve fit errors are presented in Figure 61. These plots show the domain around the airfoil as well as a close-up view of the nose region. The curve fit error is largest near the top of the leading edge ($x = 0$). The curve fit error plots use different contour levels between the overview and close-up to show the behavior throughout the domain. Large oscillations are seen in the plots of the curve fitting errors, possibly due to the large gradients present near the airfoil boundary. Maximum absolute percent curve fit errors are given in Table 6 for density, pressure, and energy. The velocities are omitted from the table because they both cross through zero at some point in the domain which causes large percent curve fit errors even though the absolute error is not necessarily large.



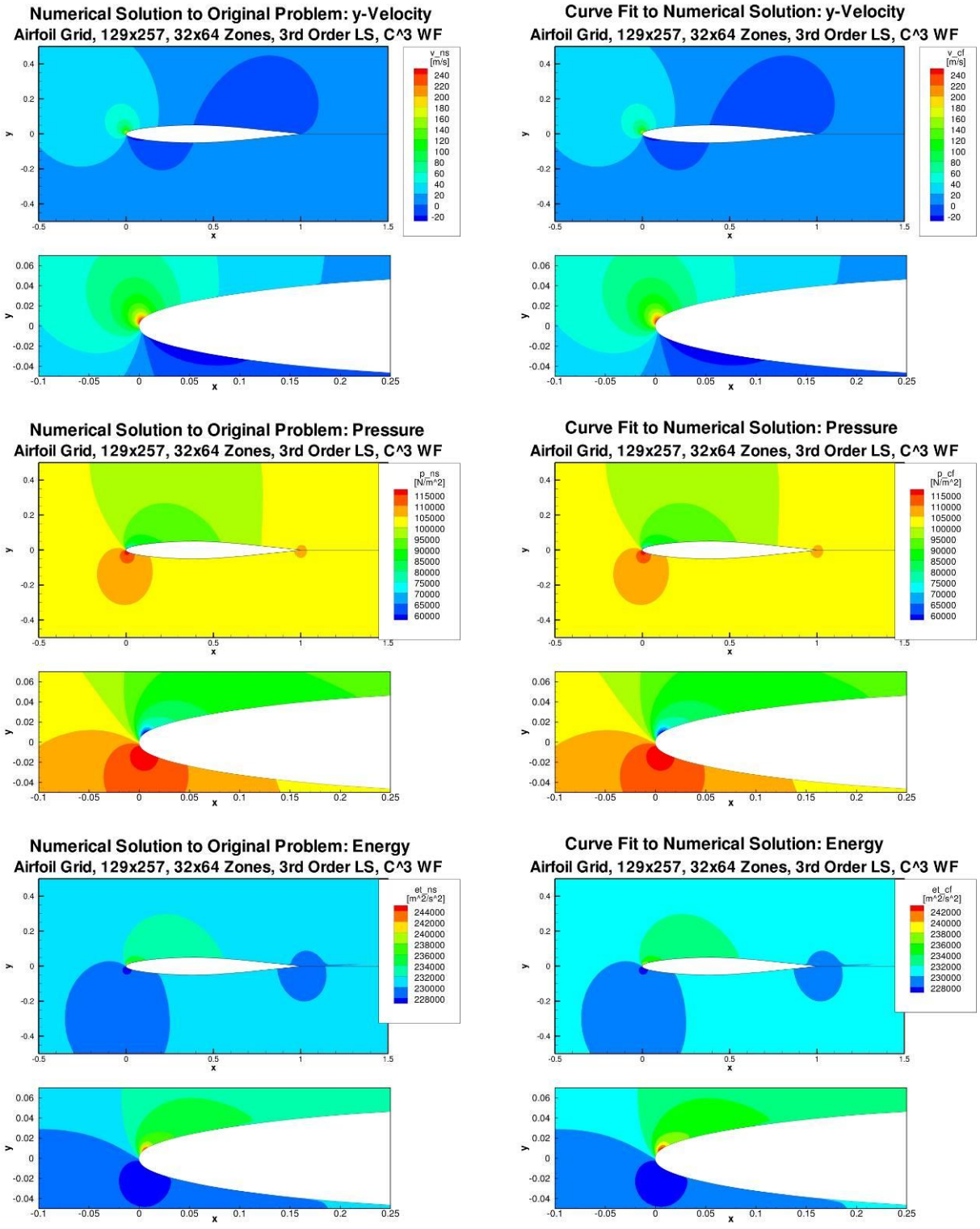


Figure 60. Airfoil numerical solution (left) and curve fits (right).

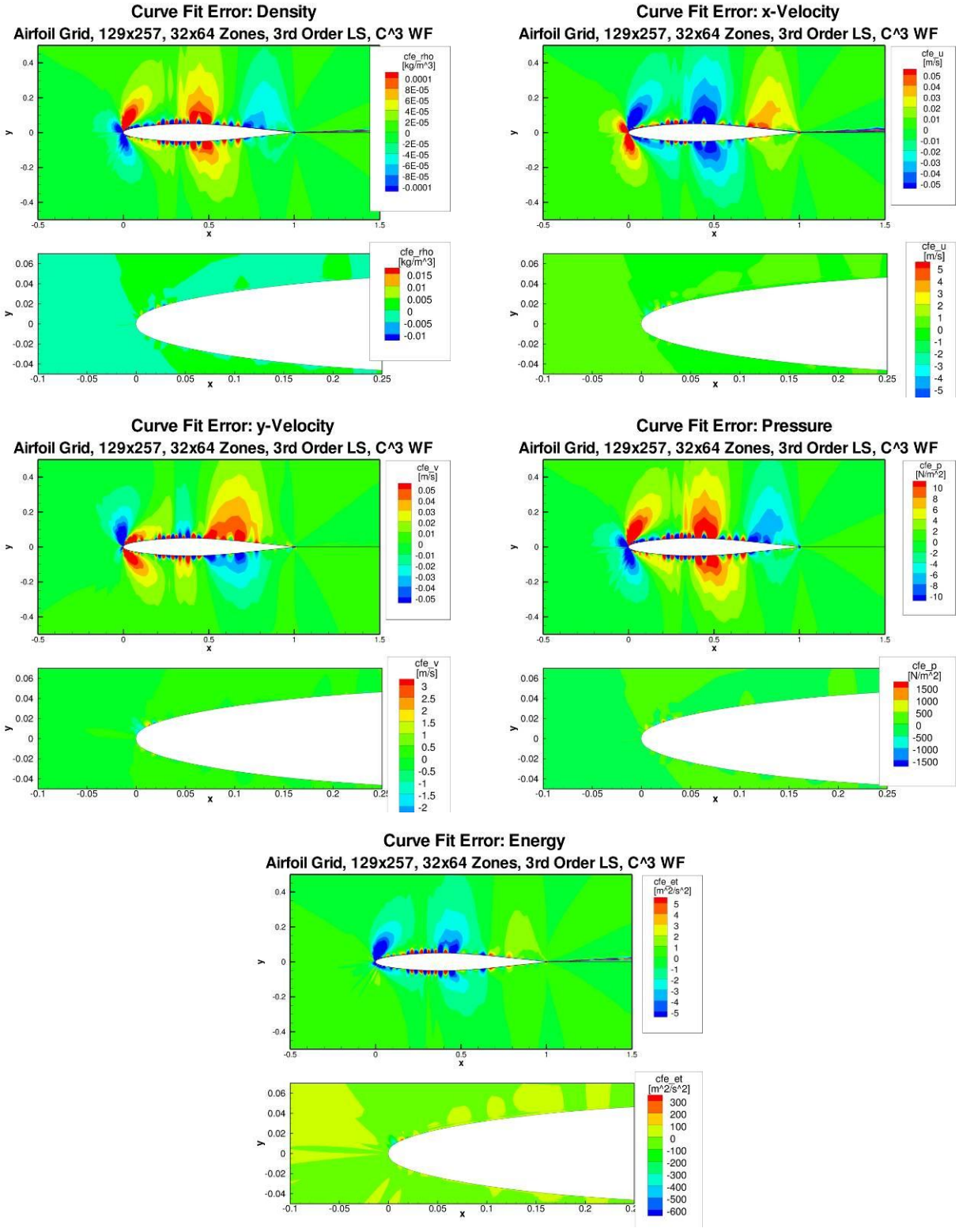


Figure 61. Airfoil curve fit errors.

Table 6. Subsonic airfoil maximum percentage absolute curve fit errors

	ρ	u	v	p	e_t
% CF Error	1.981	-	-	2.741	0.2878

The source terms are shown in Figure 62. The source terms are concentrated at the surface of the airfoil, which is expected as this is where the solution changes most rapidly. This causes some curve fitting errors which translate into source terms. Some cells near the nose of the airfoil have significantly higher values than anywhere else within the domain.

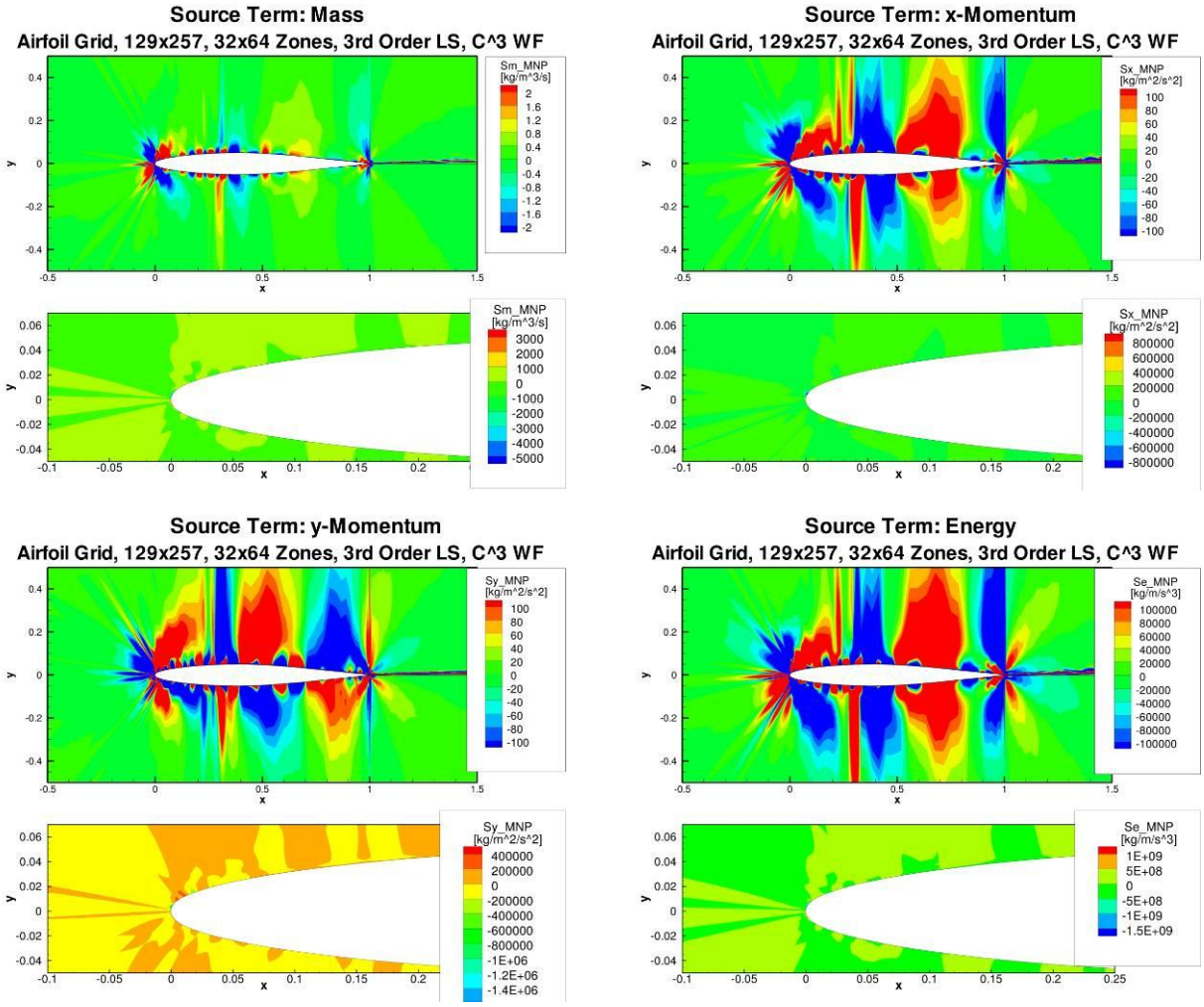


Figure 62. Airfoil MNP source terms.

The exact discretization error in the original problem cannot be calculated, as the exact solution to the original problem in not known. The discretization error in the nearby problem is shown in Figure 63, but is likely to be a poor approximation of the actual discretization error due to the large curve fitting errors and MNP source terms.

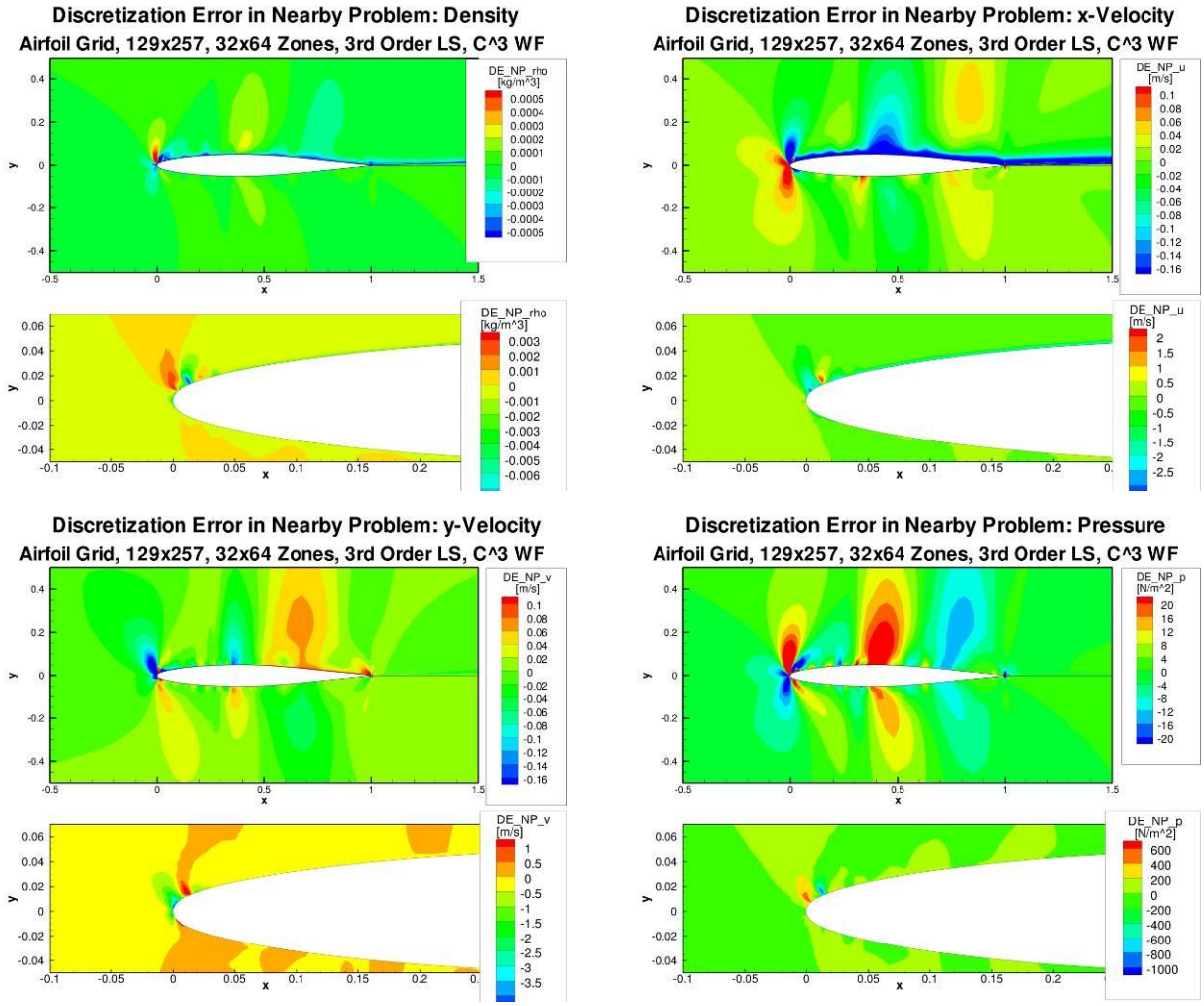


Figure 63. Airfoil nearby problem discretization errors.

Several x-y line plots are shown in Figure 64, showing the pressure discretization error for a constant $y = 0$. One is an enlargement of the interesting behavior, while the other shows an overview. There is a large oscillation in the discretization error just before the flow reaches the surface of the airfoil. This is not unexpected as there is a sharp pressure gradient here, with the stagnation point close by. The region around the stagnation point is a difficult region to resolve with the current, relatively coarse grid. There is no data shown behind the airfoil, due to the nature of the grid. The values plotted are cell-centered. Another set of x-y line plots are shown in Figure 65. This time the pressure discretization error is shown for a constant *x/c* = 0.05. There are some oscillations as near the surface of the airfoil, as expected.

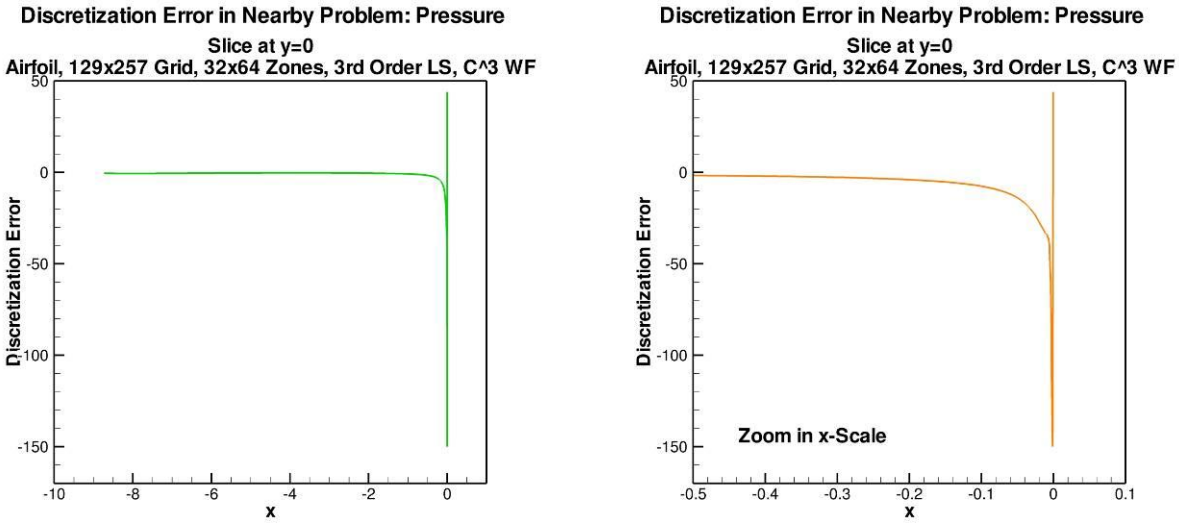


Figure 64. Airfoil x-y plot of pressure discretization error (slice at y=0).

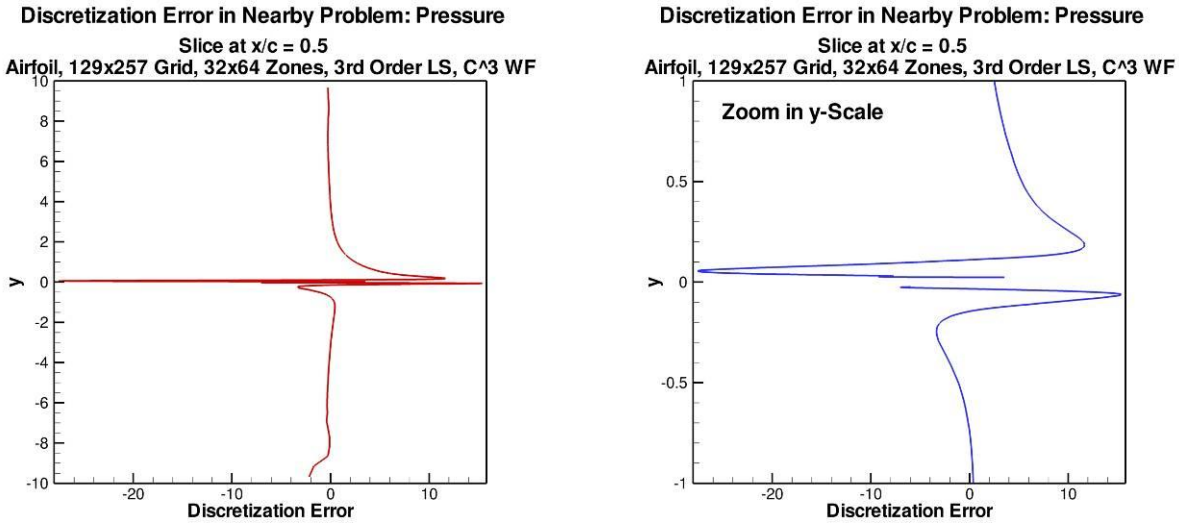


Figure 65. Airfoil x-y plot of pressure discretization error (slice at x/c=0.05).

Richardson extrapolation was also performed on the airfoil problem. This calculation uses the fine grid (129×257) and coarse grid (65×129), for a refinement factor of two. The formal order of accuracy of two was also used. Equation (32) shows the expression for the Richardson extrapolation error estimate in the fine solution

$$f_h - \bar{f} = -\frac{f_h - f_{rh}}{r^p - 1} \tag{32}$$

where $\bar{f}$ is the estimated exact solution, h is the fine grid spacing parameter, r is the grid refinement factor, and p is the formal order of accuracy. It follows that f_h is the fine grid solution and f_{rh} is the coarse grid solution. The Richardson extrapolation error estimates are

shown in Figure 66. The Richardson extrapolation error estimates seem to be on the same order of magnitude as the MNP nearby problem discretization errors. Richard extrapolation also suggests that the location of maximum discretization error tends to be on the upper surface of the airfoil leading edge.

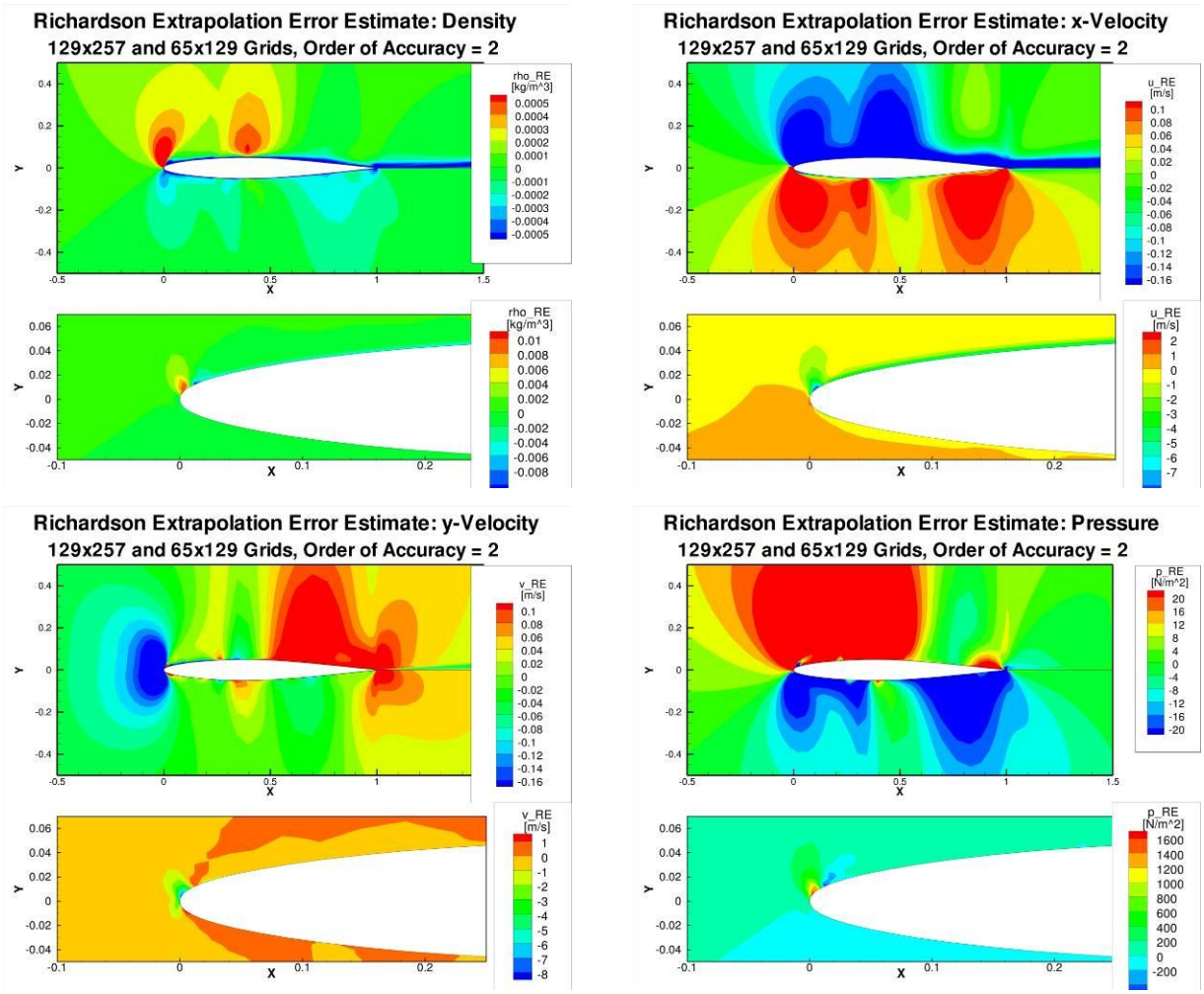


Figure 66. Airfoil Richardson extrapolation error estimate.

6. CONCLUSIONS

The Method of Nearby Problems (MNP) has been developed to generate exact solutions to partial differential equations and as a single grid discretization error estimator. When used in the generation of exact solutions, local spline-type fits were required to overcome limitations of global curve fitting. A weighting function approach was further developed which uses a series of overlapping local fits which are combined together using the weighting functions. These weighting functions can be tailored to provide arbitrary levels of continuity along the spline fitting boundaries. While this approach is extendable to an arbitrary number of dimensions (which may be useful for optimization or uncertainty quantification), the tool we developed is capable of handling four-dimensional problems (i.e.., three spatial dimensions and time). While the tool is limited to Cartesian domains, curvilinear structured grid problems are addressed by formulating the fitting in a Cartesian computational space. Initial testing of this approach in one- and two-dimensions indicated that regions of high gradients, especially near the domain boundaries, led to oscillations in the spline approximation. In order to overcome this problem, additional work is needed on structured mesh adaptation and advanced local fitting procedures.

A residual-based error estimation framework was developed based on the Generalized Truncation Error Expression (GTEE). This framework is particularly useful for developing residual-based error estimators for finite difference and finite volume methods. Within this framework, a number of residual-based error estimators can be examined including defect correction, error transport equations, and adjoint methods. The defect correction approach was developed in both continuous and discrete forms. MNP is in fact a form of continuous defect correction where numerical solutions are fit in a continuous manner before being operated on by the partial differential equation (PDE). However, when the goal is discretization error estimation, continuity is not required between the local fits and simpler, local fits can be used.

We have examined MNP/defect correction for discretization error estimation on a series of problems including 1D and 2D Burgers' equation, the compressible Euler equations, and the incompressible Navier-Stokes equations. The error estimation results were quite promising, especially with only local fits were used (i.e., when the weighting functions were omitted). The key advantage to MNP/defect correction is that it only requires a single additional solution on the same grid, as opposed to approaches such as Richardson extrapolation which require numerical solutions on multiple grids which have been systematically refined. Issues still remaining to be addressed include extensions to unstructured grids (which are now possible since the weighting functions are no longer deemed necessary) and the proper treatment of the truncation error at the boundaries.

7. REFERENCES

1. C. J. Roy, "Review of Code and Solution Verification Procedures for Computational Simulation," *Journal of Computational Physics*, Vol. 205, No. 1, 2005, pp. 131-156.
2. Oberkampf, W. L. and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, Cambridge, 2010.
3. P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, New Mexico, 1998.
4. D. A. Venditti and D. L. Darmofal, "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22-46.
5. C. J. Roy and M. M. Hopkins, "Discretization Error Estimates using Exact Solutions to Nearby Problems," AIAA 2003-0629, invited paper for session on Uncertainty in CFD, AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 6-9, 2003.
6. R. D. Skeel, "Thirteen Ways to Estimate Global Error," *Numerische Mathematik*, Vol. 48, No. 1, 1986, pp. 1-20.
7. Raju, A., Roy, C. J., and Hopkins, M. M., "On the Generation of Exact Solutions using the Method of Nearby Problems," AIAA Paper 2005-0684, 2005.
8. O'Neal, P. V., *Advanced Engineering Mathematics, 3rd Ed.*, Wadsworth Publishing Co., Belmont, California, 1991, pp. 400-412.
9. Lee, S., and Junkins, J. L., "Construction of Benchmark Problems for Solution of Ordinary Differential Equations," *Journal of Shock and Vibration*, Vol. 1, No. 5, 1994, pp. 403-414.
10. Mullges, G. E., and Uhlig, F., *Numerical Algorithms with Fortran*, Springer-Verlag, Berlin, 1996.
11. C. J. Roy, A. Raju, and M. M. Hopkins, "Estimation of Discretization Errors using the Method of Nearby Problems," *AIAA Journal*, Vol. 45, No. 6, June 2007 (see also A. Raju, C. J. Roy, and M. M. Hopkins, "Evaluation of Discretization Error Estimators using the Method of Nearby Problems," AIAA Paper 2005-4993, 2005).
12. J. R. Jancaitis and J. L. Junkins, "Modeling in n Dimensions Using a Weighting Function Approach," Journal of Geophysical Research, Vol. 79, No. 23, 1974, pp. 3361-3363.
13. J. L. Junkins, G. W. Miller, and J. R. Jancaitis, "A Weighting Function Approach to Modeling of Irregular Surfaces," *Journal of Geophysical Research*, Vol. 78, No. 11, April 1973, pp. 1794-1803.
14. Kurzen, M. J., "Discretization Error Estimation and Exact Solution Generation Using the 2D Method of Nearby Problems," Masters Thesis, Virginia Tech, Feb. 2010.
15. M. J. Kurzen, T. S. Phillips, C. J. Roy, and A. J. Sinclair, "Method of Nearby Problems for Generating Exact Solutions to 1D Unsteady and 2D Steady Problems," AIAA Paper 2009-3652, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, Texas, June 2009.

16. C. J. Roy, "Strategies for Driving Mesh Adaptation in CFD," AIAA 2009-1302, invited paper for session on Error Estimation and Control, 47th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 5-8, 2009.
17. Zadunaisky, P.E. (1966). "A Method for the Estimation of Errors Propagated in the Numerical Solution of a System of Ordinary Differential Equations," in Proceedings of the Astronautical Union, Symposium No. 25, New York, Academic Press.
18. Stetter, H.J. (1978). "The Defect Correction Principle and Discretization Methods," Numerische Mathematik, Vol. 29, pp. 425-443.
19. Layton, W., Lee, H.K., and Peterson, J. (2002). "A Defect-Correction Method for the Incompressible Navier-Stokes Equations," Applied Mathematics and Computation, Vol. 129, pp. 1-19.
20. Naumovich, A., Forster, M., and Dwight, R. (2010). "Algebraic Multigrid within Defect Correction for the Linearized Euler Equations," Numerical Linear Algebra with Applications, Vol. 17, pp. 307-324.
21. C. J. Roy and A. J. Sinclair, "On the Generation of Exact Solutions for Evaluating Numerical Schemes and Estimating Discretization Error," *Journal of Computational Physics*, Vol. 228, No. 5, 2009, pp. 1790-1802 (DOI: 10.1016/j.jcp.2008.11.008).
22. Benton, E. R., and Platzman, G. W., "A Table of Solutions of the One-Dimensional Burger's Equation," *Quarterly of Applied Mathematics*, Vol. 30, 1972, pp. 195-212.
23. Ollivier-Gooch, C., Nejat, A., Michalak K., "Obtaining and Verifying High-Order Unstructured Finite Volume Solutions to the Euler Equations," *AIAA Journal*, Vol. 47, No. 9, 2009, pp. 2105-2120.
24. Satav, V., Hixon, R., Nallasamy, M., Sawyer, S., "Validation of a Computational Aeroacoustics Code for Nonlinear Flow about Complex Geometries Using Ringleb's Flow," Presented at the 11th AIAA/CEAS Aeroacoustics Conference, 23-25 May 2005, Monterey, California.
25. A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, No. 1, 1967, pp. 12-26.
26. D. Tafti, "Alternate Formulations for the Pressure Equation Laplacian on a Collocated Grid for Solving the Unsteady Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 116, No. 1, 1995, pp. 143-153.
27. F. Sotiropoulos and S. Abdallah, "The Discrete Continuity Equation in Primitive Variable Solutions of Incompressible Flow," *Journal of Computational Physics*, Vol. 95, No. 1, 1991, pp. 212-227.
28. T. S. Phillips and C. J. Roy, "Residual Methods for Discretization Error Estimation," AIAA Paper 2011-3870, 20th AIAA Computational Fluid Dynamics Conference, June 27-30, 2011, Honolulu, Hawaii.
29. P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

30. B. van Leer, “Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov’s Method,” *Journal of Computational Physics*, Vol. 32, 1979, pp. 101-136.

APPENDIX A: PUBLICATIONS PRODUCED

The following publications were produced due to funding (either partial or full) from this effort.

Journal Articles

1. C. J. Roy, A. Raju, and M. M. Hopkins, "Estimation of Discretization Errors using the Method of Nearby Problems," *AIAA Journal*, Vol. 45, No. 6, 2007, pp. 1232-1243 (DOI: 10.2514/1.24282).
2. C. J. Roy and A. J. Sinclair, "On the Generation of Exact Solutions for Evaluating Numerical Schemes and Estimating Discretization Error," *Journal of Computational Physics*, Vol. 228, No. 5, 2009, pp. 1790-1802 (DOI: 10.1016/j.jcp.2008.11.008).
3. C. J. Roy, T. S. Phillips, A. Choudhary, and J. Derlaga, "A Residual-Based Framework for Error Estimation and Adaptivity for Finite Volume and Finite Difference Methods," manuscript in preparation for submission to the *Journal of Computational Physics*.

Conference Papers

1. C. J. Roy and A. J. Sinclair, "On the Generation of Exact Solutions for Evaluating Numerical Schemes and Estimating Discretization Error," NATO RTO Symposium on Computational Uncertainty, Athens, Greece, Dec. 3-6, 2007.
2. C. J. Roy, "Strategies for Driving Mesh Adaptation in CFD," AIAA 2009-1302, invited paper for session on Error Estimation and Control, 47th AIAA Aerospace Sciences Meeting, Orlando, Florida, January 5-8, 2009.
3. M. J. Kurzen, T. S. Phillips, C. J. Roy, and A. J. Sinclair, "Method of Nearby Problems for Generating Exact Solutions to 1D Unsteady and 2D Steady Problems," AIAA Paper 2009-3652, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, Texas, June 2009.
4. T. S. Phillips and C. J. Roy, "Residual Methods for Discretization Error Estimation," Accepted for presentation at the 20th AIAA Computational Fluid Dynamics Conference, June 27-30, 2011, Honolulu, Hawaii.
5. C. J. Roy, "Survey of Residual-based Methods for Estimating Discretization Error," American Nuclear Society: 2010 Winter Meeting and Nuclear Technology Expo, November 7-11, 2010, Las Vegas, Nevada.

Graduate Student Theses

1. Anil Raju, "Discretization Error Estimation Using the Method of Nearby Problems: One-Dimensional Cases," Masters Thesis, Auburn University, August 2005.
2. Matthew Kurzen, "Discretization Error Estimation and Exact Solution Generation Using the 2D Method of Nearby Problems," Masters Thesis, Virginia Tech, February 2010.

DISTRIBUTION

1 Virginia Tech
Attn: C. Roy (electronic copy)
215 Randolph Hall
Blacksburg, VA 24061

1	MS0825	Jeff Payne	01515 (electronic copy)
1	MS0825	Justin Smith	01515 (electronic copy)
1	MS0828	Kenneth Alvin	01544 (electronic copy)
1	MS0828	Amalia Black	01544 (electronic copy)
1	MS0828	Ryan Bond	01541 (electronic copy)
1	MS0828	Brian Carnes	01544 (electronic copy)
1	MS0836	Matt Hopkins	01516 (electronic copy)
1	MS0836	Sheldon Tieszen	01541 (electronic copy)
1	MS0897	Kevin Copps	01544 (electronic copy)
1	MS1124	Matt Barone	06121 (electronic copy)
1	MS1318	Patrick Knupp	01318 (electronic copy)
1	MS1318	James Stewart	01318 (electronic copy)
1	MS0899	Technical Library	9536 (electronic copy)
1	MS0359	D. Chavez, LDRD Office	1911


Sandia National Laboratories