AIAA JOURNAL
Vol. 45, No. 6, June 2007

Estimation of Discretization Errors Using the Method of Nearby Problems

Christopher J. Roy* and Anil Raju†
Auburn University, Auburn, Alabama 36849
and
Matthew M. Hopkins‡
Sandia National Laboratories, Albuquerque, New Mexico 87185-0834

DOI: 10.2514/1.24282

The method of nearby problems is developed as an approach for estimating numerical errors due to insufficient mesh resolution. A key aspect of this approach is the generation of accurate, analytic curve fits to an underlying numerical solution. Accurate fits are demonstrated using fifth-order Hermite splines that provide for solution continuity up to the third derivative, which is recommended for second-order differential equations. This approach relies on the generation of a new problem (and corresponding exact solution) that is "nearby" the original problem of interest, and the nearness requirements are discussed. The method of nearby problems is demonstrated as an accurate discretization error estimator for steady-state Burgers's equation for a viscous shock wave at Reynolds numbers of 8 and 64. A key advantage of using the method of nearby problems as an error estimator is that it requires only one additional solution on the same mesh, as opposed to multiple mesh solutions required for extrapolation-based error estimators. Furthermore, the present results suggest that the method of nearby problems can produce better error estimates than other methods in the preasymptotic regime. The method of nearby problems is also shown to provide a useful framework for evaluating other discretization error estimators. This framework is demonstrated by the generation of exact solutions to problems nearby Burgers's equation as well as a form of Burgers's equation with a nonlinear viscosity variation.

Nomenclature

a_i–f_i	=	spline coefficients
E	=	discrete L^2 norm error function
f	=	general function
h	=	global measure of cell or element size
L	=	differential operator
L_{ref}	=	reference length of domain
N	=	number of mesh points
n	=	number of spline points
p	=	order of accuracy
r	=	grid refinement factor ($r > 1$)
S	=	spline polynomial
s	=	source term
t	=	time coordinate, s
u	=	local solution variable, m/s
x	=	spatial coordinate, m
α	=	scaling constant
ν	=	viscosity, m^2/s
Ω	=	solution domain

Subscripts

exact	=	exact solution to differential equation
i	=	spline zone number
j	=	node number
k	=	mesh level (1 = finest mesh)

Superscripts

$'$	=	dimensionless variable, first derivative
$''$	=	second derivative
$'''$	=	third derivative
$\bar{}$	=	scaled variable
$\tilde{}$	=	numerical solution
$\hat{}$	=	estimated exact solution to differential equation

Received 29 March 2006; revision received 15 January 2007; accepted for publication 16 January 2007.

*Assistant Professor, Aerospace Engineering Department, 211 Aerospace Engineering Building. Associate Fellow AIAA.
†Research Assistant, Aerospace Engineering Department, 211 Aerospace Engineering Building. Student Member AIAA.
‡Senior Member of Technical Staff, Multiphase Transport Processes Department, P.O. Box 5800, Mail Stop 0834.

I. Introduction

SOURCES of error in computational simulation can be classified into two main categories: modeling errors and numerical errors. Modeling errors arise due to a given model's inability to reproduce the behavior observed in the real world. For example, a turbulence model may be calibrated for attached, zero-pressure gradient flow, but fail to predict the correct separation characteristics in a flow with a strong adverse pressure gradient. Numerical errors can be associated with a number of sources including mesh resolution, time step, discretization scheme, iterative convergence, and roundoff. For complex simulations (e.g., coupled, nonlinear partial differential equations with arbitrary geometry), it is particularly important to control and understand numerical errors. Failure to do so can not only lead to poor engineering decisions, but can also erode the general confidence in computational simulation.

Discretization error estimators can be broadly classified into two main categories: extrapolation-based and finite-element-based error estimators. The extrapolation-based error estimators are based on Richardson extrapolation [1,2], where the numerical solutions on two or more meshes are extrapolated to zero element size to approximate the exact solution. The finite-element error estimators can be further divided into residual-based error estimators [3], which usually involve the solution to the adjoint or dual problem, and recovery-based error estimators (e.g., the popular Zienkiewicz–Zhu error estimator [4,5]) which compare local solution values or gradients to those found using patches of neighboring elements.

One aspect that all of the aforementioned error estimators have in common is that they require the underlying numerical solution (or solutions) to be in the asymptotic convergence range. Asymptotic convergence occurs when the spatial mesh and/or time step are sufficiently small that further refinement will result in a reduction in the discretization error at the theoretical rate of the numerical scheme, that is, the formal order of accuracy. For many engineering applications, this formal order of accuracy is second order, meaning that refinement in each coordinate direction by a factor of 2 should result in a reduction of the discretization error by a factor of 4. (Some applications such as acoustics and turbulence simulations may be more efficient when higher-order methods are used.) The formal order is usually found by truncation error analysis for finite-difference and finite-volume schemes, and by interpolation theory for finite-element methods, and is determined by the leading error term, which will dominate the other terms in the limit as the cell size and/or time step approach zero. A coarse grid error estimator that can operate in the preasymptotic regime (where the error is not entirely described by the leading error term) remains an elusive goal for complex problems with coupled equations, nonlinearity, and mixed hyperbolic/elliptic character. The development of a coarse grid error estimator would reduce the cost of performing numerical simulations, provide better error/uncertainty estimates for computational predictions, and allow more complex simulations to be performed with a higher degree of confidence in the solutions.

The standard methods for evaluating the efficacy of error estimators involve the use of either exact solutions or benchmark numerical solutions. For complex partial differential equations (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 and is therefore not exercised. The use of a benchmark numerical solution or highly refined "truth" mesh is also problematic because 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.

There has been some prior work in the literature dealing with the generation of exact solutions. One example is the method of manufactured solutions [2,6–8], where an analytic solution is chosen a priori and the governing equations are modified by the addition of analytic source terms. These source terms come from operating the original governing equations onto the chosen solution. This chosen solution is now the exact solution to the modified set of governing equations, which consists of the original governing equations combined with the generated analytic source terms. The purpose of manufactured solutions is for code verification, that is, to ensure to the highest degree possible that a given simulation code is free from coding mistakes. The manufactured solutions are generally chosen a priori for their smoothness and for their ability to exercise all terms in the governing equations. However, code verification is a mathematical exercise that does not attempt to assess the adequacy of the physical models, thus the solutions are generally nonphysical by design. Our current work is related to manufactured solutions, with the important difference that the solutions are required to be physically realistic and thus involve an important curve-fitting step not found in the manufactured solutions approach. A related exercise, that of determining if a particular numerical solution is accurate (e.g., the mesh is sufficiently refined), is termed solution verification. It is this latter exercise that we address.

Another approach to generating exact solutions was developed by Lee and Junkins [9] for 1-D nonlinear ordinary differential equations (ODEs). The basic idea behind their work is summarized in the following steps:

1) Compute a numerical solution on a highly refined mesh.

2) Generate an analytic solution from a global fit to the fine grid numerical solution based on the least-squares approach using Chebyshev polynomials.

3) Use symbolic manipulation (in their case MACSYMA) to plug the analytic polynomial solution into the original ODEs to generate small source terms.

4) Solve the nearby problem, consisting of the original ODEs plus the small source terms, on a series of different discretizations.

The goal of their work was to determine the optimal numerical integration parameters for a given problem. Junkins and Lee [10] later extended their methodology to nonlinear hybrid ODE/PDEs that arise from flexible multibody dynamical systems in two dimensions.

Recently, our group has developed a method for both estimating the discretization error and for generating exact solutions, called the method of nearby problems (MNP) [11–15]. Although our earlier work referred to MNP as a new approach to estimating discretization error, an astute reviewer of this manuscript pointed out that this approach has in fact been around for three decades and is commonly referred to as the differential form of defect correction (or the differential correction). This form of defect correction was first introduced by Zadunaisky [16] in 1976 to estimate discretization error for ordinary differential equations. An excellent review of defect correction methods is presented by Skeel [17], which includes details of the standard and iterated versions of differential correction as well as its discrete analog difference correction. Skeel further suggests that when the estimated errors are of sufficient accuracy to be used to improve the solution that they be referred to as error estimates, but recommends that the term uncertainty estimate be used for "any error estimate that is too crude for improving the solution."

In our initial work on MNP [11], we examined the generation of exact solutions to problems near an original problem of interest. We examined two cases: fully developed laminar flow in a channel and a lid-driven cavity. Nearby problems with exact solutions for the channel flow case were successfully generated, whereas nearby problems with exact solutions for the driven cavity case were elusive, primarily due to the strong corner singularities at the driven wall. A later paper [12] presented rigorous mathematical theory for the method as applied to first-order quasi-linear ODEs. A more recent effort [13] focused on the generation of nearby problems with exact solutions for steady-state Burgers's equation using global polynomial fits. In this work, both Legendre polynomials and standard polynomials (monomials) were shown to be inadequate due to large errors generated near strong gradients and at boundaries.

In this article, we seek to overcome the problems with global polynomials by using fifth-order Hermite splines [14,15]. We first demonstrate the effectiveness of local spline fits relative to global polynomial fits. We discuss the "nearness" requirements for the nearby problem, namely, that the generated source terms be smooth and small in magnitude. MNP is then demonstrated to provide accurate discretization error estimates even where other methods are unable to produce estimates, or produce increasingly poor estimates due to increasing nonasymptotic behavior as the mesh is coarsened. In addition, MNP is also used as a framework for evaluating various extrapolation-based discretization error estimators.

II. Method of Nearby Problems

A. MNP as an Error Estimator

The steps involved in using MNP as an error estimator can be summarized as follows:

1) Compute a numerical solution on a chosen mesh.
2) Generate an analytic curve fit to the numerical solution.
3) Generate analytic source terms.
4) Numerically solve the nearby problem.
5) Evaluate the discretization error in the nearby problem.
6) Apply nearby problem error estimates to the original problem.

These six steps are described in detail in the following paragraphs.

1) Compute original numerical solution: Once the problem of interest is identified, the first step is to compute the numerical solution on a given mesh. This solution will have some associated

discretization error, which we would like to estimate, even where the error is no longer well represented by an asymptotic description (i.e., the error is not described well by a leading error term).

2) Analytic curve fit: This step is generally the most difficult and involves generating an accurate analytic fit of the numerical solution computed in step 1. When spline fits are employed, a certain amount of continuity is required across spline zones. Because we generally require that the source terms generated by the method be smooth, when examining a second-order differential equation such as Burgers's equation, C^3 continuity is required. That is, we must have continuity of the solution as well as its first, second, and third derivatives across the spline zones. Once the curve fit has been generated, some measure of the goodness of the fit must be quantified to determine how well it satisfies the given data (i.e., the original numerical solution) [12].

3) Generation of analytic source terms: The analytic curve fit from step 2 now becomes the exact solution to a set of equations near the original equations. In fact, these neighboring equations differ from the original equations only by a (it is hoped) small source term. These source terms come from operating the original equations (along with any auxiliary equations) onto the curve fit solution from step 2. As the size of these source terms approaches zero, solutions to the perturbed equations approach the solution of the original equations, with possible smoothness and uniqueness constraints.

How small do the source terms need to be? Relating the solution difference between the original and nearby problems to the difference between the equations (i.e., the size of the generated source term) is difficult. Theory can tell us how to measure these distances as a function of source term size for simple cases such as linear differential equations (based on the condition number of the Jacobian of the underlying algebraic system), but we are not aware of any work addressing this issue for general coupled nonlinear partial differential equations. The resulting perturbed equations would still be valuable as a verification problem, but would possibly not be as close to the starting equations as one would like (i.e., the "physics regime" is too different). We nonetheless expect that the resulting methodology presented here to be of great practical value.

The closeness of these neighboring equations to the original equations is determined by examining the size of the associated source terms. Recall the definition of the L^2 norm of a function f on a domain Ω,

$$\|f\|^2_{L^2(\Omega)} = \int_\Omega f^2 \, \mathrm{d}\omega \tag{1}$$

In the present case, the L^2 norm of the source term for each governing equation is calculated by integrating over the domain of interest Ω. In theory this integration could be performed analytically. Unfortunately, as the governing equations or curve fit functions become more complex, analytic integration (even with a symbolic math package) becomes less efficient. We instead numerically evaluate the integral in Eq. (1) using the trapezoidal rule with quadrature points given by the mesh. It should be noted that the number of quadrature points is much larger than the number of spline zones used. The size of the source term is influenced by a number of factors. One is the difference between a piecewise defined function (the numerical solution) and what the spline basis can represent. Another is the accuracy of the underlying numerical solution: the more accurate a numerical solution, the closer the spline approximation is to the solution to the original equations (and thus the smaller the source term).

4) Numerical solution to nearby problem: The nearby problem is then discretized and computed using the same numerical method on the same mesh as the original solution. This solution process is quite similar to that of the original problem, but includes the additional analytic source terms and any perturbed initial and boundary conditions.

5) Evaluation of the discretization error: Because the exact solution to the nearby problem is known (it is simply the analytic curve fit from step 2), the discretization error for the nearby problem can be evaluated exactly and does not need to be estimated. To evaluate the discretization error in a global sense, we define a discrete error function

$$E(u_k) = \left(\frac{1}{N} \sum_{j=1}^{N} (u_{k,j} - u_{\text{exact},j})^2 \right)^{1/2} \tag{2}$$

where k refers to the discrete mesh level and N is the number of mesh nodes in space (including both interior and boundary nodes) with the exception of the Dirichlet boundary nodes for which the discretization error is identically zero. Here, $u_{\text{exact},j}$ refers to the exact solution evaluated at node j.

6) Apply error estimate to original problem: If the nearby problem is sufficiently close to the original problem, then the discretization error in the original problem can be approximated by the error in the nearby problem, either locally or globally.

We note that the approximation functions used in step 2 should satisfy certain criteria. The approximation functions should not be in the kernel of the differential operator [e.g., $\cos(nx)$ functions for $u''(x) + u(x) = 0$]. Ideally, they are also of sufficiently high order so that the source terms are nontrivial (e.g., they do not resolve to a constant).

These six steps involved in using MNP as an error estimator can be best explained by a simple example. Consider that

$$L(u) = \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} \tag{3}$$

is the differential operator of interest, and we wish to solve $L(u) = 0$. The first step in MNP involves obtaining a numerical solution to this original problem. The second step involves fitting an analytic curve fit to this numerical solution. We require that the analytic fits be C^3 continuous to ensure differentiability of the source terms because the highest order of the differential operator is two. We expect differentiable source terms to offer advantages over source terms that are simply continuous. For this example, we use a global analytic curve fit of the following simple form:

$$\tilde{u}(x) = a + bx + cx^2 + dx^3 + ex^4 \tag{4}$$

The third step of MNP is to operate the original equation on the analytic curve fit and come up with an analytic source term. By operating the original problem of interest on the analytic curve fit, we get

$$s(x) = L(\tilde{u}) = b + 2c(x+1) + 3dx(x+2) + 4ex^2(x+3)$$

The source term $s(x)$ is only equal to zero if $\tilde{u}$ exactly solves the original equation. An inadmissible approximation function would be $a + be^{-x}$, which is in the kernel of the operator. Although approximation functions of the form $a + bx$ would generate nonzero source terms, we do not consider symbolic form of $s(x) = b$ to be rich enough. Now the analytic curve fit $\tilde{u}(x)$ becomes the exact solution to the modified equation (i.e., the nearby problem)

$$L(u) = s(x)$$

It should be noted that as $s(x)$ approaches zero, the nearby problem approaches the original problem, with sufficient regularity and uniqueness conditions, which Eq. (3) has. The next step is to come up with a numerical solution to the nearby problem. Because we have an exact solution $\tilde{u}$ to the nearby problem, we can then evaluate the discretization error exactly. If the source term is in fact small and the nearby problem is sufficiently close to the original problem, then the *exact discretization error evaluated on the nearby problem* can be applied as an *error estimate on the original problem*.

B. MNP as a Framework for Error Estimation

The method of nearby problems can also be used as an approach for evaluating the effectiveness of other error estimators. The steps in using MNP as an error estimation framework are similar to those for applying it as an error estimator, and can be summarized as follows:

1) Establish an accurate numerical solution.

2) Generate an analytic curve fit to the numerical solution.
3) Generate analytic source terms.
4) Numerically solve the nearby problem.
5) Apply different error estimators to the nearby problem and compare their accuracies.

These five steps are described in more detail in the following paragraphs:

1) Accurate numerical solution: Once the problem of interest is identified, the first step is to compute a numerical solution. Although this solution will have some associated discretization error, this fact will not pose a problem as will be shown later.

2) Analytic curve fit: An analytic curve fit to the numerical solution from step 1 is then constructed. See step 2 from Sec. II.A for more details.

3) Generation of analytic source terms: Analytic source terms are generated by operating the original governing equation on the analytic curve fit from step 2. See step 3 from Sec. II.A for more details.

4) Numerical solution to nearby problem: The neighboring problem is then discretized and computed on a given mesh, including the source term from the last step and any perturbed boundary conditions. For consistent numerical schemes and sufficiently refined meshes, the formal order of accuracy of the numerical scheme should be observed, even on the perturbed equations. In general, as the grid is refined the discretization error should drop as $1/r^p$, where r is the grid refinement factor and p the formal order of accuracy. To examine the global discretization error behavior, we employ the discrete error function given in Eq. (2).

5) Apply different error estimators to the nearby problem and compare their accuracies: Because the exact solution to the nearby problem is known (i.e., it is the analytic curve fit), the discretization error in the numerical solution to the nearby problem no longer has to be estimated, but can now be evaluated exactly. Thus, we can apply different error estimators to the nearby problem and know how well they truly approximated the error.

III. Burgers's Equation

A. Exact Solution

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

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{5}$$

where $u(x, t)$ is a scalar field. We have selected Burgers's equation because it is a quasi-linear scalar equation with a number of known exact solutions [18]. Of these, we have chosen the steady-state viscous shock wave solution for our initial testing. This solution is chosen because it is smooth, nontrivial, 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

$$u'(x) = \frac{-2\sinh(x')}{\cosh(x')} \tag{6}$$

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

$$Re = \frac{u_{\text{ref}} L_{\text{ref}}}{\nu} \tag{7}$$

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

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

$$x' = x/L_{\text{ref}} \quad \text{and} \quad u' = uL_{\text{ref}}/\nu \tag{8}$$

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

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

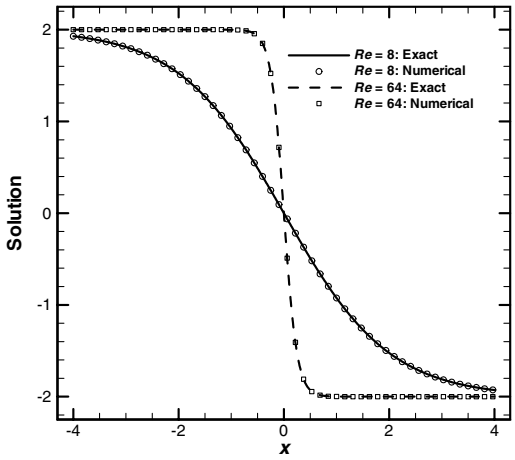


Fig. 1 Exact and numerical solutions to Burgers's equation for a steady, viscous shock wave at Reynolds numbers of 8 and 64.

This exact solution is represented by the curves in Fig. 1, in which x is on the ordinate and u the abscissa. Two different values for ν are shown, corresponding to Reynolds numbers of 8 and 64.

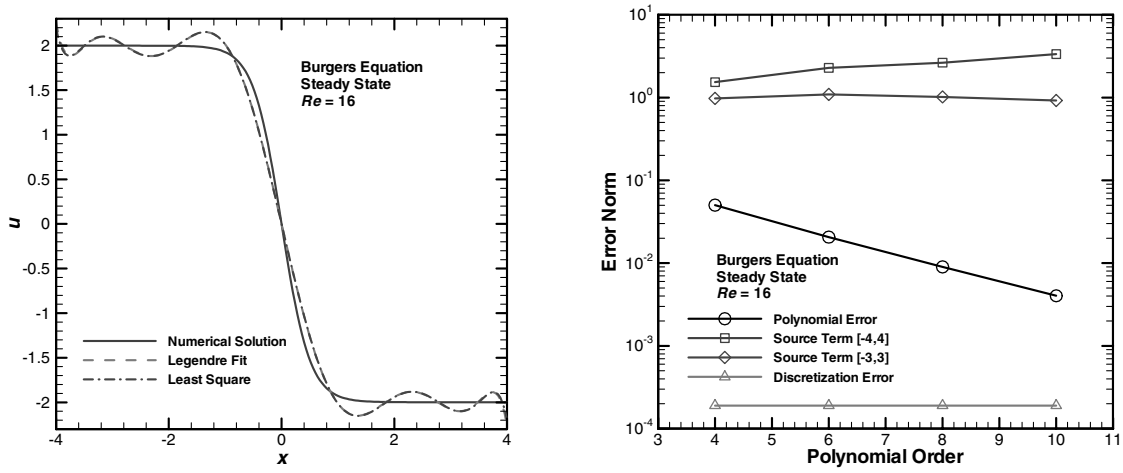
Two additional unsteady exact solutions to Burgers's equation are appropriate for use with 2-D curve-fitting procedures are presented in [14,15].

B. Discretization Scheme

A fully implicit finite-difference code was developed to solve the steady-state form of Burgers's equation. The nonlinear term is linearized and the resulting linear tridiagonal system is solved directly using the Thomas algorithm. This fully implicit method is formally second-order accurate in space because the leading truncation error term is on the order of Δx^2 for both the convection and diffusion terms. The temporal term is retained and discretized using a backward difference in time. The resulting equations are marched in pseudotime until the nonlinear system is converged to machine zero (an approximately 12 order of magnitude reduction in the residual). The numerical solutions on a highly refined mesh for steady-state Burgers's equation are also shown in Fig. 1 for Reynolds numbers 8 and 64. These numerical solutions are visibly indistinguishable from the exact solutions. All numerical solutions presented herein are on equally spaced grids, and all grid refinement is determined by injecting nodes between existing nodes (i.e., grid doubling).

IV. Curve-Fitting Procedure

In our previous work [13] 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 [19] in Fig. 2a for steady-state Burgers's equation at a Reynolds number of 16 (see [13]). This lack of agreement is also evident in Fig. 2b, which gives L^2 norms of the source term and the curve-fitting error as a function of the polynomial order. Although 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 Fig. 2b). Lee and Junkins [9] also observed an increase in polynomial error near the boundaries using global Chebyshev polynomials. In the current work, we rely exclusively on spline fits, where a piecewise-polynomial approximation is made by dividing the domain into a sequence of equally spaced adjacent zones (overlapping only at their boundary points). Different polynomials



a) b)

Fig. 2 Global polynomial fit for Burgers's 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 magnitude vs polynomial order.

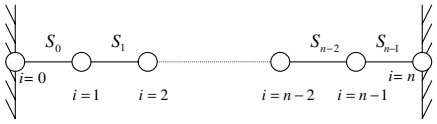


Fig. 3 Schematic of the spline-fitting system.

are constructed on each of these zones, with continuity constraints enforced at the zone boundaries.

A. Cubic Spline Fits

A cubic spline is constructed of piecewise third-order polynomials [20]. A cubic spline is twice continuously differentiable (i.e., is C^2 continuous) and depends on four parameters. It can be written as

$$S_i(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \qquad (9)$$

for

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

and a schematic of this spline system is shown in Fig. 3. The domain is broken up into $n + 1$ nodes (0 to n) and therefore n spline zones $(S_0, S_1, \ldots, S_{n-1})$. Equation (9) 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

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

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.

B. Hermite Spline Fits

A fifth-degree Hermite spline is constructed of piecewise fifth-order polynomials [20]. This spline polynomial is given by

$$S_i(x) := a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 + e_i(x - x_i)^4 + f_i(x - x_i)^5 \qquad (10)$$

for

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

where the same spline system given in Fig. 3 is used. The conditions used to construct the fifth-degree Hermite spline are

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

where the first two conditions match values and derivatives of the given solution. In our case, the derivatives come from the same method for discretizing the first derivative, namely, second-order accurate central differences. This choice is not unique, and any reasonable approximation for the derivative can be used. 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's equation. The results for steady-state Burgers's equation with a Reynolds number of 64 are presented in Fig. 4. The numerical solution appears to be approximated quite well because the fifth-order Hermite spline fit with 65 spline points is

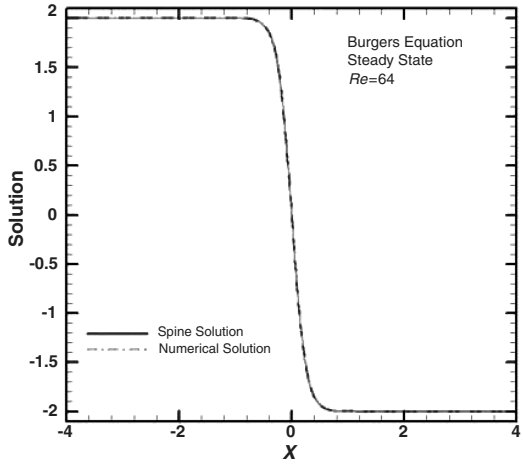


Fig. 4 Fifth-order Hermite spline fit and underlying numerical solution for Burgers's equation at a Reynolds number of 64 using 65 spline points.

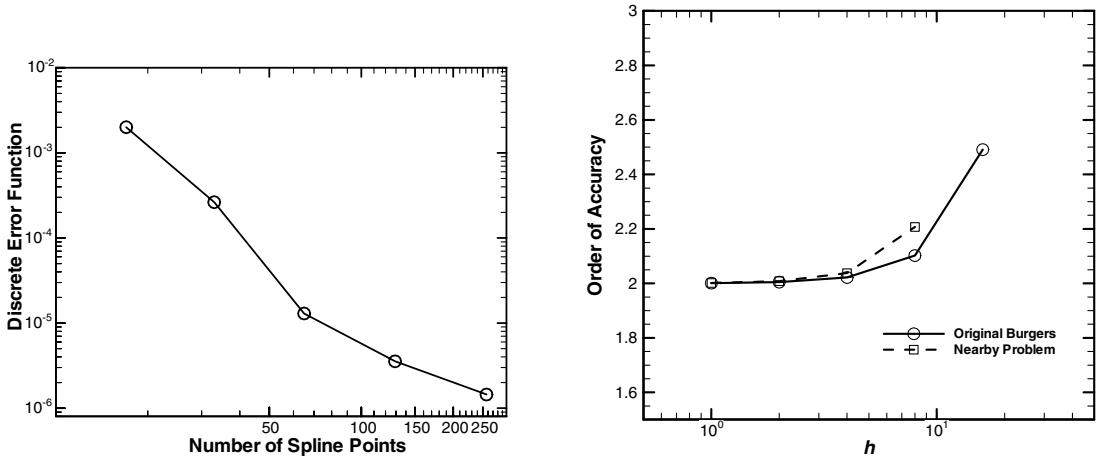


Fig. 5 Burgers's equation at a Reynolds number of 64: a) discrete error function for the difference between the spline fit and the underlying numerical solution, and b) order of accuracy variation with mesh refinement (nearby problem is also shown).

visually indistinguishable from the underlying numerical solution. Note that even though the Reynolds number for this case is larger by a factor of 4 than the global polynomial fit shown in Fig. 2 (thus resulting in a sharper gradient), the results are significantly better with the Hermite spline fit. The number of spline zones required is related to the nearness of the nearby solution to the underlying numerical solution, and is examined in detail in the next section.

V. Nearness of the Nearby Problem

For steady-state Burgers's equation, the viscosity values ν were chosen as $2\ m^2/s$ and $1/4\ m^2/s$ for the current analysis, corresponding to Reynolds numbers of 8 and 64, respectively. As the Reynolds number is increased, the viscous shock becomes sharper (see Fig. 1). Because the shock needs to be resolved in all cases, additional grid points are needed for the higher Reynolds number solution. The number of grid points employed in computing the underlying numerical solution for the Reynolds number 8 and 64 cases were 257 and 513, respectively. The norms of the error between the Hermite splines fits and the underlying numerical solution for a Reynolds number of 64 are presented in Fig. 5a as a function of the number of spline points used. These errors norms are shown to decrease as the number of spline points increases. The errors for this Reynolds number 64 case are also much lower than those seen for the global polynomials at a Reynolds number of 16 (see Fig. 2b), even with only 17 spline points used. Because in the special case of Burgers's equation an exact solution exists, the numerical solution to the original equation can be examined for convergence at the formal order of accuracy of 2. The order of accuracy of the discrete error function [Eq. (2)] is shown in Fig. 5b for both Burgers's original equation and the nearby problem as a function of the mesh spacing h (note: smaller values of h indicate finer meshes). As expected, the order of accuracy approaches 2 as the mesh is refined. This provides strong evidence that the codes for solving both Burgers's equation and the nearby problem are working correctly and free of coding mistakes.

In our previous work [13] we used global Legendre polynomial fits to the underlying numerical solution. An example for a Reynolds number of 16 was shown in Fig. 2a. The solutions using global polynomials exhibit large oscillations, especially at the boundaries. 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 Fig. 6. Furthermore, the source terms become larger at the boundaries as the polynomial order is increased.

When the fifth-order Hermite splines are implemented, the magnitude of the source term is significantly smaller. The source term for the Reynolds number 8 case using just five spline points is shown in Fig. 7 along the entire domain. The source term is much smaller than that seen in the global polynomial fits (Fig. 6), with maximum magnitudes over the entire domain approximately 0.02. The use of additional spline points further reduces the size of the source term as shown in Fig. 8, in which 17 spline points gives a maximum magnitude of roughly 0.0017. The choice for the number of spline points depends on the application, with additional spline points generally resulting in source terms of lower magnitudes, but with higher frequency content. 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 Fig. 9. The maximum magnitude of the source term is on the order of

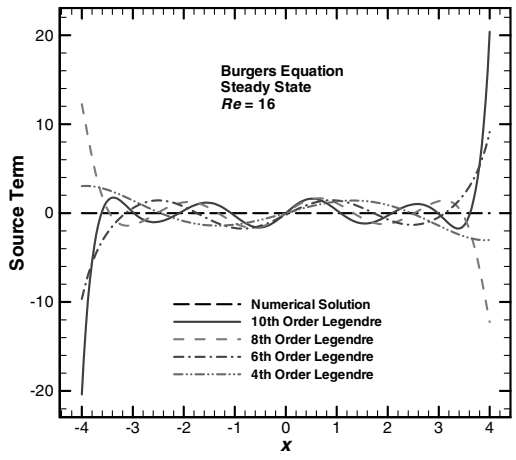


Fig. 6 Source term distribution using different order Legendre polynomial fits for Burgers's equation at a Reynolds number of 16.

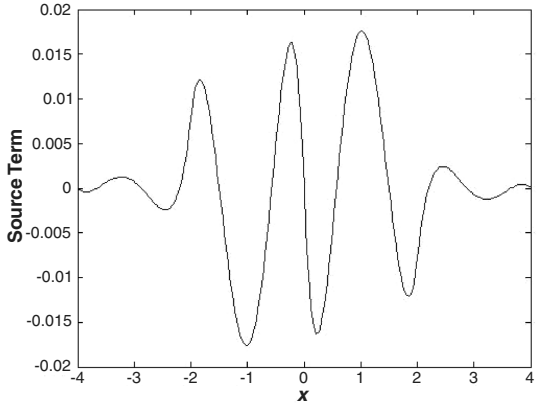


Fig. 7 Source term for the nearby problem with five Hermite spline points for a Reynolds number of 8.

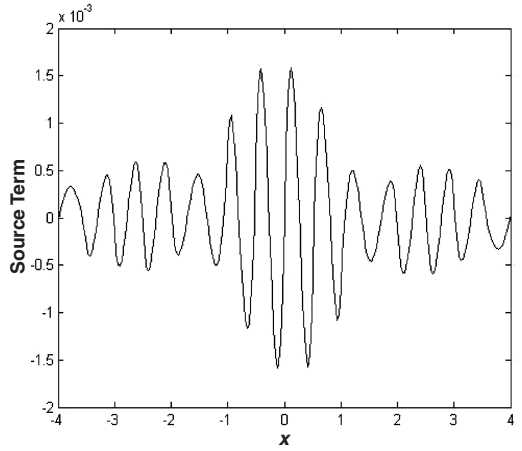


Fig. 8 Source term for the nearby problem with 17 Hermite spline points for a Reynolds number of 8.

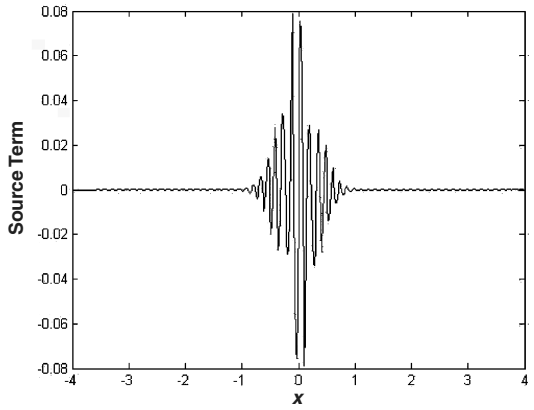


Fig. 9 Source term for the nearby problem with 65 Hermite spline points for a Reynolds number of 64.

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.

The Reynolds number 8 case was also approximated using standard cubic splines. Figure 10 shows the distribution of the source term along the domain using 17 spline points. Although 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. Because we desire smooth source terms that are C^1 continuous, we will employ only Hermite splines for the remainder of this article.

To summarize, we have used three methods to judge the nearness of the nearby problem. The first method is to simply visually compare the analytic nearby solution to the numerical solution data on which it is based. This comparison can be made for fifth-order Hermite splines at a Reynolds number of 64 (see Fig. 4) and for the global 10th-order Legendre polynomial at a Reynolds number of 16 (see Fig. 2a), which shows that the global polynomial approach is clearly inadequate. The second method is to compute the discrete error function of the difference between the analytic nearby solution and the numerical solution data on which it is based. This comparison is made for Hermite splines in Fig. 5a (Reynolds number 64) and for the global Legendre polynomials in Fig. 2b at Reynolds number 16 (circles). Even with 10th-order global polynomials, the discrete error function is only reduced to 4×10^{-3}, whereas using 65 spline points for a much higher Reynolds number (64 vs 16) yields a much smaller discrete error function of 1×10^{-5}. Note that this criteria for nearness could be modified to incorporate more appropriate normed spaces (e.g., Sobolev spaces). Finally, we compare the amplitude of the source terms from the Legendre polynomial in Fig. 6 (Reynolds number 16)

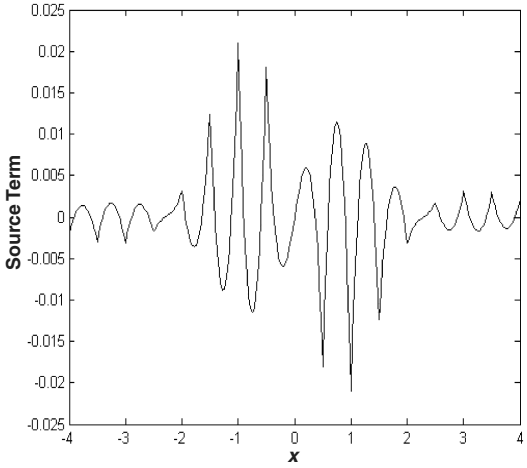


Fig. 10 Source term for the nearby problem with 17 cubic spline points for Burgers's equation at a Reynolds number of 8.

and from the Hermite spline with 65 spline zones in Fig. 9 (Reynolds number 64). Again, the Hermite splines are found to be superior even at a much higher Reynolds number, especially at the domain boundaries. Our choice of C^3 continuous Hermite splines ensures smooth (C^1 continuous) source terms for second-order differential equations.

Although the requirements on smoothness as well as oscillation frequency for the source terms are still under investigation, we can offer some preliminary guidelines regarding the number of spline nodes to employ. Clearly the grid for the underlying numerical solution provides an upper limit on the number of spline nodes to employ. Furthermore, we recommend adding spline nodes until the magnitude of the source terms are deemed sufficiently small. (Note that this might take the form of physical constraints on the size of the distributed source term, such as a gravity-induced buoyancy term for the momentum equations.) As soon as the source terms are judged to be sufficiently small, no additional spline nodes should be used due to the linear increase in the oscillation frequency of the source terms with the number of spline nodes. Again, our future plans for evaluating the nearness of the nearby problems include the use of Sobolev norms, which measure not only the magnitude of the source terms, but also the magnitudes of derivatives of the source terms.

VI. Discretization Error Estimators

There are three sources of numerical error that can arise during the numerical solution to differential equations [8]. Roundoff error occurs due to the finite number of significant digits used to store floating point numbers on digital computers. Iterative convergence error arises when the discretized equations are not solved exactly, but rather in an iterative manner. Iterative methods are generally required for nonlinear systems, and are often the most efficient methods for large linear systems. Discretization error is defined as the difference between the exact solution to the discretized equations (assuming zero roundoff and iterative error) and the exact solution to the original (continuous) partial differential equations. For all cases presented herein, double-precision computations were used with approximately 15 significant digits, and the iterative error was reduced down to machine zero as judged by the discrete, steady-state residual reduction. Thus the difference between the current numerical solutions and the exact solution to the continuous differential equations is expected to accurately represent the discretization error. The relative discretization error (RDE) on the fine grid can be written as

$$\mathrm{RDE}_1 = \frac{f_1 - f_{\mathrm{exact}}}{f_{\mathrm{exact}}} \tag{11}$$

where f_{exact} is the exact solution to the differential equation.

A. Richardson Extrapolation with Global Order

We can estimate the exact solution to the partial differential equation f_{exact} using Richardson extrapolation [1,2], which is given by

$$\hat{f}_{\text{exact}} = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{12}$$

where f_2 and f_1 are the solutions on a coarse and fine mesh, respectively. The formal order of accuracy can be found with a truncation error analysis of the discretization scheme and is equal to two for the current case. This approach requires two mesh levels: fine and coarse meshes. In general, this approach requires both solutions to be within the asymptotic mesh convergence range where only the leading truncation error term dominates.

B. Richardson Extrapolation with Local Order

The Richardson extrapolation expression given in Eq. (12) requires the order of accuracy p as an input. Instead of assuming the formal order of accuracy, the observed order of accuracy can be employed at each grid node (i.e., locally) using solutions on three meshes [2] as

$$p = \frac{\ell_n[(f_3 - f_2)/(f_2 - f_1)]}{\ell_n(r)} \tag{13}$$

This approach requires three different mesh levels: coarse, medium, and fine meshes, which have been successively refined by the same factor r.

C. Mixed-Order Error Estimator

Roy has developed an error estimator to work in cases where only first- and second-order error terms are significant contributors to the overall error [21]. This error estimator also requires three mesh levels. For a constant mesh refinement factor between the three meshes, the estimate of the exact solution takes the following form:

$$\hat{f}_{\text{exact}} = \frac{(f_3 - f_2) - (r^2 + r - 1)(f_2 - f_1)}{(r+1)(r-1)^2} \tag{14}$$

D. Method of Nearby Problems

As stated earlier, MNP itself can be used as an error estimator. Because the MNP approach involves the generation of an exact solution to the nearby problem, the discretization error for the nearby problem can be evaluated exactly. If the nearby problem is "close enough" to the original problem of interest, then the error on a given mesh for the nearby problem is expected to be very close to the error in the original problem on the same mesh. The relative error in the fine grid for the nearby problem can be evaluated exactly as

$$\text{MNP}_k = \frac{f_{k,\text{MNP}} - f_{\text{exact,MNP}}}{f_{\text{exact,MNP}}} \tag{15}$$

The discretization error for the original problem on mesh level k is then assumed to be equal to the error in the nearby problem on mesh k, that is.,

$$\text{RDE}_k \cong \text{MNP}_k \tag{16}$$

The cost of using MNP as an error estimator is approximately equal to the cost of computing the numerical solution to the original problem because the same mesh is used. (There is, of course, some additional overhead associated with the spline-fitting procedure.) An advantage to using MNP as an error estimator is that it may provide reasonable error estimates even when used in the preasymptotic regime. Our results presented in the next section suggest that this may indeed be the case.

VII. Results

Having established the usefulness of Hermite splines for generating exact solutions to a nearby problem, we are now in a position to evaluate the various error estimators. We now examine the RDE estimates on various grid levels using four different methods: 1) RDE with global p: Richardson extrapolation assuming the formal order of accuracy (requiring two grids); 2) RDE with local p: Richardson extrapolation employing the locally (at each grid point) observed order of accuracy (requiring three grids); 3) mixed-order: a mixed-order error estimator (requiring three grids); and 4) MNP: the method of nearby problems requiring only a single grid (i.e., the same grid as used in the numerical solution). Numerical solutions are computed on a wide range of grid levels. In cases where multiple mesh levels are required to obtain the error estimate, the error in the fine grid solution is presented. Grid refinement is performed by halving the node spacing (i.e., grid doubling) in all cases. For the MNP method, a nearby problem is generated for each grid level.

A. Burgers's Equation

The discretization error for Burgers's equation is computed for the Reynolds number of 8 case using all four discretization error estimators and compared to the true error. The distribution of the various discretization error estimates are shown in Fig. 11 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 underpredicting the true error by a factor of 2 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 underpredicting the error by 10–15%. Both MNP and Richardson extrapolation using the formal order of accuracy provide good error estimates for this case. For this case, Richardson extrapolation with the local order underpredicts the size of the error because the observed order of accuracy asymptotically approaches the formal order of 2 from above.

For the Reynolds number 64 case, the discretization error estimates are shown in Fig. 12 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 underpredicts the error, Richardson extrapolation with the local order of accuracy slightly underpredicts 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, because 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 Fig. 12c, Richardson extrapolation gives error estimates that are nearly twice as large as the true error, whereas 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.

B. Nearby Problem to Burgers's Equation

The three extrapolation-based discretization error estimators were also applied to the nearby problem found from using Hermite spline fits to highly refined numerical solutions to Burgers's equations. For the Reynolds number 8 case, 17 spline fit points were used to fit the numerical solution to Burgers's equation on a 257-node mesh, thus generating the nearby problem. Using this nearby problem as a framework for evaluating error estimators, the discretization error results are presented in Fig. 13 for 257- and 65-node meshes. For the finer grid (Fig. 13a) all the approaches give good error estimates relative to the true error. For the coarser grid (Fig. 13b), only Richardson extrapolation with the formal order of accuracy matches the true error, with the other approaches predicting lower values. It is interesting to compare these error estimator results with those for

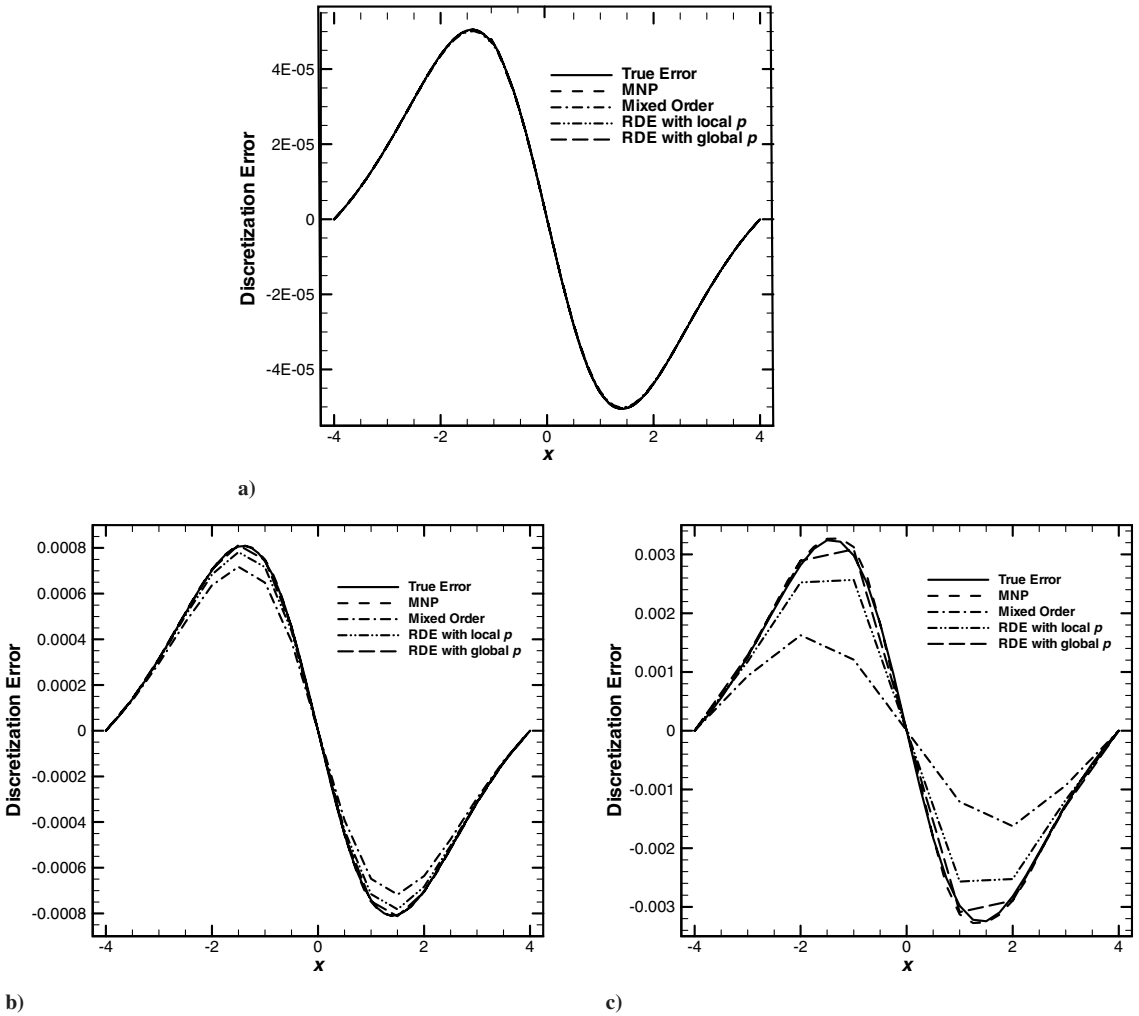


Fig. 11 Discretization error estimates for Burgers's equation with Reynolds number 8 using a finest mesh of a) 257 nodes, b) 65 nodes, and c) 33 nodes.

Burgers's original equation in Figs. 11a and 11b. The error estimates on the nearby problem appear to be identical to those for Burgers's original equation.

The discretization error was also computed for the nearby problem at a Reynolds number of 64, with the results presented in Fig. 14. For this Reynolds number 64 case, 33 spline fit points were used to fit the numerical solution to Burgers's equation on a 513-node mesh. All of the extrapolation-based error estimators provide good predictions on the fine mesh (Fig. 14a), whereas only Richardson extrapolation with the formal order of accuracy matches the true error for the coarse grid case (Fig. 14b). Again, these estimates on the nearby problem are nearly identical to those on Burgers's original equation given in Fig. 12a and 12b. These findings further support the idea of using MNP itself as an error estimator because the error for a given mesh on the nearby problem matches the error on the same mesh for the original problem.

C. Modified Burgers's Equation

A modified form of Burgers's equation was generated, which includes a nonlinear viscosity that varies as a function of both u and x:

$$\frac{\nu}{\nu_0} = \left(\frac{u}{u_0}\right)^2 + \left(\frac{x - x_L}{x_R - x_L} + \frac{1}{4}\right)^{1/4} \tag{17}$$

The constants were chosen as $\nu_0 = 0.25\ \text{m}^2/\text{s}$, $u_0 = 2\ \text{m/s}$, $x_L = -4$ m, and $x_R = 4$ m, thus giving a nominal Reynolds number of 64. This solution is shown in Fig. 15a, and no exact solution is known for this nonlinear viscosity variation. This modified form of Burgers's equation was solved numerically using a mesh with 1025 spatial points. This numerical solution was then used to generate spline fits with varying number of splines. The source term found using 65 spline points is presented in Fig. 15b and has a maximum magnitude of 0.015.

The nearby problem to the modified form of Burgers's equation was solved and the three extrapolation-based error estimators were used to estimate the discretization error. The nominal Reynolds number 64 case was run for meshes of 257 and 1025 nodes. The distribution of the discretization error over the spatial domain is presented in Fig. 16 for a fine grid of 257 nodes. The discretization error in the domain was very low except for a small region where the shock was located. In the vicinity of the viscous shock, the mixed-order error estimator underpredicts the true error by a factor of 2, whereas Richardson extrapolation with the local order of accuracy underpredicts the error by nearly 20%. Only Richardson extrapolation with the formal order of accuracy gives accurate error estimates.

VIII. Conclusions

The method of nearby problems has been extended using local, fifth-order Hermite spline fits. When applied to Burgers's equation, these local splines are shown to provide smaller source terms (than cubic splines or global polynomial approximations). The accuracy of the error estimates suggest that the nearby problems are indeed "near" the original equation. In addition, the fifth-order Hermite splines are shown to provide smooth source terms (continuous slopes and derivatives) due to the enforcement of C^3 continuity on the curve

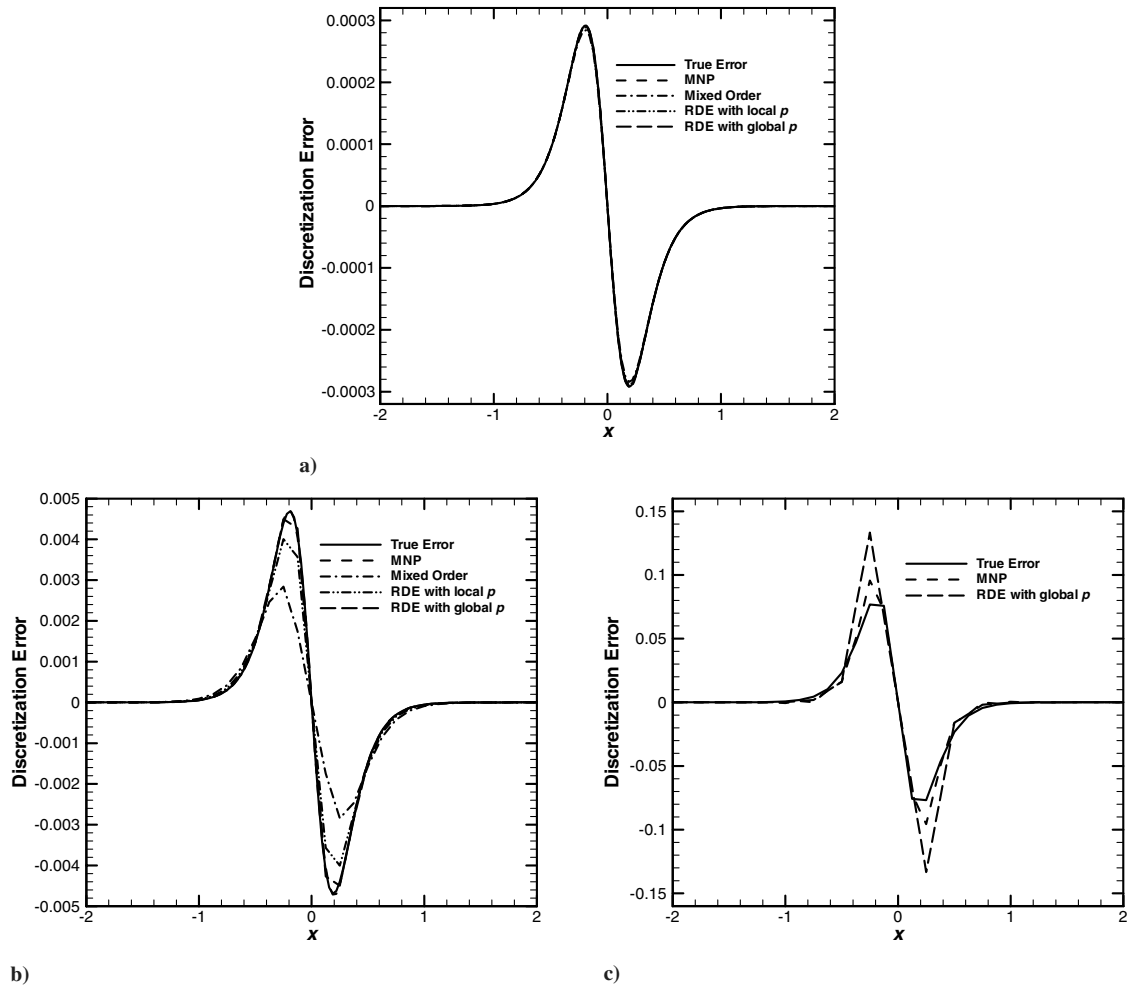


Fig. 12 Discretization error estimates for Burgers's equation with Reynolds number 64 using a finest mesh of a) 1025, b) 257, and c) 65 nodes.

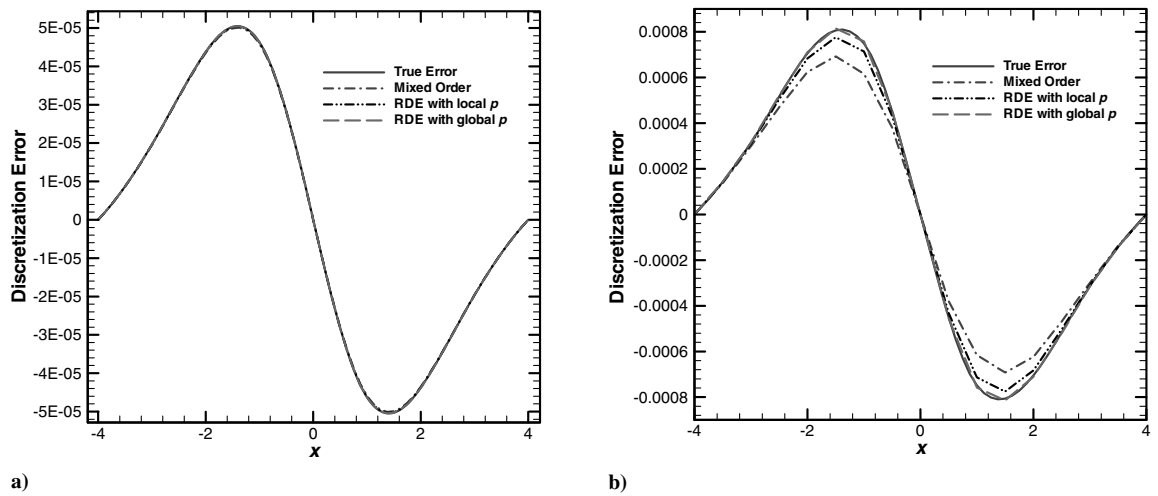


Fig. 13 Discretization error estimates for the nearby problem at a Reynolds number of 8 with a fine mesh of a) 257 nodes and b) 65 nodes.

fits at spline zone boundaries, as compared with cubic splines, which only enforce C^2 continuity. When applied as a discretization error estimator, the MNP approach is shown to provide accurate error estimates, while only requiring two calculations: the original problem and the nearby problem on the same grid. The MNP approach thus does not suffer from the problem of requiring multiple asymptotic grid solutions as do the extrapolation-based error estimators.

MNP is also demonstrated as an effective framework for assessing different discretization error estimators. Of the three extrapolation-based error estimators examined, Richardson extrapolation with the formal order of accuracy is found to provide the best error estimates when compared to the true error for the cases examined. Although this conclusion cannot be extended to all problems, it nevertheless provides an additional contribution to the growing body of evidence on the effectiveness of discretization error estimators. In all cases, the error estimates in the nearby problem are found to be identical to those seen in Burgers's original equation. This result further supports the use of MNP, both as a discretization error estimator and as a framework for evaluating error estimators.

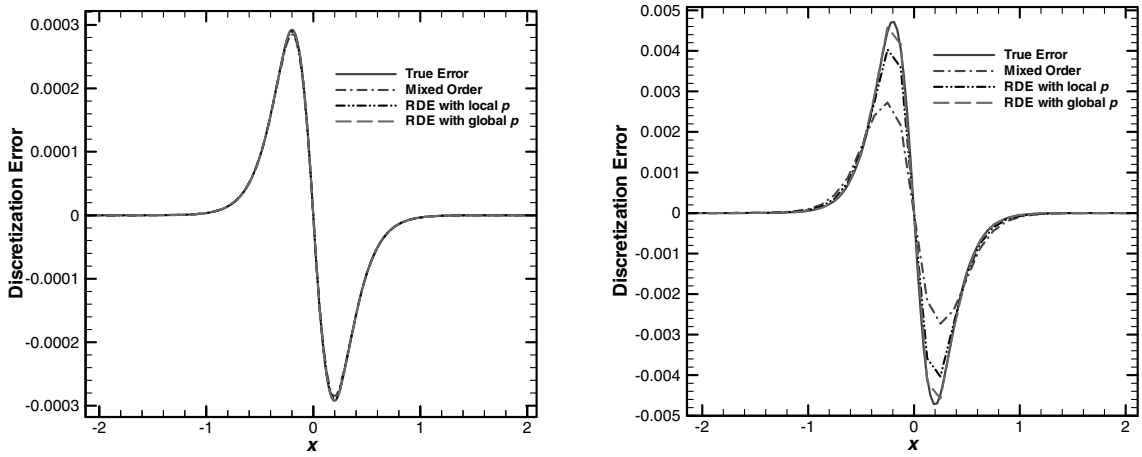


Fig. 14 Discretization error estimates for the nearby problem at a Reynolds number of 64 with a fine mesh of a) 1025 nodes and b) 257 nodes.

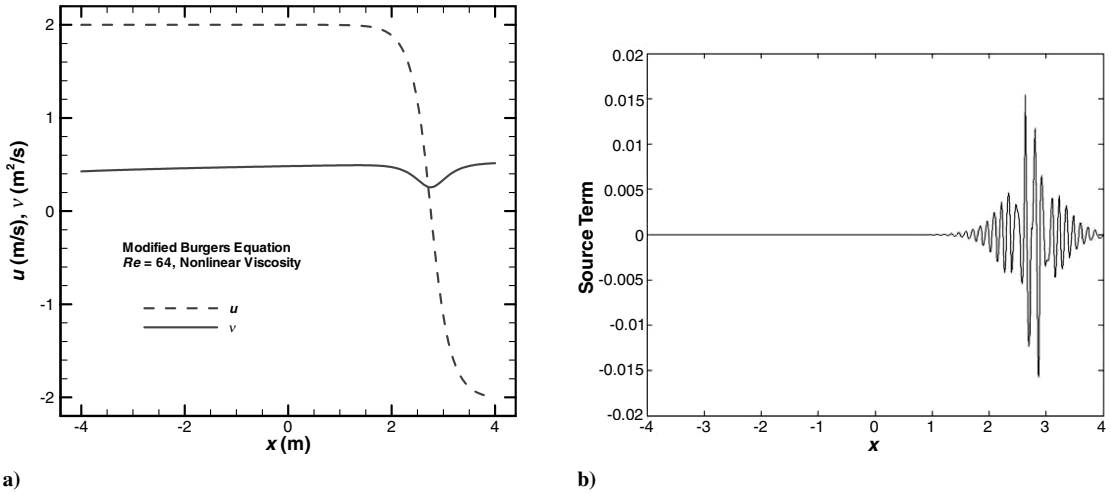


Fig. 15 Modified form of Burgers's equation with a nonlinear viscosity and a nominal Reynolds number of 64: a) solution and viscosity variation, and b) source term distribution using 65 spline points.

MNP has been successfully demonstrated for steady-state Burgers's equation, a 1-D, quasi-linear, second-order differential equation. However, the most difficult aspect of MNP is the generation of an accurate curve fit to the underlying numerical solution. To achieve practical value for engineering simulations, MNP must be extended to 2-D, 3-D, and 4-D problems (i.e., in space and time). Accurate curve-fitting procedures in multiple dimensions are expected to pose additional challenges. Prior work by Rouff on C^k continuous spline fits in multiple dimensions [22] may offer some relief to this multidimensional curve-fitting problem. Finally, the extension of MNP to complex flows (e.g., turbulence, chemistry, shock waves) is expected to provide additional challenges.

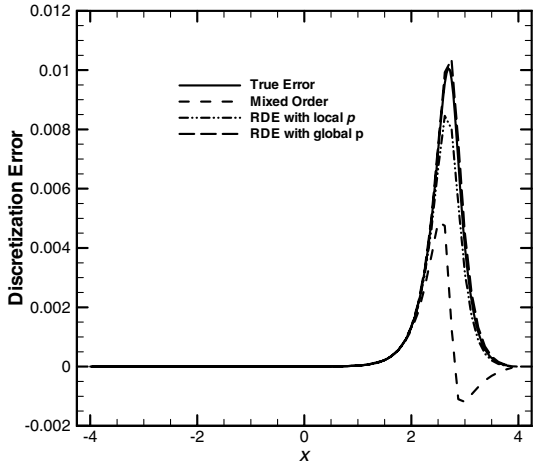


Fig. 16 Discretization error of the modified Burgers's equation with Reynolds number 64 using a finest mesh of 257 nodes.

Acknowledgments

The authors would like to thank Kevin Copps and Ryan Bond of Sandia National Laboratories for their helpful comments on this manuscript. This work was funded by Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations with an Application to the Stresses in a Masonry Dam," *Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 210, March 1910, pp. 307–357.

[2] Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM, 1998.

[3] Babuska, I., and Miller, A., "Post-Processing Approach in the Finite Element Method—Part 3: A Posteriori Error Estimates and Adaptive

Mesh Selection," *International Journal for Numerical Methods in Engineering*, Vol. 20, No. 12, 1984, pp. 2311–2324.
[4] Zienkiewicz, O. C., and Zhu, J. Z., "The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 1: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 7, 1992, pp. 1331–1364.
[5] Zienkiewicz, O. C., and Zhu, J. Z., "The Superconvergent Patch Recovery and a Posteriori Error Estimates, Part 2: Error Estimates and Adaptivity," *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 7, 1992, pp. 1365–1382.
[6] Knupp, P., and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, edited by K. H. Rosen, Chapman and Hall/CRC, Boca Raton, FL, 2003.
[7] Roy, C. J., Nelson, C. C., Smith, T. M., and Ober, C. C., "Verification of Euler / Navier-Stokes Codes Using the Method of Manufactured Solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, No. 6, 2004, pp. 599–620.
[8] Roy, C. J., "Review of Code and Solution Verification Procedures for Computational Simulation," *Journal of Computational Physics*, Vol. 205, No. 1, 2005, pp. 131–156.
[9] Lee, S., and Junkins, J. L., "Construction of Benchmark Problems for Solution of Ordinary Differential Equations," *Shock and Vibration*, Vol. 1, No. 5, 1994, pp. 403–414.
[10] Junkins, J. L., and Lee, S., "Validation of Finite-Dimensional Approximate Solutions for Dynamics of Distributed-Parameter Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 87–95.
[11] Roy, C. J., and Hopkins, M. M., "Discretization Error Estimates Using Exact Solutions to Nearby Problems," AIAA Paper 2003-0629, Jan. 2003.
[12] Hopkins, M. M., and Roy, C. J., "Introducing the Method of Nearby Problems," *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*, edited by P. Neittaanmaki, T. Rossi, S. Korotov, E. Onate, J. Periaux, and D. Knorzer, University of Jyväskylä (Jyvaskyla), Jyväskylä, Finland, July 2004.
[13] 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.
[14] Raju, A., Roy, C. J., and Hopkins, M. M., "Evaluation of Discretization Error Estimators Using the Method of Nearby Problems," AIAA Paper 2005-4993, June 2005.
[15] Raju, A., "Discretization Error Estimation Using the Method of Nearby Problems: One-Dimensional Cases," M.S. Thesis, Aerospace Engineering Dept., Auburn Univ., Auburn, AL, 2005.
[16] Zadunaisky, P. E., "On the Estimation of Errors Propagated in the Numerical Integration of Ordinary Differential Equations," *Numerische Mathematik*, Vol. 27, No. 1, 1976, pp. 21–39.
[17] Skeel, R. D., "Thirteen Ways to Estimate Global Error," *Numerische Mathematik*, Vol. 48, No. 1, 1986, pp. 1–20.
[18] Benton, E. R., and Platzman, G. W., "A Table of Solutions of the One-Dimensional Burger's Equation," *Quarterly of Applied Mathematics*, Vol. 30, No. 2, July 1972, pp. 195–212.
[19] O'Neal, P. V., *Advanced Engineering Mathematics*, 3rd ed., Wadsworth Publishing, Belmont, CA, 1991, pp. 400–412.
[20] Mullges, G. E., and Uhlig, F., *Numerical Algorithms with Fortran*, Springer–Verlag, Berlin, 1996.
[21] Roy, C. J., "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 595–604.
[22] Rouff, M., "The Computation of C^k Spline Functions," *Computers & Mathematics with Applications*, Vol. 23, No. 1, 1992, pp. 103–110.

C. Bailly
Associate Editor