Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 225 (2007) 1827–1848
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Accurate monotonicity- and extrema-preserving methods through adaptive nonlinear hybridizations

William J. Rider [a,*], Jeffrey A. Greenough [b], James R. Kamm [a]

[a] *Applied Physics Division, Los Alamos National Laboratory, MS F699, Los Alamos, NM 87545, USA*
[b] *Lawrence Livermore National Laboratory, Livermore, CA 94550, USA*

Received 30 August 2004; received in revised form 15 February 2007; accepted 20 February 2007
Available online 12 March 2007

Abstract

The last 20 years have seen a wide variety of high-resolution methods that can compute sharp, oscillation-free compressible flows. Here, we combine a complementary set of these methods together in a nonlinear (hybridized) fashion. Our base method is built on a monotone high-resolution Godunov method, the piece-wise parabolic method (PPM). PPM is combined with WENO methods, which reduce the damping of extrema. We find that the relative efficiency of the WENO methods is enhanced by coupling them with the relatively inexpensive Godunov methods. We accomplish our hybridizations through the use of a bounding principle: the approximation used is bounded by two nonlinearly stable approximations. The essential aspect of the method is to have high-order accurate approximations bounded by two non-oscillatory (nonlinearly stable) approximations. The end result is an accuracy-, monotonicity- and extrema-preserving method. These methods are demonstrated on a variety of flows, with quantitative analysis of the solutions with shocks.

1. Overview

Early linear differencing methods have been superseded by nonlinear methods using limiters that hybridized the various first generation methods based on the nature of the local solution. These hybrid schemes comprised the initial high-resolution methods, which combined the accuracy of the oscillatory schemes with the monotonicity and physically significant solutions of the first-order monotone methods. These methods pursue the goal of exploiting the best of each linear method while avoiding the unfavorable characteristics that also plague the solutions. A veritable alphabet soup of the methods [19,20] has arisen in the last two decades including high-resolution Godunov methods (MUSCL [35], PLM [5], PPM [8]), FCT [2], UNO [14], ENO [14], WENO [18], and TVD [13]. Extending the PPM method will be the focus of this paper although the original PPM method [8,37] has been extended by Woodward [38] using a different approach then the one taken in this paper, but with similar goals. Each of these methods can be identified as having a different principle

[*] Corresponding author. Present address: Computational Shock and Multiphysics, Sandia National Laboratories, MS 0378, Albuquerque, NM 87185-0378, United States. Tel.: +1 505 844 1572; fax: +1 505 844 0918.
E-mail addresses: wjrider@sandia.gov (W.J. Rider), greenough1@llnl.gov (J.A. Greenough), kammj@lanl.gov (J.R. Kamm).

doi:10.1016/j.jcp.2007.02.023

leading to a different algorithm for achieving high-resolution results. At their core, however, they are all based on nonlinear hybridizations of several linear methods.

The methods described in this paper constitute a step in this progression by taking the *nonlinear* methods and nonlinearly combining or hybridizing them to achieve higher resolution results than the original nonlinear methods. Included will be a combination of PPM and WENO, relying upon new nonlinear "limiters" to select which nonlinear principle to follow locally. This approach can be applied to other combinations of methods such as MUSCL and UNO or PPM and ENO for example. The principle we employ is to assure that approximations used to advance the conservation laws are bounded by nonlinearly stable methods; this assures the nonlinear stability of the result because the resulting method can be written as a convex combination of nonlinearly stable methods. The accuracy and stability properties of the methods are a by-product of the median function used to compose the methods [17].

We have studied the connection between method accuracy and error in a preceding paper [12]. In that paper, we examined the relative computational effort of PLM and fifth-order WENO. There we found that the computational simplicity of PLM and its relatively high accuracy compared to WENO made it more efficient on the problems examined. WENO, however, outperformed PLM on a per-mesh-cell basis for a problem with substantial high-frequency structure. Our aim in this work is to provide a means of achieving the best of each method: the simplicity (efficiency) and accuracy (resolution) of compressible flow structures of PLM *combined with* the formal accuracy and the resolution of extrema available with WENO.

The success of our new methods will be judged by the solution accuracy they provide relative to computational effort normalized by comparable existing methods. Quantifying error will provide guidance for subsequent development and an assessment of progress. In computing error, one must be cognizant that shock captured solutions with shocks are demonstrably first-order, at best, in convergence. Majda and Osher [21] proved that a coupled first-order system of hyperbolic PDEs loses high-order accuracy for all characteristics emanating from a discontinuity. This result has been confirmed by others in recent years [11,4], and holds even for linear systems of equations.

The process of quantifying error is based on the error ansatz,

$$E_p = \|S - A\|_p = Ch^\alpha + \text{higher order terms}, \tag{1}$$

where E is the error, S is the numerical solution, A is the true solution, $p = (1, 2, \ldots, \infty)$ is the norm used, and h is the mesh size, with the quantities C and α to be determined through mesh refinement. If an analytic solution, A, is known, solutions on two grids are needed to determine these values. Under this condition the rate of convergence for the p-norm is simply given by

$$\alpha_p = \frac{\log(E_{p,2}/E_{p,1})}{\log(h_2/h_1)}, \tag{2}$$

where we assume that the higher number corresponds to a finer grid ($h_2 < h_1$, this equation also assumes that the refinement ratio is equal $h_2/h_1 = h_3/h_2$). If there is no analytic solution, three grids are needed, because the solution itself must be estimated along with the convergence rate and constant. Without the analytic solution, the numerical solution should be compared at identical locations (or more properly volumes). In all the cases presented herein, we have volume averaged the fine grid solutions onto the coarse grid. We believe that this form of averaging is the most consistent approach for the solution of conservation laws with discontinuous solutions. For example, if the grid were refined by a factor of two in one dimension, the solutions would be compared on the coarse grid and the coarsened fine grid solution would be produced by averaging two fine grid cells to produce a single value on the coarse grid. With these caveats in mind, this leads to a similar formula to the case where the analytical solution is known:

$$\alpha_p = \frac{\log[\|S_{p,3} - S_{p,2}\|_p / \|S_{p,2} - S_{p,1}\|_p]}{\log(h_2/h_1)}, \tag{3}$$

again, noting that we are assuming that the refinement ratio is constant.

In the following section, we will introduce our methods and describe them in detail. Then we will test the algorithms on a set of one-dimensional problems. The first of these tests are two linear advection problems. One contains multiple waveforms in the initial conditions and the other is a single Gaussian pulse. The next set of one-dimensional tests are nonlinear and consist of the simple wave evolution [9], Sod's (Sod) shock tube [31] problem which both have analytic solutions for comparison. The next set of one-dimensional nonlinear tests, the Woodward and Colella (Blast) blast wave problem [36] and the Shu-Osher (SO) problem [30], require a high-resolution fiducial solution to be constructed for determining accuracy since there is no analytic solution. We use a mesh eight times finer than any of the test problem meshes for generating the fiducial solution. The SO problem is run in with two variations. One is examined at early time when the post-shock flow consists of four smooth compressed waves and the other is examined at late time with 16 such waves after they have steepened into a train of *N*-waves. For a two-dimensional test we use the Mach 10 double Mach reflection. Finally, we examine the performance of the new method on the three-dimensional Taylor–Green vortex problem [3,29].

2. Extrema, accuracy and monotonicity-preserving methods

In this work, we consider the solution to the compressible Euler equations in one-, two- or three-dimensions. The equations are written in conservation form:

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{4}$$

where $\mathbf{U} = (\rho, \mathbf{m}, E)^{\mathrm{T}}$. We will restrict our attention to the ideal gas equation of state, $p = (\gamma - 1)\rho e$. In the above, ρ is the density, $\mathbf{m}$ is the momentum, p is the pressure, and E is the total energy, $E = \rho e + 1/2(\mathbf{m} \cdot \mathbf{m})/\rho$. The flux vector is $\mathbf{F}(\mathbf{U}) = (\mathbf{m}, \mathbf{mm}/\rho + p\delta, \mathbf{m}(E+p)/\rho)^{\mathrm{T}}$.

We also use the Euler equations in primitive form:

$$\mathbf{V}_t + A_{p,x}(\mathbf{V})\frac{\partial \mathbf{V}}{\partial x} + A_{p,y}(\mathbf{V})\frac{\partial \mathbf{V}}{\partial y} + A_{p,z}(\mathbf{V})\frac{\partial \mathbf{V}}{\partial z} = 0, \tag{5}$$

with $\mathbf{V} = (\rho, \mathbf{u}, \rho e, p)^{\mathrm{T}}$. In one dimension, $A_p = M^{-1}A_c$, $A_c = \partial \mathbf{F}_x(\mathbf{U})/\partial \mathbf{U}$, $M = \partial \mathbf{U}/\partial \mathbf{V}$, with two and three dimensions following similarly. The vector of primitive variables is redundant (p and ρe both describe the linearized evolution of energy). This form is useful in defining the numerical solutions [7,24]. Despite our use of the full multi-dimensional equations, we use one-dimensional algorithms, with which multi-dimensional integration methods are constructed through the use of operator splitting.

The details of the implementation matter a great deal in contributing to the accuracy of the method. Among the most important of these is the use of characteristic variables for the basic finite difference approximations. These are constructed through an eigenvalue decomposition of the flux Jacobian in each zone. It takes the following form for the *m*th characteristic variable in the *j*th zone:

$$\mathbf{w}_k = L(\mathbf{V}_j)\mathbf{V}_k. \tag{6}$$

Once one dispenses with strictly monotonicity-preserving methods the use of characteristic variables is helpful – if not essential for very difficult problems involving strong shocks.

We use the characteristic decomposition defined in terms of the primitive variables. The primitive variable form provides simpler characteristic variables, $\mathbf{w} = L\mathbf{V}$, defined by the eigenvalue decomposition of $A_p = R_p \Lambda L_p$. The characteristic decomposition in primitive variables is equivalent to the conservative variable decomposition through a linear transformation [17]. The conservative Jacobian can be recovered with the conservative right-eigenvectors, $A_c = R_c \Lambda L_p$, the eigenvalues, and the primitive characteristic variables.

In the case of the present algorithm, we use the following characteristic decomposition with six characteristic variables matching our six primitive variables. These characteristics form a vector as follows:

$$\mathbf{w} = L\mathbf{V} = \begin{bmatrix} \rho_k \rho_j^{-1} - p_k(\rho c)_j^{-1} \\ v_k c_j^{-1} \\ w_k c_j^{-1} \\ \rho_k e_k (\rho c^2)_j^{-1} - p_k(p_j + \rho_j e_j)(\rho c^2)_j^{-2} \\ 1/2(p_k(\rho c^2)_j^{-1} - u_k c_j^{-1}) \\ 1/2 p_k (\rho c^2)_j^{-1} + u_k c_j^{-1} \end{bmatrix},$$

where c is the sound speed, v and w are the transverse velocities, k is any zone in the stencil, and j is cell being reconstructed. Once any operation is conducted on these characteristic variables, the primitive variables can be recovered by post-multiplying by the right eigenvectors,

$$R = \begin{bmatrix} \rho_j & 0 & 0 & 0 & \rho_j & \rho_j \\ 0 & 0 & 0 & 0 & -c_j & c_j \\ 0 & c_j & 0 & 0 & 0 & 0 \\ 0 & 0 & c_j & 0 & 0 & 0 \\ 0 & 0 & 0 & (\rho c^2)_j & p_j + \rho_j e_j & p_j + \rho_j e_j \\ 0 & 0 & 0 & 0 & (\rho c^2)_j & (\rho c^2)_j \end{bmatrix}.$$

This returns a vector of six primitive variables $(\rho, u, v, w, \rho e, p)^{\mathrm{T}}$.

Next, we review the basic design for a Godunov-type approach for solving the Euler equations.

1. Compute the polynomial interpolation in a cell and employ nonlinear stability enhancing constraints,
$$\mathbf{V}^n(\mathbf{x}) = \mathbf{P}(\theta), \tag{7}$$
where
$$\mathbf{P}(\theta) = \mathbf{P}_0 + \mathbf{P}_1\theta + \mathbf{P}_2\theta^2 + \ldots, \quad \theta = \frac{x - \bar{x}}{\Delta x}.$$
2. Apply time-accurate differencing. The details of the PPM interpolation will be discussed below in Algorithm 2.1.
3. Solve the Riemann problem at the cell interfaces
$$\mathbf{F}(\mathbf{V})^{n+1/2} = R(\mathbf{V}_-, \mathbf{V}_+), \tag{8}$$
where $R(,)$ is the (approximate) Riemann solver given these primitive variable values. The Riemann solver is an essential aspect of the resulting solution fidelity. The chief aspect that directly impacts solution fidelity is the approximation of rarefactions. Recall that a rarefaction is a continuous and adiabatic transition. The vast majority of Riemann solvers simply approximate a rarefaction as a jump discontinuity either directly or as a necessary side effect. In some cases, this simplification can significantly degrade solution accuracy. By approximating the rarefaction as continuous in the Riemann solution, one can decrease the error in computing rarefaction solutions in simple shock tubes by more than 50%. Furthermore, this change in approximation also removes sonic point "glitches" (expansion shocks) [27].
4. Advance the conservation laws (and conserved variables) using the Riemann fluxes. In a one-dimensional sweep, this step is simple,
$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{\Delta t}{\Delta x}(\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}).$$
5. Update the constitutive relations and prepare for the next time step
$$p_j^{n+1} = p\left(\rho_j^{n+1}, e_j^{n+1}\right). \tag{9}$$

These steps comprise a single dimensional sweep in what can be composited into a multi-dimensional method through Strang splitting [32], where the order of the sweeps can be alternated to give second-order accuracy (linearly). For example, a two-dimensional solution would composite an x-direction sweep,

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n - \frac{\Delta t}{\Delta x}(\mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j}),$$

followed by a y-direction sweep,

$$\mathbf{U}^{n+1}_{i,j} = \mathbf{U}^{n}_{i,j} - \frac{\Delta t}{\Delta y}(\mathbf{F}_{i,j+1/2} - \mathbf{F}_{i,j-1/2}).$$

This would form the algorithm to complete a single time step. Strang splitting would then invert the order in the next time step to a y-direction sweep followed by an x-direction sweep.

In examining typical computational results, several questions arise regarding what restricts the performance of the Godunov-type methods. The most severe loss of accuracy comes when the limiter reduces the scheme to first-order at extrema. We take the approach of Huynh [17] and Suresh [33] a step further by clarifying the various limiters, extending the order, and unifying the edge value limiting in keeping with Huynh's approach (less like that of Suresh and Huynh).

Algorithm 2. PPM godunov-type integrator

1. Begin with cell-centered conserved variables, $\mathbf{U}^n_j$.
2. Convert the conserved variables $\mathbf{U}^n_j$ to primitive variables $\mathbf{V}^n_j$, in cell j at time-step n.
3. Convert to local characteristic variables sufficient for the width of the computational stencil in cell j, $\mathbf{w}_k = L(\mathbf{V}_j)\mathbf{V}_k$, for $k = j - S$ to $k = j + S$ where S is the width of the difference stencil for cell j.
4. Select an algorithm to use for cell reconstruction and time centering in a cell from Algorithm 2.1 (PPM).
5. Convert cell-edged, time-centered variables, $\mathbf{w}^{n+1/2}_{j\pm1/2,\mp}$, to primitive variables, $\mathbf{V}^{n+1/2}_{j\pm1/2,\mp}$. Here, the subscript "−" ("+") denotes the value at the specified edge when approached from the right (left).
6. Compute cell-edged, time-centered fluxes using a Riemann solver to produce a single-valued flux from the multivalued reconstruction, $\mathbf{F}(\mathbf{V}^{n+1/2}_{j+1/2})$, where $\mathbf{V}^{n+1/2}_{j+1/2} = \text{Riemann}(\mathbf{V}^{n+1/2}_{j+1/2,-}, \mathbf{V}^{n+1/2}_{j+1/2,+})$.
7. Update cell-centered, conserved variables to the new time-level, $n+1$, using fluxes, $\mathbf{U}^{n+1}_j = \mathbf{U}^n_j - (\Delta t/\Delta x)[\mathbf{F}(\mathbf{V}^{n+1/2}_{j+1/2}) - \mathbf{F}(\mathbf{V}^{n+1/2}_{j-1/2})]$.
8. Evaluate constitutive relations using the new time conserved variables.
9. Return to 1.

When implementing multi-dimensional integrators for the Godunov-type methods we will use Strang splitting in this paper. Algorithm 1 is applied direction-by-direction to build two- and three-dimensional methods. If Algorithm 1 is represented symbolically as the operator $G\mathbf{U}$ with a subscript to denote the direction, the two-dimensional integrator for two time-steps takes the form $\mathbf{U}^{n+2} = G_xG_yG_yG_x\mathbf{U}^n$. In three dimensions the form using Strang splitting is $\mathbf{U}^{n+2} = G_xG_yG_zG_zG_yG_x\mathbf{U}^n$.

Nothing precludes extending the basic methods described here to an unsplit integration method as described in a number of papers. A good example of the methodology was introduced by Colella [6] and recently extended by Miller and Colella [22,23].

2.1. Piece-wise parabolic method (PPM)

The same ideas can be used to construct a third-order piece-wise parabolic method. The parabola is determined by three values: $w_{j,m}$, the integral average of the mth characteristic variable w in cell j, and the edge values, $w_{j\pm1/2,m}$, which are approximated by a linear combination of neighboring zone data. The parabolic reconstruction can be described in a brief sequence of steps in which one can vary the order of accuracy for the edge values and the method used for nonlinear stability.

Algorithm 2.1. Piece-wise Parabolic Reconstruction

1. The polynomial is piece-wise parabolic, $\mathbf{w}(x) = \mathbf{P}_j(\theta)$; $P(\theta) = \mathbf{P}_0 + \mathbf{P}_1\theta + \mathbf{P}_2\theta^2$; $\mathbf{P}_0 = 3\mathbf{w}_j/2 - (\mathbf{w}_{j-1/2} + \mathbf{w}_{j+1/2})/4$; $\mathbf{P}_1 = \mathbf{w}_{j+1/2} - \mathbf{w}_{j-1/2}$. $\mathbf{P}_2 = 3(\mathbf{w}_{j+1/2} - 2\mathbf{w}_j + \mathbf{w}_{j-1/2})$.
2. Choose the initial value for the edge values, $\mathbf{w}_{j\pm1/2}$. Algorithms 2.2.1, 2.2.2, 2.2.3, or Algorithm 2.2.4 can be used. If Algorithm 2.2.3 is used, the method is equivalent to a characteristic variable version of the classic PPM algorithm.
3. Choose a method to insure nonlinear stability: either a monotonicity Algorithm 2.1.1, or an accurate monotonicity- and extrema-preserving Algorithm 2.1.2. If a WENO initial edge value is chosen (Algorithm 2.2.4), this step can be avoided because this edge value is non-oscillatory.

4. The piece-wise linear polynomial provides a time-centered value of $\mathbf{w}^{n+1/2}_{j\pm1/2,\mp} = \mathbf{P}_0 + \mathbf{P}_1(\pm 1/2 - \boldsymbol{v}/2) + \mathbf{P}_2(1/4 \mp \boldsymbol{v}/2 + \boldsymbol{v}^2/3)$, where $v = \Delta t\boldsymbol{\lambda}/\Delta x$ is the Courant number for the characteristic being reconstructed. For PPM, we remove the linear, $\mathbf{P}_1$, and quadratic terms, $\mathbf{P}_2$, from the expansion. The filter can be defined by $\beta_+ = (\mathrm{sign}(v) + 1)/2$ and $\beta_- = (1 - \mathrm{sign}(v))/2$. $\mathbf{w}^{n+1/2}_{j\pm1/2,\mp} = \mathbf{w}_j + \beta_\pm(\mathbf{P}_0 + \mathbf{P}_1(\pm 1/2 - \boldsymbol{v}/2) + \mathbf{P}_2(1/4 \mp \boldsymbol{v}/2 + \boldsymbol{v}^2/3) - \mathbf{w}_j)$.

2.1.1. cPPM

The algorithm used to define the monotonicity of a piece-wise parabolic reconstruction can be written compactly. For PPM, monotonicity is assured if

$$\mathbf{w}_{j\pm1/2} \in [\mathbf{w}_j, \mathbf{w}_{j\pm1}]$$

and

$$\mathbf{w}_{j\pm1/2} \in [\mathbf{w}_j, 3\mathbf{w}_j - 2\mathbf{w}_{j\mp1/2}].$$

We use a median function to provide appropriately bounded values satisfying the conditions for monotonicity. The median function can be defined using a minmod function, where

$$\mathrm{minmod}(a, b) = 1/2(\mathrm{sign}(a) + \mathrm{sign}(b))\min(|a||, |b||),$$

and

$$\mathrm{median}(a, b, c) = a + \mathrm{minmod}(b - a, c - a).$$

An important property is that the median function's result is independent of the order of the arguments, $\mathrm{median}(a, b, c) = \mathrm{median}(c, a, b) = \mathrm{median}(b, c, a)$.

Algorithm 2.1.1. Piece-wise Parabolic Monotonicity

1. First, compute the following relations to bound the edge values by adjacent data,

   $$\mathbf{w}_{j\pm1/2} := \mathrm{median}(\mathbf{w}_j, \mathbf{w}_{j\pm1/2}, \mathbf{w}_{j\pm1}).$$

2. Finish the algorithm through bounding the edge values accounting for transport effects,

   $$\mathbf{w}_{j\pm1/2} := \mathrm{median}(\mathbf{w}_j, \mathbf{w}_{j\pm1/2}, 3\mathbf{w}_j - 2\mathbf{w}_{j\mp1/2}).$$

   This algorithm has the cost of four applications of the median function per zone.

2.1.2. xsPPM

Algorithm 2.1.2. Piece-wise Parabolic Accurate Monotonicity- and Extrema-Preserving

1. Check the monotonicity of the high-order edge values.
2. Compute the following relations to bound edge values with adjacent data,

   $$\mathbf{w}_{j\pm1/2} := \mathrm{median}(\mathbf{w}_j, \mathbf{w}_{j\pm1/2}, \mathbf{w}_{j\pm1}).$$

3. Finish the algorithm through bounding the edge values accounting for transport effects,

   $$\mathbf{w}^*_{j\pm1/2} = \mathrm{median}(\mathbf{w}_j, \mathbf{w}_{j\pm1/2}, 3\mathbf{w}_j - 2\mathbf{w}_{j\mp1/2}).$$

4. If both the high-order edge values are unmodified, then the algorithm is finished; return. This is implemented using a tolerance, such as $(w^*_{j\pm1/2,m} - w_{j\pm1/2,m})^2 < 10^{-12}$. These values are checked component-by-component. The edge values are a vector, $\mathbf{w}_{j\pm1/2} = (w_{j\pm1/2,1}, w_{j\pm1/2,2}, \ldots, w_{j\pm1/2,M})^{\mathrm{T}}$. Should either edge be modified, however, return to the initial edge values $w_{j\pm1/2,m}$ and use the following algorithm.
   (a) Compute a fifth-order WENO edge value $w_{j\pm1/2,m,*}$ using Algorithm 2.2.4.
   (b) If the monotone limiter indicates that there is a local extremum $(w^*_{j\pm1/2,m} = w_j)$,

i. Bound the WENO estimate with the local data, Algorithm 2.2.4

$$w_{j\pm1/2,m,*} = \text{median}(w_{j,m}, w_{j\pm1/2,m,*}, w_{j\pm1/2,m}).$$

ii. Compute the monotone WENO values

$$w^*_{j\pm1/2,m} = \text{median}(w_{j,m}, w_{j\pm1/2,m,*}, w_{j\pm1,m}),$$
$$w^*_{j\pm1/2,m} := \text{median}(w_{j,m}, 3w_{j,m} - 2w^*_{j\mp1/2,m}, w_{j\pm1/2,m,*}).$$

iii. Bound the final value by the monotone WENO and WENO values (and the original high-order value),

$$w_{j\pm1/2,m} = \text{median}(w^*_{j\pm1/2,m}, w_{j\pm1/2,m,*}, w_{j\pm1/2,m}).$$

(c) Otherwise the limiter indicates that the region is monotone and too steep to be approximated with the original high-order value.

i. Compute steepened high-order values, $w^*_{j\pm1/2,m}$, using Algorithm 2.2.3.

ii. The steepened value is not high-order accurate, so bring it to high-order using the median function and the original high-order and WENO values,

$$w^*_{j\pm1/2,m} = \text{median}(w^*_{j\pm1/2,m}, w_{j\pm1/2,m,*}, w_{j\pm1/2,m}).$$

iii. Compute monotone steepened high-order values,

$$w^*_{j\pm1/2,m} := \text{median}(w_{j,m}, w^*_{j\pm1/2,m}, w_{j\pm1,m}),$$
$$w^*_{j\pm1/2,m} := \text{median}(w_{j,m}, 3w_{j,m} - 2w^*_{j\mp1/2,m}, w^*_{j\pm1/2,m}).$$

iv. Bound the final value using the monotone steepened high-order values, the WENO result, and the original high-order values,

$$w_{j\pm1/2,m} = \text{median}(w^*_{j\pm1/2,m}, w_{j\pm1/2,m,*}, w_{j\pm1/2,m}).$$

The final edge values are the median of a high-order approximation, ENO/WENO, and a monotoncity-preserving value. This will produce an approximation that is correct to the spatial order of the high-order or ENO/WENO edge value, whichever is lower.

2.2. PPM edge variables

The formula for defining the edge variables will be defined as an algorithmic choice so that the approach can be referenced appropriately for our more complex methods. The simplest basic choice is a standard fourth-order centered edge value.

Algorithm 2.2.1. Fourth-order centered edge value

$$\mathbf{w}_{j+1/2} = \frac{7(\mathbf{w}_j + \mathbf{w}_{j+1}) - (\mathbf{w}_{j-1} + \mathbf{w}_{j+2})}{12}. \tag{10}$$

We have found that replacing the fourth-order differencing used in the original PPM can be improved by moving to higher order approximations such as a sixth-order or seventh-order edge value, leading to

Algorithm 2.2.2. Seventh-order centered edge value

$$\mathbf{w}_{j+1/2} = -\frac{3}{420}\mathbf{w}_{j-3} + \frac{25}{420}\mathbf{w}_{j-2} - \frac{101}{420}\mathbf{w}_{j-1} + \frac{319}{420}\mathbf{w}_j + \frac{214}{420}\mathbf{w}_{j+1} - \frac{38}{420}\mathbf{w}_{j+2} + \frac{4}{420}\mathbf{w}_{j+3}.$$

The original PPM method modifies the fourth-order edge value so that it is guaranteed to lie between the local point values. These modifications allow the resulting reconstruction to be steeper where the solution itself is steep (and under-resolved). We describe this method below and a modified version of this approach that produces a sixth-order edge value.

Algorithm 2.2.3. Fourth-order steepened edge value

1. Compute cell-centered monotone slopes

$$\begin{aligned}
&\mathbf{S}_j = 1/2(\mathbf{w}_{j+1} - \mathbf{w}_{j-1}),\\
&\mathbf{S}_j^M = 2\text{median}(0, (\mathbf{w}_{j+1} - \mathbf{w}_j), (\mathbf{w}_j - \mathbf{w}_{j-1})),\\
&\mathbf{S}_j := \text{median}(0, \mathbf{S}_j, \mathbf{S}_j^M).
\end{aligned} \tag{11}$$

2. Combine these slopes with the simple average at the cell-edge,

$$\mathbf{w}_{j+1/2} = 1/2(\mathbf{w}_j + \mathbf{w}_{j+1}) - \frac{1}{6}(\mathbf{S}_{j+1} - \mathbf{S}_j). \tag{12}$$

The WENO methods can be used for general reconstruction of variables, not only for constructing fluxes. They can also be used to provide states to input into a Riemann solution as a step in a finite volume methods. As noted earlier, the principal approximation in WENO methods is the use of a weighted combination of high-order stencils based on local smoothness, to produce a higher order stencil.

Algorithm 2.2.4. Fifth-order WENO edge value

1. Compute the third-order stencils

$$\begin{aligned}
&\mathbf{w}_{j+1/2,1} = (11\mathbf{w}_j - 7\mathbf{w}_{j-1} + 2\mathbf{w}_{j-2})/6,\\
&\mathbf{w}_{j+1/2,2} = (2\mathbf{w}_{j+1} + 5\mathbf{w}_j - \mathbf{w}_{j-1})/6,\\
&\mathbf{w}_{j+1/2,3} = (-\mathbf{w}_{j+2} + 5\mathbf{w}_{j+1} + 2\mathbf{w}_j)/6.
\end{aligned}$$

2. Compute the smoothness indicators

$$\begin{aligned}
IS_{j+1/2,1} &= \frac{13}{12}(\mathbf{w}_j - 2\mathbf{w}_{j-1} + \mathbf{w}_{j-2})^2 + \frac{1}{4}(3\mathbf{w}_j - 4\mathbf{w}_{j-1} + \mathbf{w}_{j-2})^2,\\
IS_{j+1/2,2} &= \frac{13}{12}(\mathbf{w}_{j+1} - 2\mathbf{w}_j + \mathbf{w}_{j-1})^2 + \frac{1}{4}(\mathbf{w}_{j+1} - \mathbf{w}_{j-1})^2,\\
IS_{j+1/2,3} &= \frac{13}{12}(\mathbf{w}_{j+2} - 2\mathbf{w}_{j+1} + \mathbf{w}_j)^2 + \frac{1}{4}(\mathbf{w}_{j+2} - 4\mathbf{w}_{j+1} + 3\mathbf{w}_j)^2.
\end{aligned}$$

3. Compute the weights

$$\alpha_{j+1/2,1} = 0.1/(IS_{j+1/2,1}), \quad \alpha_{j+1/2,2} = 0.6/(IS_{j+1/2,2}), \quad \alpha_{j+1/2,3} = 0.3/(IS_{j+1/2,3}),$$

where $\delta = 1 \times 10^{-40}$. Finally, normalize the weights using $\omega_{j+1/2,3} = \alpha_{j+1/2,k}/\sum\alpha$.

4. Compute the mapping weights [15] to improve accuracy. First set

$$\alpha_{j+1/2,k} := \frac{\alpha_{j+1/2,k}(\bar{\omega}_k + (\bar{\omega}_k))^2 - 3\bar{\omega}_k\alpha_{j+1/2,k} + (\alpha_{j+1/2,k})^2}{(\bar{\omega}_k)^2 + \alpha_{j+1/2,k}(1 - 2\bar{\omega}_k)},$$

where $\bar{\omega}_k$ is the optimal weight that produces a fifth-order approximation when combined with the other third-order edge values ($\bar{\omega}_1 = 1/10$, $\bar{\omega}_2 = 6/10$ and $\bar{\omega}_3 = 3/10$). Finally, normalize the weights again using $\omega_{j+1/2,k} = \alpha_{j+1/2,k}/\sum\alpha$.

5. Compute the final edge values using the weights

$$\mathbf{w}_{j+1/2} = \omega_{j+1/2,1}\mathbf{w}_{j+1/2,1} + \omega_{j+1/2,2}\mathbf{w}_{j+1/2,3} + \omega_{j+1/2,3}\mathbf{w}_{j+1/2,3}. \tag{13}$$

3. Results for the new methods

We have implemented these methods in the same manner utilizing as much common code as possible. All of the methods use similar data structures. We test these values with particular attention on both numerical error and computational effort.

The method that is closest to the original PPM method (using the same basic differencing for the scalar wave equation) is denoted as mPPM4s, for a monotone PPM with 4th-order steepened edge values. This will be used as our baseline scheme. It will be compared to the new xsPPM7 method introduced in the previous section.

3.1. One-dimensional results

3.1.1. Scalar wave equation

It is useful to study results for the simple scalar wave equation, $U_t + U_x = 0$, where order-of-accuracy can be experimentally determined. In addition, the basic properties of the method can be examined without the complications of nonlinear phenomena. We examine two basic results: a Gaussian pulse (to determine the design order-of-accuracy and verify claims about the nature of error associated with the new methods),

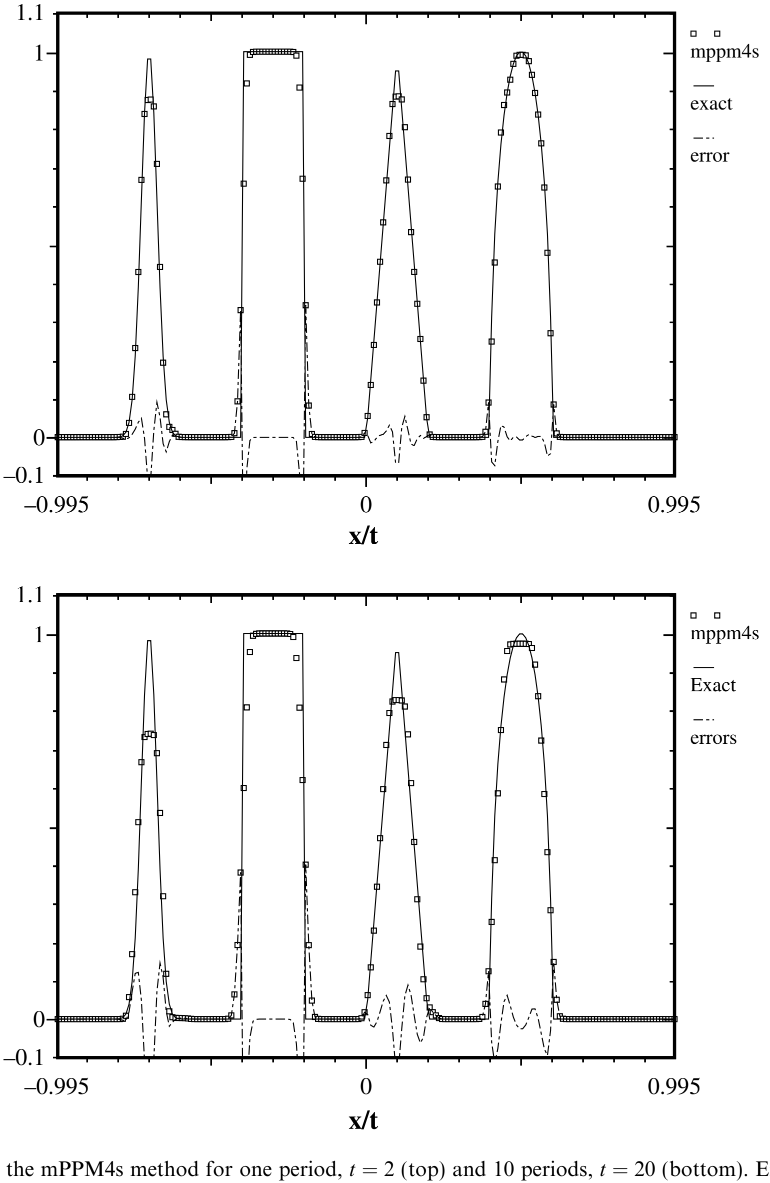


Fig. 1. The solution using the mPPM4s method for one period, $t = 2$ (top) and 10 periods, $t = 20$ (bottom). Error at $t = 2$, 1.64×10^{-2}. Error at $t = 20$, 3.10×10^{-2}. CPU time at $t = 20$ is 7.69 s.

and an advected profile containing four wave-forms. Our first problem will consist of a Gaussian, square wave, triangular wave, and a half-circle. The results provide substantial qualitative evidence of the character of solutions that can be expected with our methods.

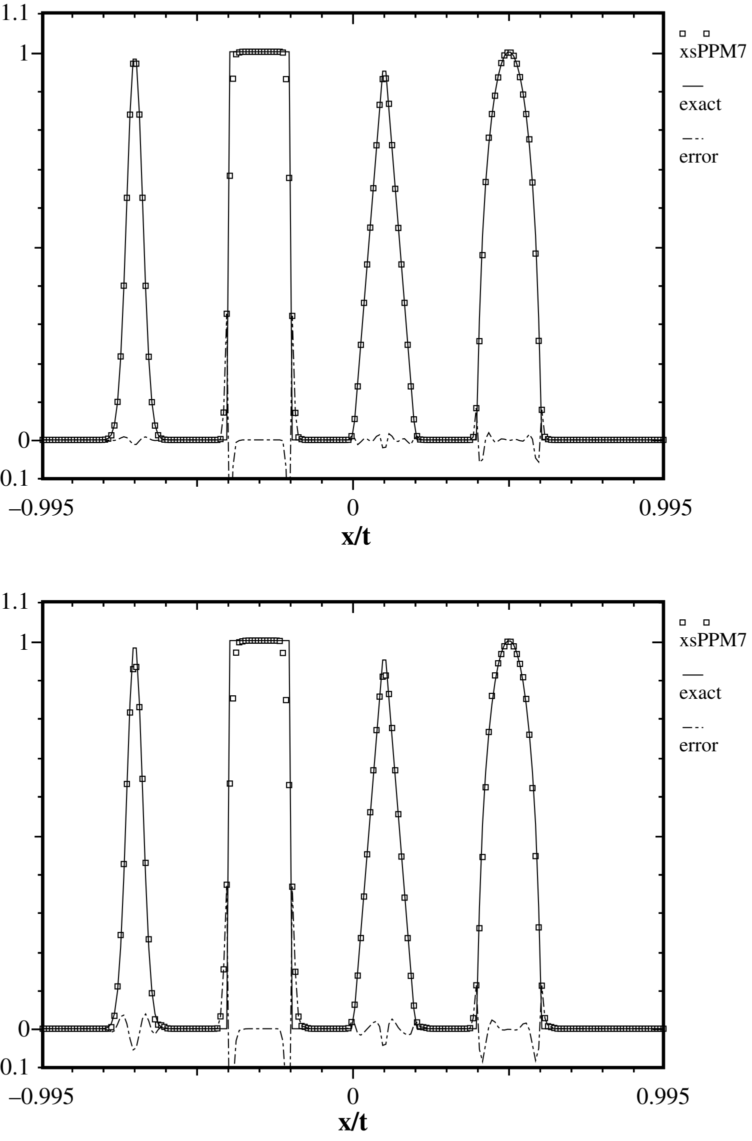


Fig. 2. The solution using the xsPPM7 method for one period, $t = 2$ (top) and 10 periods, $t = 20$ (bottom). Error at $t = 2$, 1.15×10^{-2}. Error at $t = 20$, 1.84×10^{-2}. CPU time at $t = 20$ is 8.30 s.

Table 1
The errors in L_1, L_2 and L_∞ norms for the advection of a Gaussian pulse one period on a grid using a Courant number of 0.01

Method	Grid	L_1	L_1 rate	L_2	L_2 rate	L_∞	L_∞ rate
mPPM4s	16	9.82×10^{-3}		1.39×10^{-2}		2.96×10^{-2}	
	32	2.82×10^{-3}	1.80	4.51×10^{-3}	1.62	1.27×10^{-2}	1.22
	64	7.35×10^{-4}	1.94	1.29×10^{-3}	1.81	3.96×10^{-3}	1.68
xsPPM7	16	9.09×10^{-4}		1.21×10^{-3}		2.73×10^{-3}	
	32	2.00×10^{-5}	5.51	2.84×10^{-5}	5.41	9.01×10^{-5}	4.92
	64	2.40×10^{-6}	3.06	3.62×10^{-6}	2.97	1.28×10^{-5}	2.82

The general character of the algorithms can be seen through the advection of a waveform with sufficient diversity to stress different aspects of the methods. We choose a waveform used in several other papers [33,10]:

$$u(x) = \begin{cases} \exp[-\log 2(x+0.7)^2/0.0009], & x \in [-0.8,-0.6]; \\ 1 & x \in [-0.4,-0.2]; \\ 1-|10(x-0.1)|, & x \in [0,0.2]; \\ \sqrt{1-100(x-0.5)^2}, & x \in [0.4,0.6]; \\ 0, & \text{otherwise.} \end{cases}$$

The boundary conditions are periodic. The advection of the wave form is given in Figs. 1 and 2 each on a grid of 200 cells as used in the aforementioned references.

In each figure, the exact wave form is given by a solid line, the dot-dashed line shows the error, and the numerical method's solution is given by the symbols. In both figures the upper plot is at time $t = 2$ after one period and the lower plot is at time, $t = 20$, after 10 periods. The figure caption gives the L_1 error

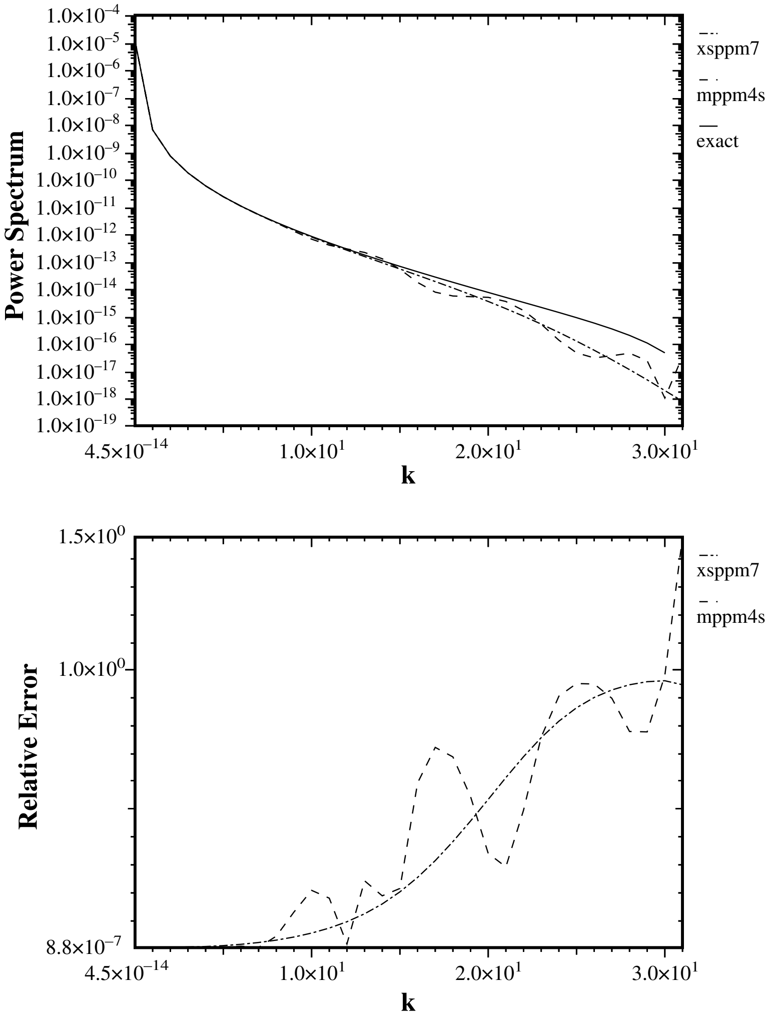


Fig. 3. A comparison of the power spectrum for solutions to the acoustic wave-breaking problem computed with the mPPM4s and xsPPM7 methods. The lower figure shows the relative error in the power spectrum using these two methods. The xsPPM7 scheme shows the deviation from the exact solution to be smooth and generally monotonic while the mPPM4s scheme deviates in a quasi-oscillatory fashion.

associated with the solution at these two time. The CPU time used for each full calculation after 10 periods is also displayed. Note that each calculation has an overhead of 0.257 s not related to the specifics of the method.

Our baseline for comparison is shown in Fig. 1 with the mPPM4s method. This method does particularly well in computing the square wave, but the monotonicity algorithm significantly clips the extrema. By using

Table 2
The errors in L_1, L_2 and L_∞ norms for the acoustic wave-breaking problem using two grids

Method	Grid	L_1	L_1 rate	L_2	L_2 rate	L_∞	L_∞ rate
mPPM4s	32	7.44×10^{-7}		1.21×10^{-6}		3.79×10^{-6}	
	64	1.08×10^{-7}	2.78	2.06×10^{-7}	2.55	8.23×10^{-7}	2.20
xsPPM7	32	2.89×10^{-7}		7.40×10^{-7}		3.04×10^{-6}	
	64	4.18×10^{-8}	2.79	1.20×10^{-7}	2.62	7.62×10^{-7}	1.99

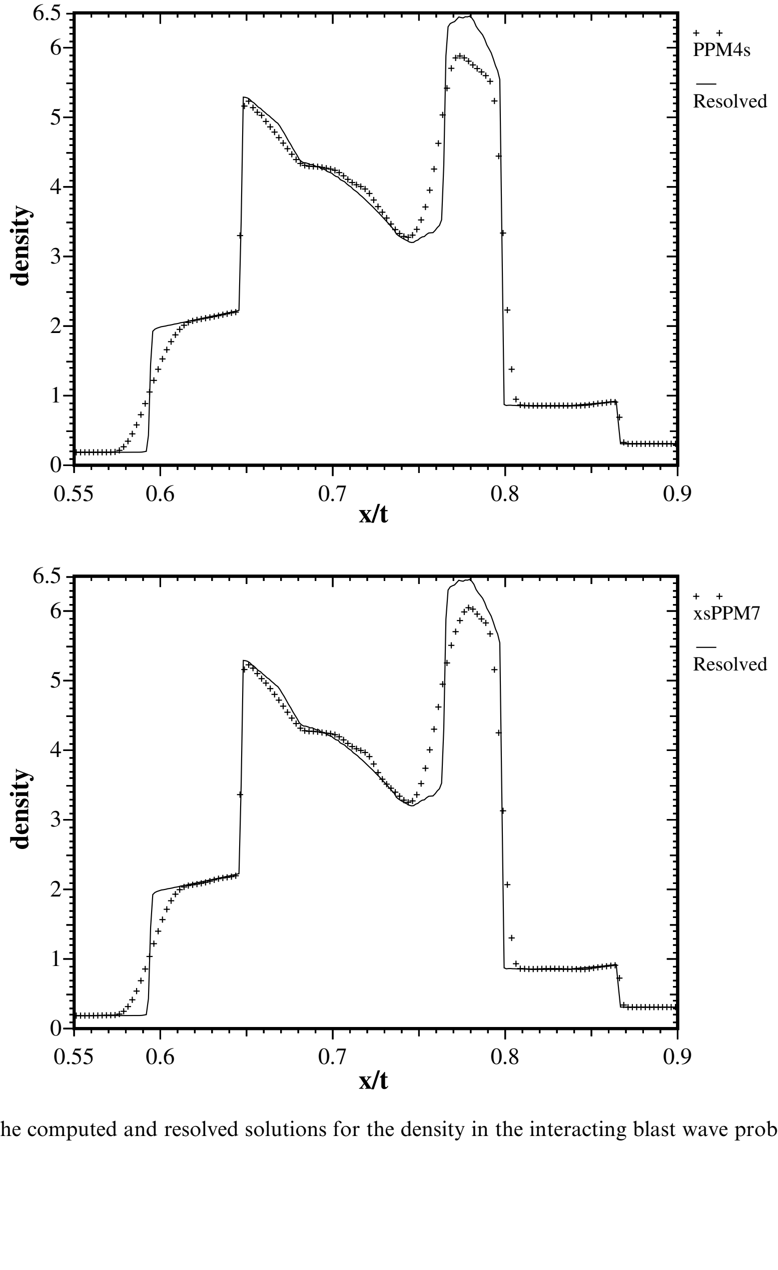


Fig. 4. A comparison of the computed and resolved solutions for the density in the interacting blast wave problem at $t = 0.038$ with 400 cells.

the ‘xs’ limiter we combine the positive effects of these algorithm modifications and recover low error, lack of clipping, and good square wave resolution, as shown in Fig. 2. These solutions are nominally more expensive than the original PPM method, but produce notably smaller errors.

Next, we evaluate the experimental order-of-accuracy for our methods. We use a Gaussian pulse, $\frac{1}{4}+\frac{1}{4}\exp(-30(x-1/2)^2)$, advected at a small Courant number, 0.01. This focuses the error on the spatial contribution by making the time error both small and nearly fixed in magnitude. Our results are shown in Table 1. The older monotone algorithm suffers a significant loss of accuracy and order-of-accuracy due to the nature of the enforcement of monotonicity. We see that the xsPPM method removes the loss of accuracy associated with the monotone limiting in the mPPM method and produces second-order accuracy in all norms measured. The xsPPM method produces high-order convergence presumably due to the melding of seventh-order base differencing with fifth-order WENO differencing at extrema, or significant under-resolution. These elements lead to a better convergence rate and significantly lower error as compared to the monotone method.

3.1.2. Compressible Euler equations

In order to study the behavior of the methods on smooth nonlinear flows that become shocked, we examine the acoustic wave breaking problem introduced by Cook and Cabot [9]. We compare methods at the same point as Cook and Cabot, 3/4 of the time at which the wave breaks and a shock forms. At that time an analytic solution is available for comparison. The spectral result is shown in Fig. 3. The new xsPPM method provides notably smoother behavior at higher wavenumbers. Furthermore, the deviation from the exact solution is smooth and generally monotonic. The mPPM4s method deviates from the analytic solution spectra in an quasi-oscillatory fashion. In Table 2, the errors for both methods are summarized. The convergence rates are very similar independent of the error norm. The magnitude of the errors are also similar for all norms considered differing at most by a factor of two.

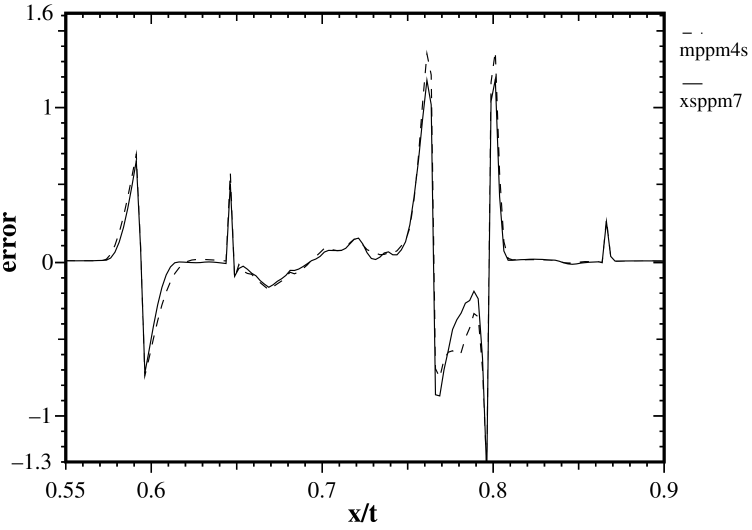


Fig. 5. A comparison of the errors for the density in the interacting blast wave problem at $t = 0.038$.

Table 3
Summary of density and pressure errors for the blast problem with 800 cells

Method	Density error	Pressure error	Cost
mPPM4s	3.42×10^{-2}	7.83×10^{-1}	1.00
xsPPM7	2.94×10^{-2}	6.38×10^{-1}	1.04

We next test our methods on problems with stronger nonlinearities. The first one is a simple self-similar shock tube problem, Sod's (Sod) shock tube [31] which has an analytic solution for comparison. On Sod we find that the two algorithms produce results that are virtually indistinguishable. The next two tests require a high-resolution fiducial solution for comparison and determining errors. The first of these is the blast wave problem of Woodward and Colella (Blast), which features very strong shocks that interact to form intricate structures [36]. The second is the Shu-Osher problem (SO) featuring compressed smooth waves that steepen into a train of N-waves [30]. We examine two variants of this problem, one at early time (with four smooth compressed waves in the post-shock region), and one at late time (with 16 such waves). For both the blast and SO problems we construct fiducial solutions using highly resolved computed results with eight times the mesh cells as the finest solution we are testing. This procedure for using a higher resolution fiducial solution was the same one used in [12] to study the accuracy of cPLM4 and WENO5.

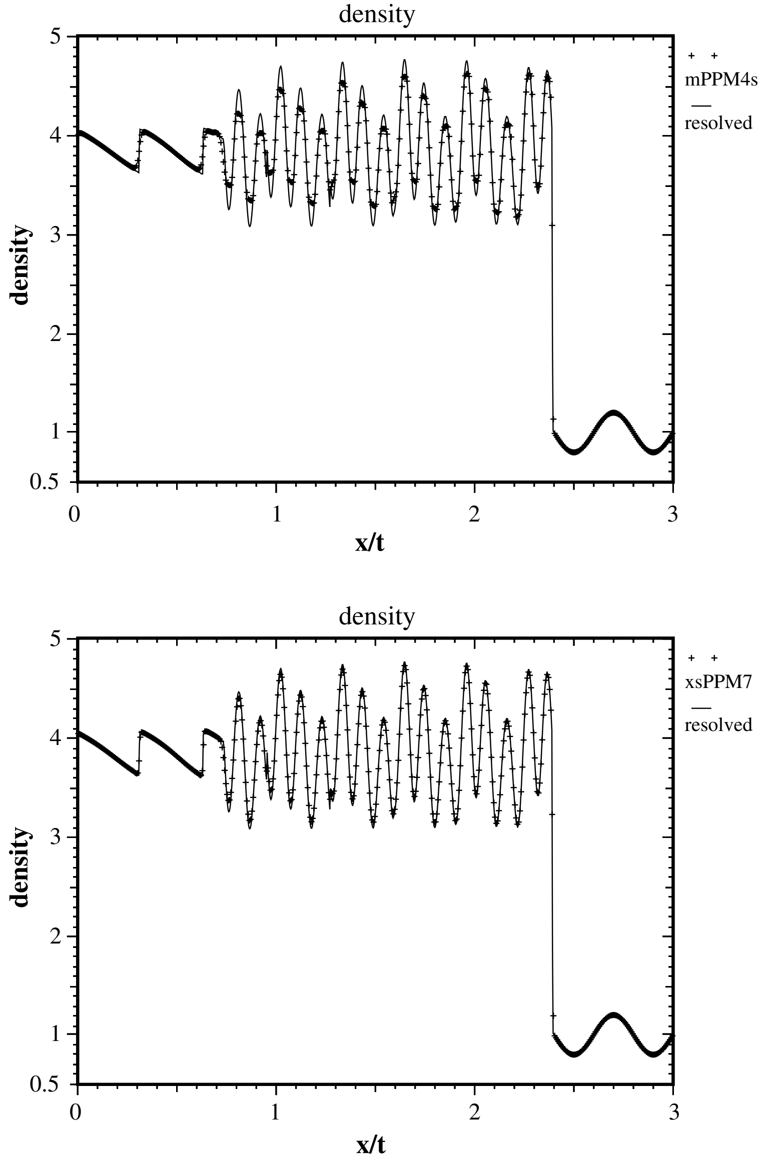


Fig. 6. A comparison of the computed and resolved solutions for the density in the Shu-Osher problem at $t = 1.8$ with 1600 cells.

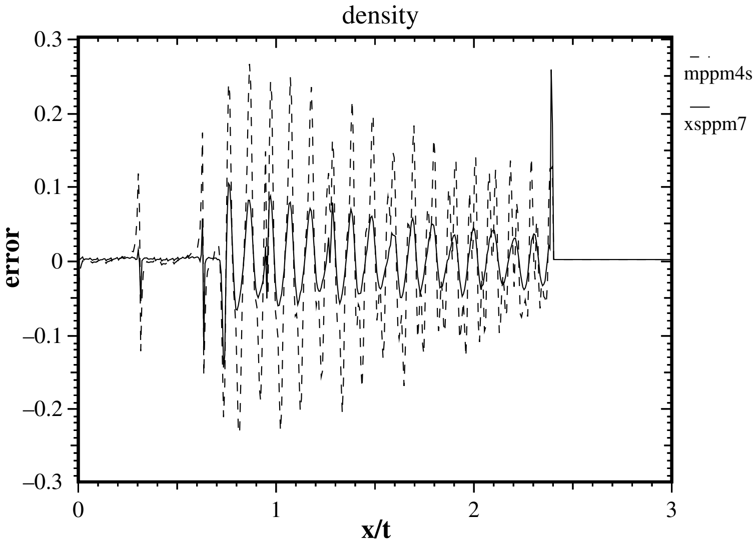


Fig. 7. A comparison of the errors for the density in the Shu-Osher problem at $t = 1.8$.

Table 4
Summary of density and pressure errors for the Shu-Osher problem with 3200 cells.

Method	Density error	Pressure error	Cost
mPPM4s	6.92×10^{-3}	1.32×10^{-2}	1.00
xsPPM7	2.90×10^{-3}	4.53×10^{-3}	1.02

The results for interacting blast wave problem are shown in Figs. 4 and 5 and Table 3. The solutions shown in Fig. 4 show that the xsPPM7 method reproduces extrema with less clipping than mPPM4s at the dominant peak near $x/t = 0.78$. This is confirmed by the error comparison shown in Fig. 5. The pressure error improvement is somewhat larger than the density improvement with xsPPM7 as Table 3 shows. From Table 3, the errors overall are similar with a low (4%) increase in computational cost.

The results for Shu-Osher's problem are shown in Figs. 6 and 7 and Table 4. Here, the lack of damping of extrema by xsPPM7 produces errors that are nearly two and a-half times smaller than the corresponding errors with mPPM4s. Table 4 shows the numerical comparison of error magnitudes. This improvement is accomplished with a very modest increase in CPU time (2%). Unlike Sod's shock tube or the interacting blast wave problem, where the two algorithms performed very similarly, the differences in the solution for this problem are more evident as shown in Fig. 6. The improvement in solution is confirmed by examining Fig. 7, which compares the errors.

3.2. Two-dimensional results

We now examine the performance of the methods on a complicated two-dimensional flow. The primary features of this flow make comparisons difficult. The first is the fact that we are computing solutions to the Euler equations, i.e. ignore physical viscosity. Hence, under grid refinement we will generate more and more fine scale structure since physical regularization provided by physical dissipation is being neglected. This flow also is dominated by slip lines emanating from shock triple points. The Kelvin–Helmhotz instability (shear instability) will also manifest similar grid dependent fine scale structure. Generally we refer to this flow, without physical dissipation to be ill-posed. Another way to view the generation of additional instabilities (primarily shear instabilities) under grid refinement is that it is the mesh spacing that is tripping the shear layer and forcing vortical roll-up. Decreasing the grid spacing is tantamount to increasing the Strouhal number thereby generating smaller scale vortical structures. Therefore, in the absence of physical regularization, more and more "detail" will be revealed under mesh refinement. This effect precludes the point-wise convergence studies undertaken in one-dimension. We must, for this problem rely, unfortunately, on more general comparisons as the grid is refined.

3.2.1. Mach 10 shock reflection

We begin by employing the standard Mach 10 shock reflection problem introduced by Woodward and Colella [36]. The problem features a Mach 10 shock in a $\gamma = 1.4$ gas reflecting from a 60° wedge. The problem is setup in the manner described in [36] (summarized by Sanders and Weiser [28]) with one key difference. The exact motion of the Mach 10 shock is prescribed along the upper boundary of the domain. We do the same except the shock transition is "smeared" over three zones in our implementation using a linear interpolation. The boundary conditions treated in this manner better match the numerical shock transition layer that develops with any shock capturing method.

The problem develops a complex Mach reflection with a triple point that features an unstable shear layer. The resulting wall jet is one of the principal distinguishing aspects of the flow. The wall jet is located behind the leading shock wave along the lower boundary of the flow-field and the vortex is associated with the nearby local density minimum, which is identified by the tight density contours along the lower right boundary in the following figures. The density fields mPPM4s and xsPPM7 are given in Figs. 8 and 9, respectively. We note that our results appear to have a slight issue with the "carbuncle" instability [26]. While not fully developed

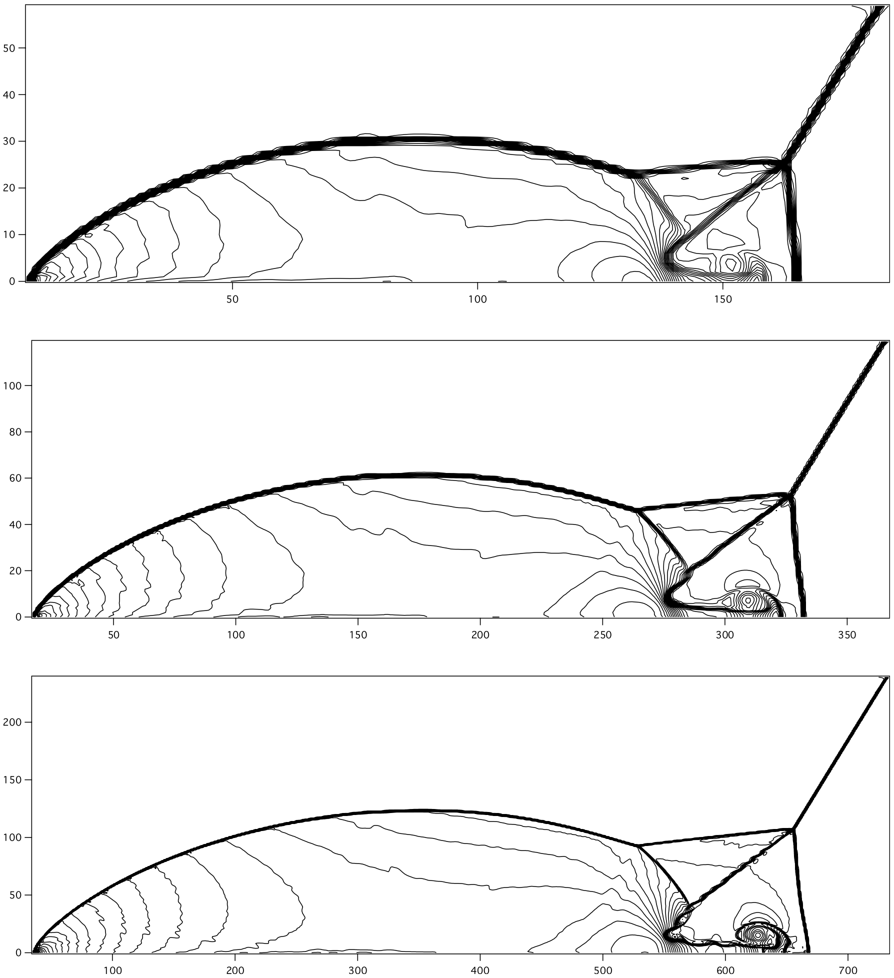


Fig. 8. A density contour plot of a mesh refinement sequence for the mPPM4s for the Mach reflection problem. The key feature to study is the triple point and Mach stem at the right hand side of the plot. The top plot uses a 240 × 60 grid, the middle plot 480 × 120 and the bottom plot is 960 × 240. The 60 density contours are chosen with a minimum of 1.5, a maximum of 25.1 with a spacing of 0.4.

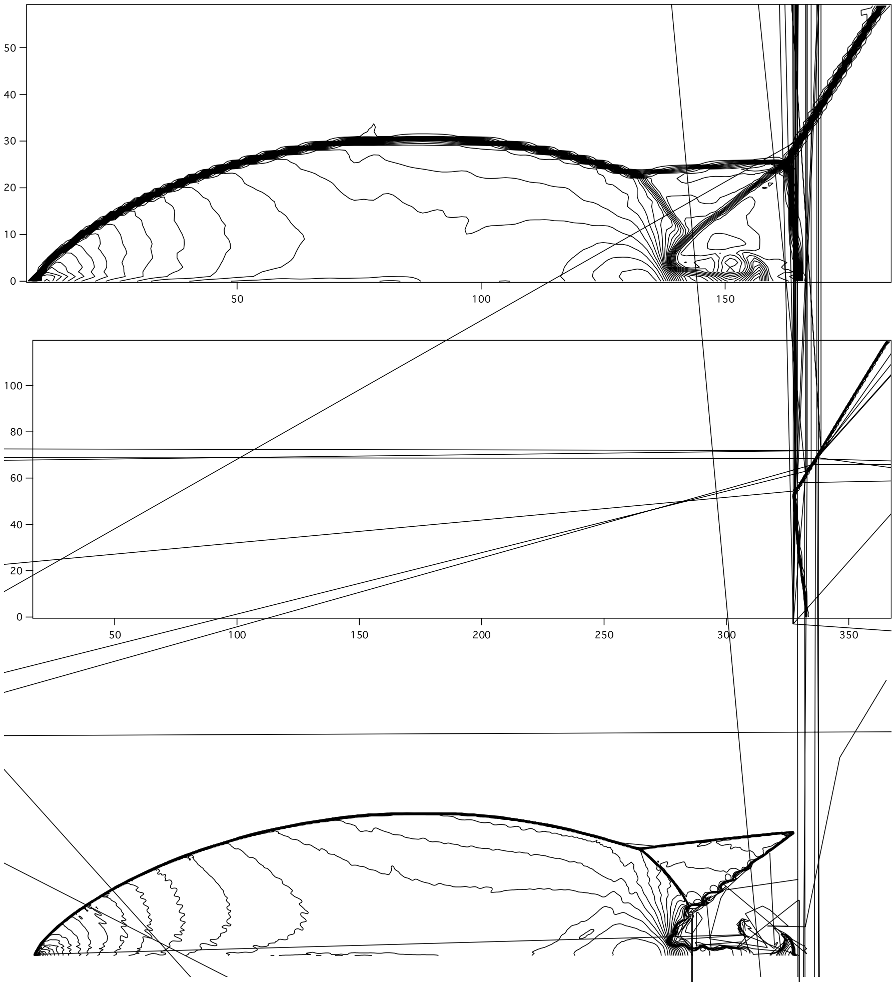


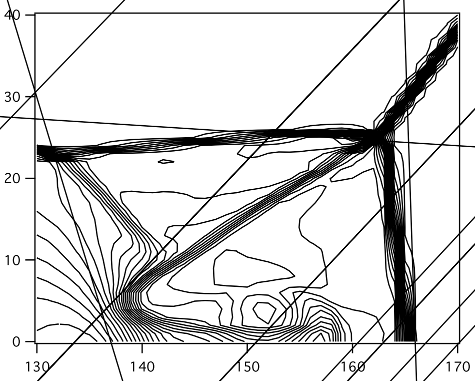
Fig. 9. A density contour plot of a mesh refinement sequence for the xsPPM7 for the Mach reflection problem. The key feature to study is the triple point and Mach stem at the right hand side of the plot. The top plot uses a 240 × 60 grid, the middle plot 480 × 120 and the bottom plot is 960 × 240. The 60 density contours are chosen with a minimum of 1.5, a maximum of 25.1 with a spacing of 0.4.

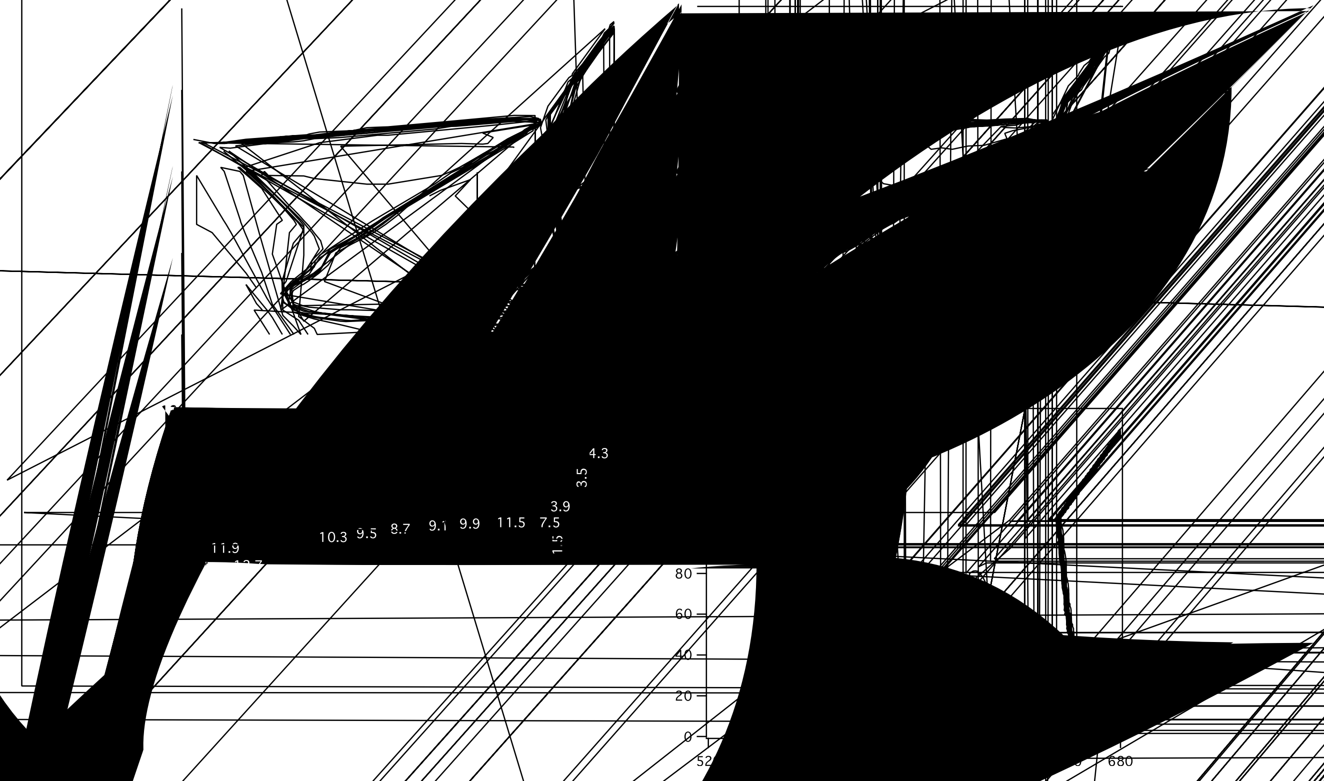
there is evidence that the instability is developing albeit at a low intensity. This problem could be eliminated through the use of a multi-dimensional artificial viscosity as suggested with the original PPM algorithm [8,6,24].

It is the strength of the main vortex in the wall jet that we examine more closely to ascertain the fidelity of our methods. The problem is solved on a domain of $x \in [0,4]$ and $y \in [0,1]$ to a time of 0.2, when all results are displayed. Visually, the xsPPM7 method shows enhanced resolution when compared with the monotonicity-preserving version of PPM. This conclusion is enhanced by the examination of the refinement sequence in Fig. 10 that shows a closeup of the triple point and the shear layer. The usual character that is related to the resolution is the degree of instability, which is substantially greater with the xsPPM7 method.

The relative degree of instability in the wall jet shear layer is used to assess method fidelity. A greater level of instability implies less numerical dissipation present in a method; this implies greater numerical accuracy. This reasoning follows from the proviso that the lower dissipation inhibits any unphysical structures from developing while allowing more vigorous shear instability.

In keeping with the theme of quantifying errors while recognizing the issues stated at the beginning of this section, we choose a different approach to examine the fidelity of our methods (although our conclusions would not change under a more traditional examination). Finally, we compare the minimum density and pres-





we examine this in Fig. 11. The xsPPM7 resolution. This is another indication of the

dimensions, we use the Taylor–Green vortex producing a low-Mach number ($u/c \approx 0.10$).

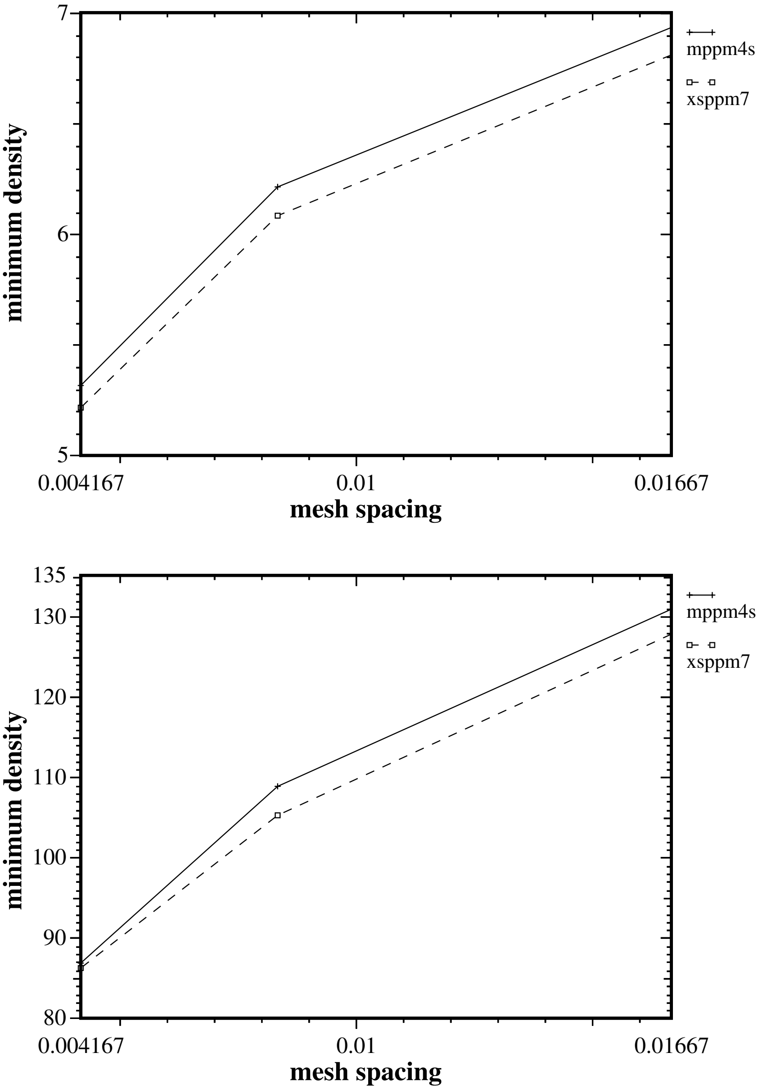


Fig. 11. A comparison of the mesh refinement for the double Mach reflection of the density and pressure minimum in the vortex rollup. The plots show the enhanced resolution of the xsPPM7 method as compared with the mPPM4s method.

$$
\begin{aligned}
&\rho(x,y,x) = 1,\\
&u(x,y,z) = \sin(x)\cos(y)\cos(z),\\
&v(x,y,z) = -\cos(x)\sin(y)\cos(z),\\
&w(x,y,z) = 0,\\
&p(x,y,z) = 100 + \frac{1}{16}[2+\cos(2z)][\cos(2x)+\cos(2y)-2].
\end{aligned}
$$

The flow is analytic at early times, but indications are that the flow may form a singularity at $t \approx 5.2$. This question is still open despite intense research over the past 30 years.

We show results at two times, "early" when the analytic accuracy of the flow solver can be meaningfully evaluated and compared, and "late" when the behavior of the method in the presence of nonlinearities is clear. Our chosen early time is $t = 2$, well before any singularity would form. In order to quantify the behavior of the methods, one can look at the evolution of kinetic energy. A more reliable metric for compressible flow is the

Table 5
The total kinetic energy loss at $t = 2$ in the Taylor–Green problem

Grid	mPPM4s	xsPPM7
$16 \times 16 \times 16$	6.48E − 03	3.00E − 03
$32 \times 32 \times 32$	7.44E − 04	5.58E − 04
$64 \times 64 \times 64$	1.10E − 04	1.05E − 04

Table 6
The total entropy gain at $t = 2$ in the Taylor-Green problem

Grid	mPPM4s	xsPPM7
$16 \times 16 \times 16$	2.58E − 03	1.19E − 03
$32 \times 32 \times 32$	2.85E − 04	2.23E − 04
$64 \times 64 \times 64$	5.34E − 05	5.15E − 05

entropy, which is well defined and should be a monotonically increasing function of time. We examine the performance of several methods on grids of 32^3 and 64^3 using the quantities of total integrated kinetic energy, $1/2\rho|\mathbf{u}|^2$, entropy, p/ρ^γ, and entrophy (vorticity squared), $1/2|\boldsymbol{\omega}|^2$.

We document the quantitative analyses of this problem in Tables 5 and 6. We find that the new method produces lower errors at the coarser resolutions by almost a factor of two. At the highest resolution considered, the errors are nearly the same with the xsPPM7 being about 5% lower. Additionally, if one desires a solution of equal accuracy, then the new methods are more efficient.

4. Concluding remarks

The new methods described here are both efficient and economical according to these considerations. This can lead to significant gains in efficiency in three dimensions (two orders of magnitude over classical Godunov methods and three orders over WENO methods). Such improvements suggest that one might reasonably undertake some calculations that would be prohibitively expensive by other methods. Alternatively, these new methods can make calculations more economical to pursue given a fixed level of solution quality.

Upon examining the literature we find that these methods are not entirely new. The first example of this approach is found in Huynh's uniformly second-order Godunov method where MUSCL and UNO are combined [17]. Later, Suresh and Huynh applied similar principles to edge-centered variables in [33]; with these methods they coined the term "accuracy- and monotonicity-preserving" methods (AMP). The same principles have been employed to one-step time integration algorithms by Daru and Tenaud [10]. The AMP approach was employed by Balsara and Shu for very high-order WENO methods in [1] as MPWENO. Also, Shu has applied WENO schemes as a limiter for discontinuous Galerkin methods [25]. More recently, the approach by Hill and Pullin of combining WENO methods selectively with centered differencing is similar to these methods [16]. Finally, Tang and Baeder [34] provide design principles for a piece-wise parabolic method with enhanced accuracy for use in vorticity-dominated flows.

The important concept is to employ the most appropriate differencing stencil for the local character of the flow, using monotonicity as the fundamental trigger for a more careful consideration of differencing options. In particular we have directly addressed a standard shortcoming of Godunov methods noted in [12] and elsewhere that extrema in solutions are clipped, the new methods do not have this character. In closing, we have presented a methodology that can employ different fundamental differencing concepts with relative ease without a dramatic increase in cost.

Acknowledgements

All the authors are supported by the US DOE's ASC initiative performed under the auspices of the Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36. We thank Robert Lowrie, Mark Taylor, Ed Dendy and Len Margolin for their assistance and comments while performing this work.

References

[1] D. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, Journal of Computational Physics 160 (2) (2000) 405–452.

[2] J.P. Boris, D.L. Book, Flux-corrected transport I. SHASTA, a fluid transport algorithm that works, Journal of Computational Physics 11 (1973) 38–69.

[3] M.E. Brachet, D.I. Meiron, S.A. Orszag, B.G. Nickel, R.H. Morf, U. Frisch, Small-scale structure of the Taylor–Green vortex, Journal of Fluid Mechanics 130 (1983) 411.

[4] J. Casper, M.H. Carpenter, Computational considerations for the simulation of shock-induced sound, SIAM Journal of Scientific Computing 19 (3) (1998) 813–828.

[5] P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, SIAM Journal on Scientific and Statistical Computing 6 (1985) 104–117.

[6] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, Journal of Computational Physics 87 (1990) 171–200.

[7] P. Colella, H.M. Glaz, R.E. Ferguson, Multifluid algorithms for Eulerian finite difference methods, in preparation.

[8] P. Colella, P. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, Journal of Computational Physics 54 (1984) 174–201.

[9] A.W. Cook, W.H. Cabot, A high-wavenumber viscosity for high-resolution numerical methods, Journal of Computational Physics 195 (2) (2004) 594–601.

[10] V. Daru, C. Tenaud, High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations, Journal of Computational Physics 193 (2) (2004) 563–594.

[11] R. Donat, S. Osher, Propagation of error in regions of smoothness for nonlinear approximations to hyperbolic equations, Computer Methods in Applied Mechanics and Engineering 80 (1990) 59–64.

[12] J. Greenough, W.J. Rider, A quantitative comparison of numerical methods for the compressible Euler equations. Fifth-order WENO and piecewise-linear Godunov, Journal of Computational Physics 196 (1) (2004) 259–281 (Los Alamos Unlimited Release Report, LA-UR-02-5640).

[13] A. Harten, High resolution schemes for hyperbolic conservation laws, Journal of Computational Physics 49 (1983) 357–393.

[14] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, Journal of Computational Physics 71 (1987) 231–303.

[15] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, Journal of Computational Physics 207 (2) (2005) 542–567.

[16] D.J. Hill, D.I. Pullin, Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shock, Journal of Computational Physics 194 (2) (2004) 435–450 (CalTech ASCI Technical Report 170).

[17] H.T. Huynh, Accurate upwind methods for the Euler equations, SIAM Journal on Numerical Analysis 32 (1995) 1565–1619.

[18] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.

[19] C. Laney, Computational Gasdynamics, Cambridge, 1998.

[20] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge, 2002.

[21] A. Majda, S. Osher, Propagation of error into regions of smoothness for accurate difference approximations to hyperbolic equations, Communications on Pure and Applied Mathematics 30 (1977) 671–705.

[22] G.H. Miller, P. Colella, A high-order Eulerian Godunov method for elastic–plastic flow in solids, Journal of Computational Physics 167 (1) (2001) 137–176.

[23] G.H. Miller, P. Colella, A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing, Journal of Computational Physics 183 (1) (2002) 26–82.

[24] G.H. Miller, E.G. Puckett, A high-order Godunov method for multiple condensed phases, Journal of Computational Physics 128 (1996) 134–164.

[25] J.X. Qui, C.W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, SIAM Journal on Scientific Computing 26 (3) (2005) 907–929.

[26] J.J. Quirk, A contribution to the great Riemann solver debate, International Journal for Numerical Methods in Fluids 18 (1992) 555–574.

[27] W.J. Rider, An adaptive Riemann solver using a two-shock approximation, Computers and Fluids 28 (1999) 741–777.

[28] R. Sanders, A. Weiser, High resolution staggered mesh approach for nonlinear hyperbolic systems of conservation laws, Journal of Computational Physics 101 (2) (1992) 314–329.

[29] C.W. Shu, W.S. Don, D. Gottlieb, O. Schilling, L. Jameson, Numerical convergence study of nearly incompressible, inviscid Taylor–Green vortex flow, Journal of Scientific Computing 24 (1) (2005) 569–595.

[30] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, Journal of Computational Physics 83 (1989) 32–78.

[31] G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, Journal of Computational Physics 27 (1978) 1–31.

[32] G. Strang, On the construction and comparison of difference schemes, SIAM Journal on Numerical Analysis 5 (1968) 506–517.

[33] A. Suresh, H.T. Huynh, Accurate monotonicity-preserving schemes with Runge–Kutta time stepping, Journal of Computational Physics 136 (1997) 83–99.

[34] L. Tang, J.D. Baeder, Improving Godunov-type reconstruction for simulation of vortex-dominated flows, Journal of Computational Physics 213 (2) (2006) 659–675.

[35] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, Journal of Computational Physics 32 (1979) 101–136.

[36] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.

[37] P.R. Woodward, Piecewise-parabolic methods for astrophysical fluid dynamics, in: K.-H.A Winkler, M.L. Norman (Eds.), Astrophysical Radiation Hydrodynamics, 1986, pp. 245–326.

[38] P.R. Woodward, Implicit Large Eddy Simulation: Computing Turbulent Fluid Flow, Cambridge University Press, 2007 (Chapter 4).