

A generalized wavelet based grid-adaptive and scale-selective implementation of WENO schemes for conservation laws

Romit Maulik · Ratikanta Behera ·
Omer San

Received: date / Accepted: date

Abstract We devise a framework for the scale-selective application of a fifth-order WENO scheme for the simulation of hyperbolic conservation laws exhibiting strong discontinuities. A generalized wavelet based multiresolution (MR) procedure, embedded in a conservative finite volume formulation, is used for a dynamic grid adaptation of the solution field. Taking advantage of the structure detection properties of this MR algorithm, the nonlinear weights of the conventional WENO implementation are selectively modified to ensure lower dissipation in smoother areas. This modification is implemented through a linear transition from the fifth-order upwind stencil at the coarsest regions of the adaptive grid to a fully nonlinear fifth-order WENO scheme at areas of high irregularity. Our computing algorithm consists of a dynamic grid adaptation strategy, a scale-selective state reconstruction, a conservative flux calculation and a total variation diminishing Runge-Kutta scheme for time advancement. Results are presented for canonical examples drawn from the inviscid Burgers, shallow water, Euler and magnetohydrodynamic equations. Our findings represent a novel direction for the addition of scale-selective dissipation without a compromise on shock capturing behavior for conservation laws, which are particularly promising for implicit large eddy simulation type approaches.

R. Maulik · O. San
School of Mechanical & Aerospace Engineering,
Oklahoma State University, Stillwater, Oklahoma-74078, USA.
E-mail: romitm@ostateemail.okstate.edu

O. San
E-mail: osan@okstate.edu

R. Behera
Department of Mathematics and Statistics,
Indian Institute of Science Education and Research Kolkata,
Nadia, West Bengal, INDIA 741246.
E-mail: ratikanta@iiserkol.ac.in

Keywords Multiresolution Analysis · WENO · Scale-selective dissipation · Generalized Wavelets

Mathematics Subject Classification (2000) 65MXX · 65D05 · 42C40 · 35L65

1 Introduction

A peculiarity of conservation laws with nonlinearities is the possibility of discontinuous solutions. Multiple examples of such discontinuities are observed in diverse fields such as gasdynamics [54, 55], nonlinear optics [19] and geophysical applications [2, 39]. These pose considerable challenges to computational and mathematical frameworks due to the requirement of very high degrees of freedom as well as the corresponding increase in computational time and storage requirements. This localized high frequency behavior thus dictates the resolution requirements of a solution field which may otherwise be rather smooth on average. The high degree of localization makes the use of spectral methods inefficient since global expansion fails to represent the solution accurately. At the same time, the method of finite differences or volumes, while suitably compact, necessitates a large number of grid points due to their inherently local nature. While improved hardware and data-oriented strategies such as parallel computing have made great strides in the past few decades, storage costs and computational expense severely limit uniform grid simulations of complex physics exhibiting sharp gradients. Multiresolution techniques bridge the gap between both spectral and finite difference/volume approaches by providing a framework for the identification of high frequency regions in the solution field through its hybrid behavior, i.e. being localized in both space and scale [4, 13, 53]. Traditionally, mesh refinements have usually been constructed according to the presence of sharp gradients in the field [8, 9] or local residuals [47]. However, a significant limitation to these approaches is the lack of reliable error control. Discrete wavelet based techniques are also commonly used for mesh-refinement purposes [23, 25, 26, 30, 34, 41, 52] and may be considered a similar approach to the framework proposed here.

Therefore the general body of wavelet transform algorithms with their inherent ability of scale localization are a natural tool for application to problems such as the detection of localized structures and active error-control[46]. Furthermore, these algorithms can be classified in different ways depending on whether they take full or partial advantage of wavelet analysis. In this regard, we may divide wavelet methods into three main classes: pure wavelet, multiresolution, and wavelet optimized methods. Pure wavelet schemes (either Galerkin or collocation) directly use wavelet decomposition to discretize the PDEs [41, 52]. Multiresolution methods were inspired by wavelets but are based on finite-difference, finite-volume, or finite-element formulation with multiresolution and automated refinement capabilities [23, 25, 26]. Finally, wavelet-optimized methods include algorithms based on classical discretizations (e.g.,

by finite differences or finite volumes) but use wavelet analysis either to define adaptive meshes or to speed up linear algebra [34].

In this investigation, we follow the methodology given in [11,18,44] by utilizing a multiresolution procedure inspired by the principles of data compression, local refinement and inexpensive flux computations to adapt our grid according to the localization of frequency content. For the purpose of our frequency detection, we utilize the generalized wavelet concepts proposed Harten in [23,25,26] and further used in [10,24] for multiresolution computations for hyperbolic problems. Indeed, the same concept of generalized wavelets has been used in [44] for grid adaptivity as well. We must remark here that the technique we propose differs from the use of the second-generation interpolated subdivision and average interpolation techniques for the fast wavelet transform [49,51,52]. Our technique utilizes a box scaling function along with a collocated volume approach towards grid adaptivity and inherently preserves cell-averages through consistent projection (a key property of the second-generation wavelets used with the lifting update[48]). At their heart however, both techniques use some form of solution comparison at different levels of resolution to calculate a measure of the local regularity. While the interpolated wavelet based approach utilizes successive high and low-pass filtering at collocated grid points to determine local wavelet coefficients (which are indicators of smoothness), the presented technique utilizes interpolation at different cell volume levels to determine local ‘details’ which aid in identifying areas of coarsening or refinement. In essence, we are attempting to ascertain the local frequency resolution ability of a certain level of resolution. We utilize the finite volume approach on our ‘adapted grid’. Once a grid has been adapted to the prevalent solution at a particular instance in time, discrete approximations are made for the calculation of the gradients in the field. Interfacial state reconstructions for the conserved variables within the finite volume framework are obtained using the classical fifth-order WENO schemes [7]. This ensures accurate capture of discontinuities in the solution field. The WENO framework may also be assumed to add an implicit dissipation to the solution field which may be assumed to mimic subgrid stresses in a classical turbulence perspective [21]. This procedure deviates from the established approach of adding an ‘artificial viscosity’ term to the hyperbolic equations to deal with spurious oscillations due to misrepresented Fourier modes [41].

A limitation of WENO schemes arises from the fact that these are too dissipative for the accurate modeling of turbulent flow [40]. This has spurred a number of investigations for the purpose of overcoming their drawbacks for the detailed simulation of compressible turbulence. The concept of coupling linear upwind (or central) stencils in smooth regions and shock capturing WENO (or ENO) schemes is quite popular in that it implies lesser dissipation in smoother areas while shock capturing ability is not sacrificed [1,42,28]. We follow the weighted average methodology of ‘switching’ between the linear upwind and fifth-order WENO schemes as given in [43,33] where a weighting function is utilized for the gradual conversion of a less dissipative stencil (incapable of shock capture) to a more dissipative (and shock capturing) stencil in the vicinity of

high or undefined gradients. Therefore, our investigation focuses on an adaptive methodology used to obtain these approximations such that an implicit dissipation may be magnified in the vicinity of high frequency content while smoother areas remain relatively untouched. This ‘scale-selective dissipation’ is conceptually appropriate for the next level of adaptive implicit large eddy simulation approaches for turbulence modeling. Our scale-selective sensor is based on a linear weighting of the smoothness indicators from the smoothest to the coarsest areas of the adapted grid. The proposed scheme leverages information about the local regularity of the solution obtained through multiresolution analysis and may be considered to be a novel weighting function for dynamically changing the dissipative nature of WENO state reconstruction approaches. Indeed, a similar investigation has been described in [37] where second-generation lifted wavelets are utilized in the method of finite differences to identify areas of high solution irregularity and to use fifth-order WENO solely at the highest levels of refinement while the inverse wavelet transform is used to calculate stencil for finite difference approximations of gradients recursively at different levels of resolution [52]. We note that several works in literature [15, 24, 25] have also suggested the utilization of central flux reconstructions in regions of relatively low irregularity. However the scale-selective dissipation we introduce to the time integration of the hyperbolic systems in this investigation differs in that central fluxes are eschewed in favor of the modification of nonlinear weights alone. This shall become clearer in future sections.

For the purpose of obtaining a compressed representation of our data and for the calculation of wavelet coefficients of the solution field, we utilize a fifth-order spatial interpolation scheme which utilizes a six point stencil [24, 44]. Our computing algorithm is implemented in an objected oriented framework with dynamic access to heap memory through an unordered-map environment in C++. A binary-tree data structure is implemented in conjunction with the proposed refinement strategy to dynamically locate the presence of each finite volume cell in the adapted grid. Further algorithmic and data-structure details are provided within. For the validation of this proposed methodology, we test our procedure on the time evolution of the inviscid Burgers’ equation with a moving shock. In addition, we test our framework on Sod’s shock tube problem which is a generalization of a gasdynamics application [35]. Similar hyperbolic systems of equations that are tested include the Brio-Wu shock tube problem utilizing the magnetohydrodynamic (MHD) equations [12] and the classical dambreak problem on a wet bottom described by the shallow water equations [3]. Our framework is thus tested for a variety of discontinuities and physical systems. The selective dissipation framework is tested on its ability to detect regions of high frequency and adapt grids accordingly. In addition, it is expected to considerably reduce the degrees of freedom for the computation of a high quality simulation without the formation of spurious numerical noise. While this represents a primary investigation into the feasibility of the proposed framework for discontinuity capturing, investigations are underway to

utilize this scale-selective dissipation mechanism for its utility as an adaptive ILES framework.

A summary of the main facets of this investigation may be given by the following bullet points

- A generalized wavelet based finite volume method is implemented for the simulation of hyperbolic partial differential equations.
- The aforementioned method is augmented by utilizing a scale-selective approach to the calculation of interfacial fluxes. Smoother regions are adapted to show more linear tendencies in WENO smoothness indicators for state reconstructions.
- The proposed mechanism for solution evolution is validated for a variety of hyperbolic systems exhibiting strong discontinuities.
- Assessments are made about the feasibility of the proposed approach for degree of freedom reduction, grid adaptation as well as numerical noise suppression.
- A comprehensive analysis of the memory and computational benefits of the proposed algorithm are also presented.

In what follows of this document, we detail our numerical algorithm for time integration as well as the proposed framework for conservative flux reconstruction at cell interfaces in Section 2. Section 3 introduces the multiresolution framework utilized for adapting our finite volume grid for one-dimensional hyperbolic systems including detailed information about algorithmic implementations. Section 4 contains results from several numerical experiments utilized to validate the proposed framework. Concluding remarks are made in section 5.

2 Integration Algorithm

2.1 Conservation Laws

As mentioned in the previous section, we shall utilize a one-dimensional hyperbolic system of equations in their conservative form as our underlying partial differential equations. These can be expressed as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad \mathbf{q}, \mathbf{F} \in \mathbb{R}^n, \quad (1)$$

where \mathbf{q} stands for an arbitrary set of conserved variables and $\mathbf{F} = f(\mathbf{q})$ represents their corresponding fluxes. Our algorithm is formulated around a general configuration given by this framework and different physical systems may thus be simulated through it. Note that the one-dimensional framework presented here may easily be extrapolated to multidimensional implementations.

2.2 Finite Volume Framework

The semi-discrete form of the governing equations can be written as

$$\frac{dq_i}{dt} + \frac{1}{\Delta x}(F_{i+1/2,j} - F_{i-1/2,j}) = 0, \quad (2)$$

with q_i being the cell-averaged vector of dependant variables, $F_{i\pm 1/2,j}$ representing the fluxes at the right and left cell boundaries. We use the method of lines to represent our system of PDE's as an ODE through time (so as to implement a Runge-Kutta scheme for time integration)

$$\frac{dq_i}{dt} = \mathcal{L}(q_i), \quad (3)$$

with the right hand side of the above equation representing the combined effect of all the spatial derivatives in the governing equations. The above ODE representation can be advanced to obtain the solution field at a future time step $n + 1$ given the current time step n by a total variation diminishing third-order Runge Kutta scheme [20] as follows

$$\begin{aligned} q_i^{(1)} &= q_i^{(n)} + \Delta t \mathcal{L}(q_i^{(n)}), \\ q_i^{(2)} &= \frac{3}{4}q_i^{(n)} + \frac{1}{4}q_i^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(q_i^{(1)}), \\ q_i^{(n+1)} &= \frac{1}{3}q_i^{(n)} + \frac{2}{3}q_i^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(q_i^{(2)}), \end{aligned} \quad (4)$$

where a time step Δt is specified explicitly according to the physical problem being solved (more details are presented in the discussion related to the Riemann solver for interfacial flux reconstruction). We utilize a CFL=0.5 criterion for the computation of the dynamically adapted time step.

2.3 State Reconstruction

The left and right states of the cell boundaries are reconstructed using the weighted essentially non-oscillatory (WENO) approach which was first introduced in [38]. The order of accuracy of these reconstructions depends on the length of the stencil chosen and affect the solution (through a different dissipative behavior). In this work, we shall focus on the fifth-order accurate WENO scheme (WENO-5) which utilizes a seven point stencil for state reconstructions at both cell faces. We note, for the purpose of clarity, that it is the conserved variables which are being reconstructed in this formulation. A modified implementation of the WENO-5 reconstruction [31] can be given by

$$q_{i+1/2}^{\pm} = \frac{w_0}{6}(2q_{i\mp 2} - 7q_{i\mp 1} + 11q_i) + \frac{w_1}{6}(q_{i\mp 1} + 5q_i + 2q_{i\pm 1}) + \frac{w_2}{6}(2q_i + 5q_{i\pm 1} - q_{i\pm 2}) \quad (5)$$

where $q_{i+1/2}^+$ refers to the interfacial state reconstruction assuming a wave moving from the left direction and $q_{i+1/2}^-$ corresponds to a wave moving from the right direction. In our reconstruction equation the nonlinear weights are defined by

$$w_k = \frac{\alpha_k}{\alpha_0 + \alpha_1 + \alpha_2}, \quad \alpha_k = \frac{d_k}{(\beta_k + \epsilon)^p}, \quad (6)$$

with the smoothness indicators defined as

$$\beta_0 = \frac{13}{12}(q_{i-2} - 2q_{i-1} + q_i)^2 + \frac{1}{4}(q_{i-2} - 4q_{i-1} + 3q_i)^2, \quad (7)$$

$$\beta_1 = \frac{13}{12}(q_{i-1} - 2q_i + q_{i+1})^2 + \frac{1}{4}(q_{i-1} - q_{i+1})^2, \quad (8)$$

$$\beta_2 = \frac{13}{12}(q_i - 2q_{i+1} + q_{i+2})^2 + \frac{1}{4}(3q_i - 4q_{i+1} + q_{i+2})^2. \quad (9)$$

In this study, we compute the smoothness indicators for each conserved variable. The optimal linear weighting coefficients are $d_0 = 1/10$, $d_1 = 3/5$ and $d_2 = 3/10$ in Eq. (5). The presence of any discontinuity thus leads to an adaptation in the weights for order reduction and increased dissipation in the stencil corresponding to the discontinuity. This expression for the calculation of the nonlinear weights of the smoothness indicator for the WENO-5 interfacial flux reconstruction is also referred to as the WENO-JS scheme. Before proceeding, we must observe that there are many variants of the WENO reconstructions described above (for example in [27, 22, 32, 29]) and this investigation is by no means exhaustive. The different variants of the WENO reconstructions have all been developed as a response to demands of either different orders of accuracy [7, 6] or reduced computational expense. The interested reader is directed to [36] for an excellent discussion on the effect of the order of reconstruction on a classical two-dimensional configuration utilizing the Euler equations. As is common with the WENO-JS approach, the value of ϵ is set to 10^{-6} and p is fixed at 2.

2.4 Scale Selective WENO Algorithm

We propose a novel approach to the application of WENO-5 in the adaptive mesh refinement framework by utilizing a resolution level dependent ‘linearization’ of the nonlinear weights. The WENO approach provides smoothness indicators which decompose to their higher order linear upwind schemes in the presence of a smooth field. However, our investigations showed that the smoothness indicators never truly evolve into the linear coefficients for a smooth field. While this behavior is beneficial for shock capturing, it proves to add excess dissipation in smooth regions which may be detrimental in implicit large eddy simulation methods for turbulence. Consequently we devise

the following scheme for the modification of the nonlinear weights. Given the least level of resolution at an instant in time in a grid adapted field l_{min} and a predefined maximum level of resolution L , we define

$$\theta_w = \frac{l - l_{min}}{L - l_{min}}, \quad (10)$$

and

$$\tilde{w}_i = (1 - \theta_w)d_i + \theta_w w_i, \quad (11)$$

where d_i and w_i are WENO coefficient parameters defined previously. We may now utilize \tilde{w}_i in our WENO-5 formulation (in place of the traditional nonlinear weights given by w_i) for a resolution based nonlinear weighting of stencils for upwinding. Fig. (1) graphically displays the scale-selective nature of our framework.

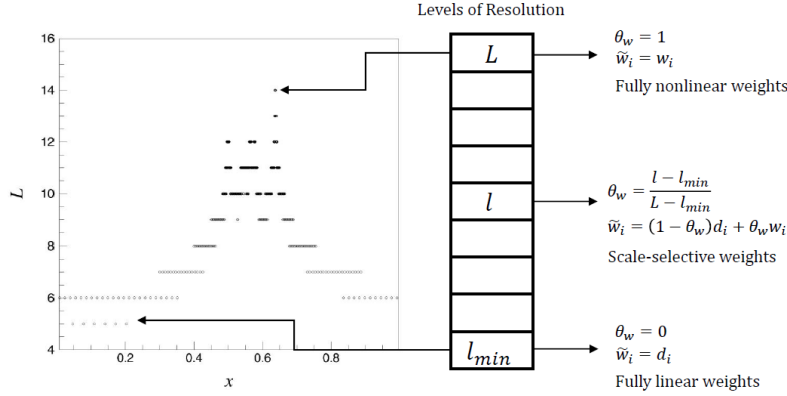


Fig. 1 A schematic of the scale-selective adaptation of the nonlinear weights in the proposed framework. Notice how smoother areas are dissipated using linear fifth-order upwind stencils.

2.5 Riemann Solver

Once the left and right states have been constructed using WENO-5, we may now use Riemann solvers to calculate the fluxes at these boundaries. The Rusanov Riemann solver [45] utilizes information from the maximum local wave propagation speed to give us the follow expression for the flux

$$F_{i+1/2} = \frac{1}{2}(F^+ + F^-) + \frac{c_{i+1/2}}{2} (q_{i+1/2}^+ - q_{i+1/2}^-), \quad (12)$$

where F^+ and F^- are the flux components using the right and left constructed states respectively (i.e. $F^+ = F(q_{i+1/2}^+)$ and $F^- = F(q_{i+1/2}^-)$). The local wave propagation speed $c_{i+1/2}$, is given by the maximum absolute eigenvalue of the Jacobian matrix of F between the cells involved in our interfacial state reconstructions i.e.

$$c_{i+1/2} = \max\left(r(A_{i-2}), r(A_{i-1}), r(A_i), r(A_{i+1}), r(A_{i+2}), r(A_{i+3})\right), \quad (13)$$

for the WENO-5 stencil where A_i stands for the Jacobian of the flux vector given by

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{q}}. \quad (14)$$

Here $r(A)$ represents the spectral radius of matrix A which depends on its eigenvalues. Thus each system of equations we shall examine will have a distinct approach towards the calculation of wavespeed. This shall be outlined in the section detailing numerical experiments.

3 Generalized Wavelet Based Grid Adaptation

3.1 Generalized Wavelets: An Overview

Our adaptive grid framework utilizes the concept of generalized orthonormal wavelets with compact supports which arises from discrete multiresolution analysis [23]. In this section we present an overview of the theory of generalized wavelets and how they are suitable for multiresolution analyses. Given a set of discrete values of a uniform grid

$$u_{i=1}^{N_L}, \text{ such that } N_L = 2^L, \quad (15)$$

which represent discrete samples of a function (assumed to be periodic for simplicity) within a domain $x \in [0, 1]$ i.e.

$$x_i^L = i \cdot h_L, \quad 0 \leq i \leq N_L, \quad h_L = 1/N_L, \quad (16)$$

$$u_i = \langle u, \phi_i \rangle = \int u(x) \phi_i(x) dx, \quad \text{where } \phi_i(x) = \frac{1}{h_L} \phi\left(\frac{x - x_i^L}{h_L}\right), \quad (17)$$

and $\langle \cdot, \cdot \rangle$ represents an inner product. The scaling function $\phi(x)$,

$$\int \phi(x) dx = 1, \quad (18)$$

is assumed to be of compact support and to satisfy the dilation equation

$$\phi(x) = 2 \sum_k a_k \phi(2x - l). \quad (19)$$

Similarly, we may design a set of nested grids in increasing order of cell sizes as

$$\left\{ \{x_i^l\}_{i=1}^{N_l} \right\}_{l=L-1}^0, \quad x_i^l = i \cdot h_l, \quad h_l = 2^l h_L, \quad N_l = \frac{1}{h_l}, \quad (20)$$

and corresponding quantities

$$u_i^l = \langle u, \phi_i^l \rangle, \quad \phi_i^l(x) = \frac{1}{h_l} \phi\left(\frac{x - x_i^l}{h_l}\right), \quad 1 \leq i \leq N_l, \quad 0 \leq l \leq L, \quad (21)$$

where u stands for a continuous underlying function from which we obtain discrete samples to get a discrete multiresolution representation of our uniform field given by

$$\left\{ \{u_i^l\}_{i=1}^{N_l} \right\}_{l=0}^L, \quad (22)$$

provided the relationship given by Eq.(19) holds. The interpretation of the above relation is that the knowledge of discrete values at level $l+1$ may be utilized to determine the knowledge of discrete values at level l or in other words that the values at a level l represent all larger scales from $l = k+1$ down to $l = 0$ provided a suitable set of coefficients a_k are chosen i.e.

$$\left\langle u, \phi_i^l - \sum_k a_k \phi_{2i+1}^{l+1} \right\rangle = 0. \quad (23)$$

A conservative reconstruction procedure $R(x; \bar{u}^l)$ may then be devised to obtain continuous approximations of discrete values at each level of resolution. The conservative property of this reconstruction satisfies

$$\left\langle R(x; u^l), \phi_i^l(x) \right\rangle = u_i^l, \quad 1 \leq i \leq N_l. \quad (24)$$

A continuous approximation of the errors in scale capture between two conservative levels may be obtained through

$$Q^l(x; u) = R(x; u^{l+1}) - R(x; u^l), \quad (25)$$

and its Euclidean inner-product with the scaling function gives us

$$d_i^{l+1} = \left\langle Q^l(x; u), \phi_i^{l+1}(x) \right\rangle. \quad (26)$$

If R is an orthogonal projection into the linear span of $\phi_i^l(x)$ i.e.

$$R(x; u^l) = \sum_{i=1}^{N_l} u_i^l \phi_i^l(x), \quad (27)$$

then our reconstruction procedure basically constitutes the transformation of the discrete samples to the compactly supported orthonormal wavelet bases of Daubechies [16] and d_i^l may be assumed to be the *discrete wavelet coefficients*

at level of resolution l . Indeed, the choice of a Dirac-delta scaling function derives the first-generation interpolated wavelet transform [53] but with the caveat that variable averages are not conserved as wavelet-coefficients are cascaded from the finest to the coarsest grids (a property intrinsically satisfied when using the box scaling function). To avoid this aliasing error, Sweldens[48] suggests an *update* procedure for the subsampled values at each level of resolution called the *lifted* wavelet transform [52]. In our investigation, we utilize a box scaling function that is integrated into the finite volume framework. As given in [23] this choice satisfies

$$\phi(x) = \phi(2x) + \phi(2x + 1) \rightarrow a_0 = a_{-1} = \frac{1}{2}. \quad (28)$$

to give us cell-averaged samples of our conserved variables i.e.

$$\bar{u}_i^l = \langle u, \phi_i^l \rangle. \quad (29)$$

The reader is directed to [23] and references therein for a discussion about the choice of the scaling function and implications for multiresolution analysis.

3.2 Multiresolution Analysis

As mentioned in the introduction, in this section we utilize a computational domain constructed through nested cell volumes. Utilizing the terminology of Harten [24] and Roussel et al. [44], let $G^l, 0 \leq l \leq L$, be a set of nested dyadic grids

$$G^l = \{x_i^l\}_{i=0}^{N_l}, \quad h_l = 2^l, \quad N_l = 2^l, \quad l \in [0, L], \quad (30)$$

which corresponds to a nested set of finite volumes on a one-dimensional computational domain $x \in (0, 1)$ with cell-size at level of resolution l given by h_l . The coarsest level of resolution corresponds to $l = 0$ which implies that the entire computational domain is approximated by one cell-average N_0 located at $x = 0.5$. We may further define the sequences of cell-averages (assuming a scalar conservation equation)

$$\bar{u}_i^l = \frac{1}{h_l} \int_{x_{i-1}^l}^{x_i^l} u(x) dx. \quad (31)$$

A cornerstone of Harten's multiresolution analysis was the utilization of a *projection* and *prediction* operation for determining local frequency content. The projection operator is utilized to estimate the cell-averages at a level l given the cell-averages at level $l + 1$. This operation is important for the preservation of the conservative property of the adaptive grid algorithm. The projection operator ($P_{l+1 \rightarrow l}$) for a one-dimensional structured dyadic grid may be represented as

$$\bar{u}_i^l = (P_{l+1 \rightarrow l} \bar{U}^{l+1})_i = \frac{1}{2} (\bar{u}_{2i}^{l+1} + \bar{u}_{2i+1}^{l+1}). \quad (32)$$

Here \bar{U}^{l+1} stands for the ensemble of ‘children’ finite volume cells of \bar{u}_i^l given by the left child \bar{u}_{2i}^{l+1} and right child cell \bar{u}_{2i+1}^{l+1} (in one dimension). In combination with the projection operator, we also utilize a prediction operator for the calculation of interpolation errors as we cascade down the cells from the coarsest to the finest volumes. In other words the prediction operator $P_{l \rightarrow l+1}$ is utilized to approximate \bar{U}^l at the level $l+1$ by interpolation. We utilize a fifth-order accurate polynomial interpolation given by [24]

$$\begin{aligned}\hat{u}_{2i}^{l+1} &= \bar{u}_i^l + \sum_{m=1}^2 \gamma_m (\bar{u}_{i+m}^l - \bar{u}_{i-m}^l), \\ \hat{u}_{2i+1}^{l+1} &= \bar{u}_i^l - \sum_{m=1}^2 \gamma_m (\bar{u}_{i+m}^l - \bar{u}_{i-m}^l).\end{aligned}\tag{33}$$

Our coefficients in the chosen interpolation scheme are given by

$$\gamma_1 = -\frac{22}{128}, \quad \gamma_2 = \frac{3}{128}.\tag{34}$$

The primary purpose of the prediction operator is to calculate *details* at each cell-center not belonging to the lowest level of resolution in the algorithm. We may thus calculate

$$\bar{d}_i^l = \bar{u}_i^l - \hat{u}_i^l.\tag{35}$$

We must remark here that in the one-dimensional case, only one of the details belonging to the two children of each cell are independent. This is in order to respect the consistency relations which preserve cell-averages. We must remark here that the choice of the interpolation stencil and the detail calculation strategy should be *consistent* with the projection operator. In other words the approximate value \hat{U}_{l+1} must be conservative with respect to the average values on the coarser grid \bar{U}^l :

$$|V_i^l| \bar{u}_i^l = \sum_{p=2i}^{2i+1} |V_p^{l+1}| \hat{u}_p^{l+1},\tag{36}$$

where p is summed for all the children cells of \bar{u}_i^l at the resolution of $l+1$ and $|V_i^l|$ stands for the volume of a cell at location i within a level of resolution L . Extensions to multiple dimensions and for vector conservation equations are straightforward and we refer the interested reader to [44, 10, 11] for further discussions. The projection and prediction operations are summarized by pseudocode in Algorithms 1 and 2 respectively.

Algorithm 1 Projection Operations

```

1: procedure PROJECTION(Tree) ▷ For calculating cell averages of parents
2:   Given solution at locally refined leaves
3:    $l = L - 1$  ▷ Starting at finer resolution
4:   while  $l \neq 0$  do
5:     Project to find cell average of parents ▷ Using Eq.32
6:      $l = l - 1$  ▷ Advance towards root
7:   end while
8: end procedure

```

Algorithm 2 Prediction Operations

```

1: procedure PREDICTION(Tree) ▷ For calculating details
2:   Use projected values to predict details  $\bar{d}_i^l$ 
3:    $l = L - 1$  ▷ Starting at finer resolution
4:   while  $l \neq 0$  do
5:     Interpolate to find  $\bar{d}_i^l$  ▷ Using Eq.33
6:      $l = l - 1$  ▷ Advance towards root
7:   end while
8: end procedure

```

3.3 Graded Implementation of Computational Domain

In order to implement our proposed framework in a dynamic memory framework, we utilize a binary-tree type data-structure which allows for easy algorithm implementation, book-keeping and speed of navigation. The binary-tree also corresponds to the dynamic-graded tree framework that is utilized for a multiresolution analysis utilizing generalized wavelets. We follow in the footsteps of [15, 44, 11] to identify some terminology related to our multiresolution framework

- The root is starting position of the tree data structure. As shall be demonstrated later, our one-dimensional framework specifies one root situated at the center of our computational domain. Note that this is not a necessary condition as one may have ‘forest’ type data structures with multiple roots.
- Each finite volume cell (if existing) is denoted as a node.
- Each node, if a parent, has 2^D children nodes, where D is the number of dimensions of the computational domain. In our investigation, each parent node thus has two children nodes.
- If one child node of a parent is required to exist from our multiresolution analysis, all the children of this parent should also be required to exist.
- A node is called a *leaf* if it possesses no children.
- At any given instant in the time integration procedure, each leaf must have the required number of neighbors in each direction for the calculation of cell face fluxes using WENO state reconstructions. If these neighbors are not present, they must be created and denoted as *virtual* leaves.
- Any creation of a node should be graded. In other words, the ancestry of the node in the graded structure should be respected. In addition to the ancestry, a *prediction* stencil should also be present for each new node. This

prediction stencil is used for interpolation across levels and is introduced in Eq.(33).

Through the enforcement of the aforementioned checks, one can guarantee that any node in the graded data structure always has its parents and the members of a prediction stencil associated with the calculation of its details. In addition, any leaf always has a certain pre-decided number of fixed neighbors. A question that arises here is for the coarsest levels of resolution, i.e., when there aren't enough neighbors to respect our conditions or enough nodes in the coarser layer to have a prediction stencil. To avoid this issue, our data structure is devised in such a way so as to leverage our boundary conditions (which for this investigation are all kept open). This discussion shall be revisited shortly after a description is provided of the class object used to contain information about each finite volume cell. A schematic of the collocated structure of our finite volume cells can be seen in Fig. (2). It can be seen that each cell in the domain possesses a unique integer identifier which may be utilized to locate it from the root of the graded tree (set at $i = 1$ for a one-dimensional case).

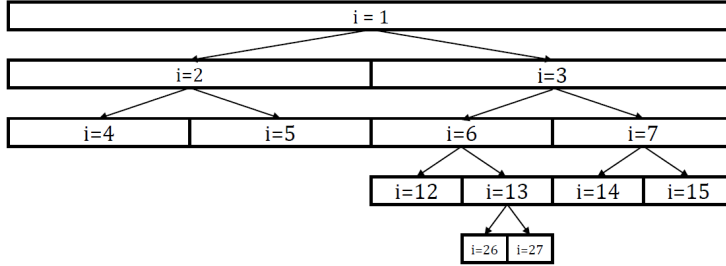


Fig. 2 A schematic of the collocated finite volume adaptive mesh refinement framework.

A finite volume cell class is implemented for the purpose of representing each unique cell in the graded tree. This class contains several flags such as those which indicate whether or not the cell is a leaf, whether it is virtual or if it is a newly created cell which needs to have its conserved variables interpolated from ancestry. In addition, we also include 8 bit precision floating point arrays for the storage of cell centered quantities and corresponding fluxes at the cell face. Several pointers are also added for maintaining linkages with neighbors at a particular level of resolution as well as with the cell's parent and its multiple children. Integer quantities corresponding to the level of resolution as well as the unique spatial index corresponding to each cell are also stored for the purpose of identification. Finally, we also provide for an array of *details* (the calculation of which has been described in Section 3.2) which quantify the magnitude of the local frequency content of the cell. A complete description of our class object is provided in listing 1 in the Appendix. Fig. (3) shows a schematic related to the number of pointers of each class object instantiated.

The orange pointers indicate connections within the same level of resolution and the black pointers indicate parent child relationships.

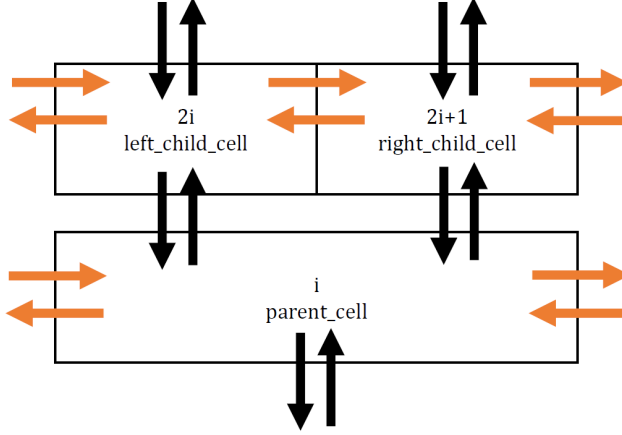


Fig. 3 A schematic of the various pointers in an instance of the class object. Black pointers are parent-child connections across different resolutions and the orange pointers are connected to the neighboring cells.

One can observe that pointers to neighbors need to be adapted to handle open boundary conditions. For this reason, boundary cells are modified to point to themselves so that the left neighbor of the left boundary cell is itself and the right neighbor of the right boundary cell is itself. Therefore any recursive requirement for points beyond the boundary are met through this criteria. We acknowledge that ghost cells may be utilized for higher order boundary condition implementation, but for the purpose of this investigation we finalize this rather simple approach. In addition, the problems we investigate here are all classical in that they exhibit the most interesting physical phenomena away from the boundary. Therefore we may safely assume that our results are not contaminated by these lower-order boundary conditions.

We also leverage the fact that each existing finite volume cell is uniquely identified through a spatial index i and its level of resolution and may thus be found through a unique path from the root of the graded-tree. We therefore implement a function `findcell` which given a particular cell value and access to the unordered map containing all the pointers to our cell class, returns a pointer to the cell class object introduced previously. Listing 2 in the appendix details our version of the implementation. We remark that any cell at resolution l may be found through $2l$ steps which points to $O(N \log N)$ complexity at the worst case. However, as mentioned by Roussel et al. [44], in general very few regions require the maximum amount of resolution and the proposed algorithm is quite efficient.

3.4 Predictive Thresholding

The projection and prediction operators are used in combination with a locally refined solution to determine which cells must be preserved for dynamic grid adaptivity. The prerequisite for the application of this thresholding process requires that the tree be graded at all times. A global threshold parameter ε_g is defined for the purpose of identifying regions with low regularity. Consequently, this threshold may be utilizing for determining areas of the computational domain which may be coarsened in comparison to those which require more points for adequate resolution. As demonstrated in [14], this threshold parameter is responsible for the control of the perturbation error introduced by the multiresolution procedure. Following Harten [25], we define a resolution dependent (or local) threshold given by

$$\varepsilon_l = 2^{l-L} \varepsilon_g, \quad (37)$$

where l stands for the local level of resolution of the cell and L is the maximum level of refinement allowed in the computational domain (as defined previously). Through this error, our perturbation error is bounded [14, 15, 44]. The pseudocode described in Algorithm 3 describes the thresholding procedure for our investigation for a one-dimensional case with one conserved variable. This thresholding procedure is ‘predictive’ in that it retains (or adds) new finite volume cells in order to accurately resolve advective physics.

Algorithm 3 Harten’s Predictive Thresholding

```

1: procedure THRESHOLD(Tree) ▷ For decision on cell deletion/retention
2:   Given  $\varepsilon_g$  evaluate local threshold  $\varepsilon_l = 2^{l-L} \varepsilon_g$  ▷ To bound perturbation error
3:    $l = L - 1$  ▷ Starting at one level below max resolution
4:   while  $l \neq 0$  do
5:     if  $d_i^l < \varepsilon_l$  then
6:       Delete Children nodes  $\bar{u}_{2i}^{l+1}$  and  $\bar{u}_{2i+1}^{l+1}$  ▷ Since this level of resolution is not
       required
8:     else
9:       Keep nodes  $\bar{u}_{2i+q}^{l+1}$  where  $-K \leq q \leq K + 1$  ▷  $K = 1$  for advective solution
       capture
10:      if  $d_i^l \geq 2^{2r} \varepsilon_l$  and  $l \neq L - 1$  then ▷  $r = 4$  hard limited by order of
       polynomial(5)
11:        Keep nodes given by  $\bar{u}_{2q}^{l+2}$  and  $\bar{u}_{2q+1}^{l+2}$  where  $2i - K \leq q \leq 2i + 1 + K$ 
12:      end if
13:    end if
14:     $l = l - 1$  ▷ Advance towards root
15:  end while
16: end procedure

```

In order to obtain a consistent thresholding approach for systems of equations we utilize the following measure of the local details for a cell at a partic-

ular level of resolution.

$$\bar{d}_i^l = \frac{|\bar{d}_i^l|_{L_1}}{\max_k |\bar{d}_k^l|_{L_1}}. \quad (38)$$

This implies that the L_1 norm of the details associated with each conserved variable are scaled by the maximum value of the L_1 norm of the details at cells of the same resolution.

3.5 Virtual Cells and Conservative Fluxes

We have previously introduced the concept of virtual cells which are necessary for the computation of fluxes on the leaves. This to ensure our state reconstructions may be carried out by accessing information through pointers to neighbors. The algorithm outlined in Algorithm 4 details the addition of virtual cells to the domain. Once virtual cells are added, fluxes may be calculated on each leaf in the domain. However, this gives rise to a problem with the conservation of fluxes. To remedy this, ingoing and outgoing fluxes are matched at leaves which have their face fluxes calculated from virtual cells. In one dimension, if a cell \bar{u}_i^l is a leaf but \bar{u}_{2i-2}^{l+1} and \bar{u}_{2i-1}^{l+1} exist, then the flux at the positive face of \bar{u}_{2i-1}^{l+1} is equated with the flux at the negative face of \bar{u}_i^l . Similarly, if \bar{u}_{2i+2}^{l+1} and \bar{u}_{2i+3}^{l+1} exist, then the flux at the negative face of \bar{u}_{2i+2}^{l+1} is equated to the flux at the positive face of \bar{u}_i^l . Conservativeness is thus ensured and these fluxes may be used in a discrete time integration. We must clarify that the time integration of the solution is carried out *only* on the ‘true’ leaves. The virtual leaves have been designed purely for the purpose of flux calculation. Once the true leaves are updated, the projection operator is used to cascade this update back through the coarser resolutions and the prediction operator is used solely to update the values in the virtual cell. A flow-chart detailing the entire dynamic adaptive grid methodology is shown in Fig. (4).

To summarize the retention of cells in the adaptive domain, we may draw up the following list of conditions for the retention of a cell. A cell must be retained

- If its details are than the local threshold of that particular resolution ε_l .
- If it is required to predict the details of a cell in a finer resolution (i.e. it belongs to a prediction stencil).
- If it is required for the construction of interfacial fluxes for any leaf cells (i.e. this cell is a virtually needed cell).
- If a parent has one child requires retention, all its children must be retained (i.e. conservative retention).

4 Numerical Experiments

In this section we outline several approaches to validating our proposed framework. Firstly, we utilize a function with a discontinuous field for assessing the

Algorithm 4 Virtual Leaf Addition

```

1: procedure VIRTUALADD(Leaf)                                ▷ For ensuring uniform flux stencils
2:   while  $l \neq L$  do
3:      $l = 0$                                                     ▷ Starting at coarsest resolution
4:     Given a particular leaf  $\bar{u}_i^l$ 
5:     if Neighbors  $\bar{u}_{i \pm n}^l$  don't exist then                ▷  $n$  to preserve flux stencil for current leaf
6:       Create virtual cells  $\bar{u}_{i \pm n}^l$ 
7:       Predict interpolated solution at virtual cells          ▷ Cells should be graded
8:     else
9:       if Neighbor  $\bar{u}_{i \pm 1}^l$  exists and is not a leaf then
10:        Create virtual cells  $\bar{u}_{i \pm m}^{l+1}$                 ▷  $m$  to preserve flux stencil for children of
neighbor
11:      end if
12:    end if
13:     $l = l + 1$                                                 ▷ Advance towards finer resolutions
14:  end while
15: end procedure

```

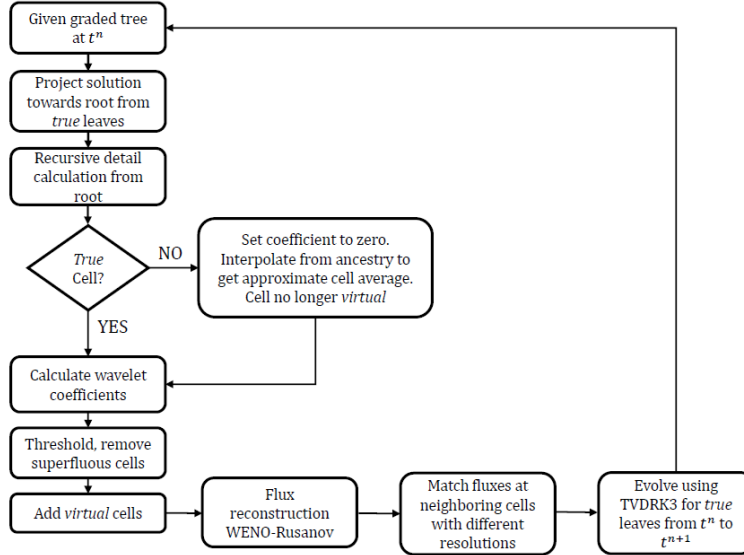


Fig. 4 A flow-chart of processes involved in the dynamic adaptation of the grid in our framework.

performance of our refinement strategy. Following this, we evaluate the proposed scale-selective flux reconstruction strategy on hyperbolic conservation laws. Attention is paid to the ability to reconstruct strong discontinuities as well as memory requirement reduction ratios. Computational gains in terms of CPU time saved are also studied for our selected physical systems.

4.1 Validation: A-priori Testing for a Static Function

We select a function given by

$$f(x) = \sin(2\pi(x - 0.5)) + e^{-10000(x-0.5)^2}, \quad (39)$$

for evaluating the grid adaptation ability of the generalized wavelet based approach. As shown in Fig. (5), the choice of a global thresholding parameter ε_g effectively controls the clustering of cell volumes in the vicinity of the sharp discontinuity at $x = 0.5$. We must note here that although the discontinuity is severe, it is not ‘undefined’ as in the case of a shocked solution. For this reason, the highest level chosen for our assessment here ($L = 14$), is not reached. We paraphrase Harten [25] who notes that the generalized wavelet coefficients \bar{d}_i^l diminish in size at finer grids according to the local regularity of the function but remain unchanged related to level of refinement in the presence of an undefined gradient. This fact is witnessed in action for our experiments with conservation laws where standing or moving shocks always exhibit the finest level of cell clustering irrespective of the global threshold criterion. A point to note with regards to the result from the static function demonstration here is that only *true leaves* are shown in Fig. (5). These correspond to the adapted grid on which flux calculations would be carried out for the purpose of time evolution. One must note however, that there exists a graded binary tree that connects the shown leaves to the single root for our computational domain. Also, virtual leaves required for flux computations are omitted from the shown images as well. We shall follow this approach for the rest of this document.

The process of refinement we follow is through the provision of the exact value of the function at a uniform grid corresponding to the finest level following which a thresholding strategy is employed (as described in the grid adaptation section). Once an adapted grid is obtained, we utilize the exact expression for the function to calculate the L_1 norm of the error between the exact and approximate values of the field at each finite volume cell. A challenge in this calculation is that the different number of grid points may possibly lead to an increasing L_1 norm for higher values of the global threshold ε_g . This was observed for the time evolution of the Euler equation in [15]. We therefore propose the utilization of a *normalized* L_1 norm which simply calculates the average L_1 norm across the total number of points at a particular instance in the domain. One can observe a consistent reduction in the L_1 norm with increasing strictness in the global threshold as shown in Fig. (6). In order to quantify the ‘order’ of accuracy of our mesh refinement procedure we make use of a projection to a uniform grid through an interpolation strategy similar to the ones described in [17, 50]. Our adapted grid solution is projected to a uniform grid at the finest level of resolution and the L_1 norm of the error between the exact and adapted solution indicates the accuracy of our refinement (also known as the perturbation error). Fig. (7) also shows that the perturbation error recovers a linear fit with ε_g as expected and validates our mesh refinement strategy. For the rest of this document we utilize the simple normalized L_1

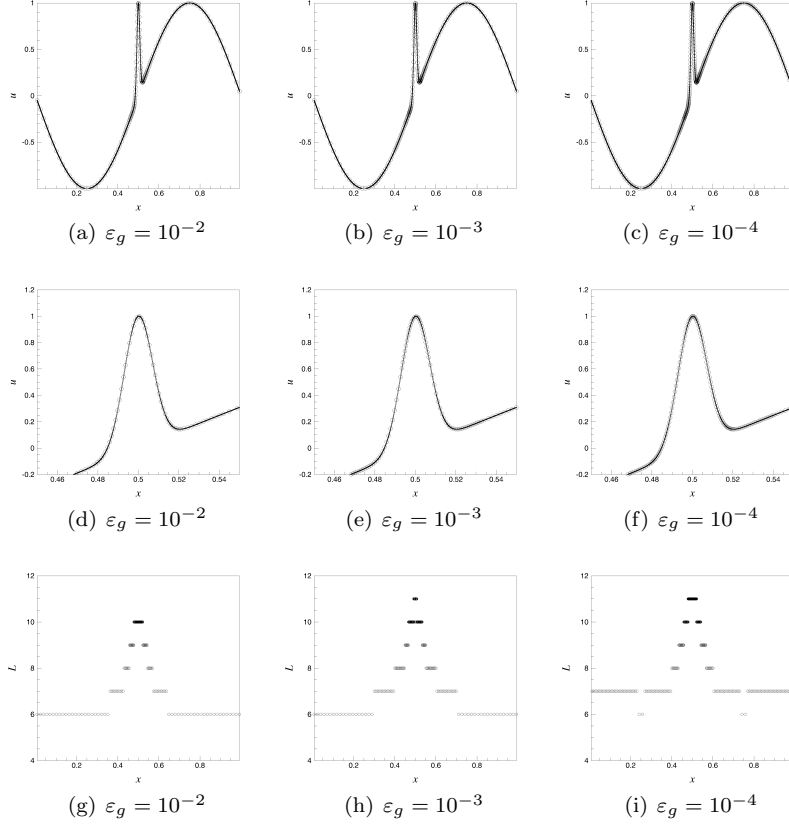


Fig. 5 A visual representation of increasing refinement due to stricter thresholding. Fields for test function shown above, zoom-in of sharp gradient at $x = 0.5$ shown in the middle and corresponding levels of resolution shown below. The solid line is given by the exact solution.

norm of the adaptive grid (i.e. without projection) for a relative quantification of benefits.

Another advantage of grid-refinement techniques is the feature of data-compression. We define a relative memory allocation (RMA) metric based on the total number of finite volume cells present in an adapted version of the grid (N_{AMR}) divided by those used in a uniform grid at the finest level L (N_{UG}) i.e.

$$\text{RMA} = \frac{N_{AMR}}{N_{UG}} \times 100. \quad (40)$$

A point of clarification here is that while the size of our adapted grid shall take into account the ancestry of all adapted grid points all the way till the root of the domain, the reference uniform grid will assume a simple array type datastructure with the only the finest level of elements present in memory. The

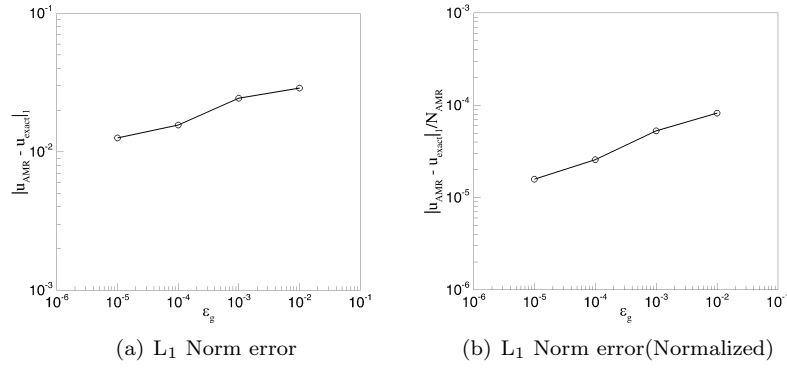


Fig. 6 Varying L1 Norm errors (left) and L1 norm errors normalized by the total number of points in the domain (middle), for the static function as a function of ε_g . On the right we also have the linear fit of the thresholding criteria ε_g with the reduction in perturbation error.

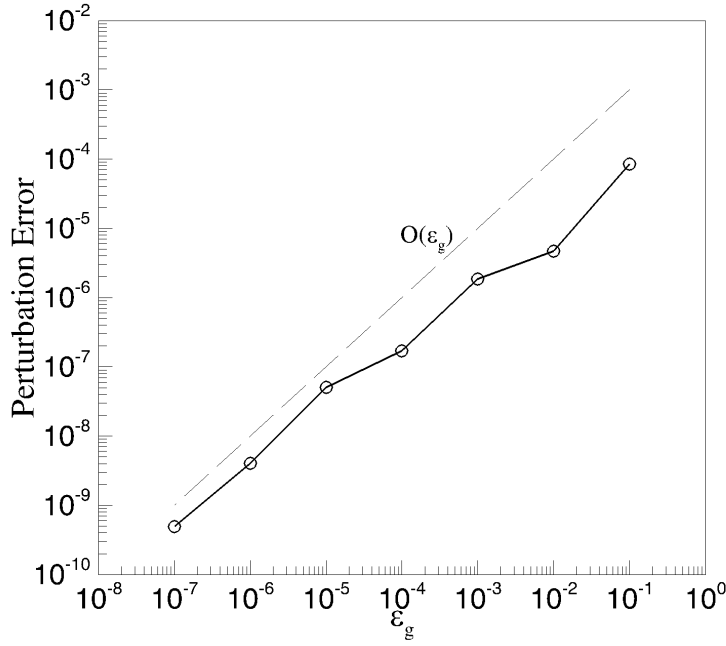


Fig. 7 Perturbation error (with order of accuracy) for the mesh refinement strategy utilized in this investigation. One can observe a linear fit with ε_g .

implication for this is that memory requirements may be greater than 100% for certain selections of thresholds or choices of maximum levels of resolution

despite an adapted grid. To clarify this particular metric, let us describe a sample case: an RMA of 20% for an AMR simulation with maximum level $L = 12$ implies that

$$\begin{aligned} N_{UG} &= 2^{12} = 2048, \\ N_{AMR} &= \frac{\text{RMA}}{100} 2048 \approx 410. \end{aligned} \quad (41)$$

For our assessments with the static function we obtain trends which agree with [44] that higher thresholds combined with lower maximum levels of resolution tend to result in high relative memory allocation as shown in Fig. (8) and the global threshold parameter $\varepsilon_g \rightarrow 0$ indicates that the grid is uniform. This is also observed for our dynamic experiments with the Burgers, Euler and MHD conservation laws.

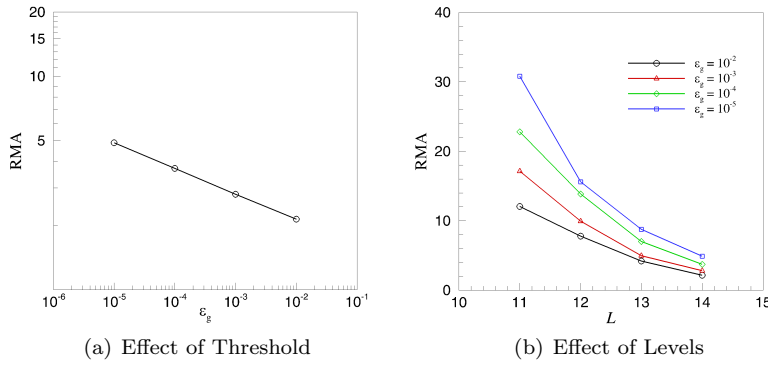


Fig. 8 Memory requirements with different ε_g and levels of resolution. Observe how high thresholds and low levels lead to infeasible memory gains.

4.2 Nonlinear Scalar Conservation Laws

Our first *dynamic* test case for the proposed framework is given by the hyperbolic inviscid Burgers equation given by the following initial and boundary conditions [5],

$$u_t + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0, \quad x \in \mathbb{R}, \quad t \geq 0, \quad (42)$$

with initial conditions given by

$$u_0(x) = \begin{cases} 1, & \text{for } 0.5 \leq x < 0.6, \\ 0, & \text{otherwise.} \end{cases} \quad (43)$$

From the point of view of our proposed framework for hyperbolic laws presented in Eq. (1), we have

$$q = u \text{ and } F = \frac{u^2}{2}. \quad (44)$$

A wavespeed estimate (for utilization in the Rusanov Riemann solver) is obtained from the following expression in accordance with Eq. (13 and 14),

$$c_{i+1/2} = \max \left(|q_{i-2}|, |q_{i-1}|, |q_i|, |q_{i+1}|, |q_{i+2}|, |q_{i+3}| \right). \quad (45)$$

The above described system of equations is solved for open boundary conditions in both positive and negative x directions till a non-dimensional time of $t = 0.075$ where we perform our numerical assessments. The challenge in simulating this configuration of the Burgers' problem is that it exhibits a moving discontinuity with time. Fig. (9) details the performance of the scale-selective WENO-5 flux reconstruction methodology in combination with the adaptive grid framework. It can be readily observed that the framework clusters cells around the vicinity of regions with sharp gradients. Our exact solution is computed using $N_{UG} = 65,536$ finite volume cells and a good approximation of this result is obtained. It is also seen that stricter thresholds cluster points in larger quantities near the discontinuities. Zoom-ins of the shock front also show negligible dissipation in terms of the capture of sharp features.

We also detail a comparison of different memory requirements as a function of L and the global refinement threshold ε_g at $t = 0.075$ which corresponds to the final point of our time evolution. The trends observed for the case of the static function are seen once more at the final time and remain consistent through the evolution of the Burgers' equation as well. This can be seen in Fig. (10) which details the time evolution of memory requirement ratios as a function of L and ε_g . As mentioned in [44], to control the growth of the perturbation error in time, the strict thresholding of $\varepsilon_g = 10^{-5}$ continues to add points to the computational domain in comparison with $\varepsilon_g = 10^{-3}$ which has a more relaxed bound for this perturbation error. It is thus clear that an indiscriminate selection of the global thresholding and level will significantly affect the data compression and computational gain of the algorithm. We also detail computational gains in terms of percentage speed up in Table 1. The following metric is defined for quantifying this gain in terms of cpu time:

$$\text{Gain} = \left(1 - \frac{t_{AMR}}{t_{UG}} \right) \times 100, \quad (46)$$

where t_{AMR} stands for the time taken to undertake an AMR simulation using the proposed framework and a designated threshold and maximum level of resolution. On the other hand t_{UG} stands for the time taken on a uniform grid using the finite volume approach at the same finest level of resolution corresponding to L in the AMR method. Therefore, positive values of this metric indicate a percentage decrease in computational cost whereas negative values indicate loss. It is predicted that for very tight thresholds and low L

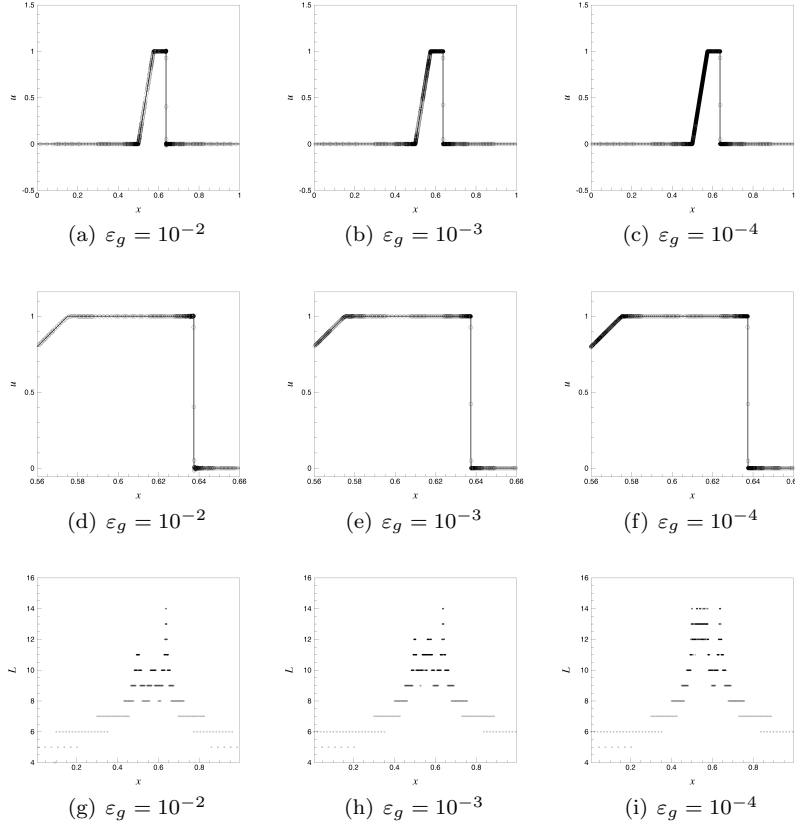


Fig. 9 A visual representation of increasing refinement due to stricter thresholding. Fields for the inviscid Burgers' equation problem at $t = 0.075$ shown above, zoom-ins of the moving shock front showing discontinuity resolution performance in the middle and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65,536 degrees of freedom.

the cost of the searching the binary tree would dominate overall computational expense and make the proposed framework infeasible from this point of view. The values tabulated indicate the percentage of time saved for the solution of the system to a designated final time at a particular level. Reference uniform grid simulations are carried out using the same binary-tree data structure at the designated level of resolution but without the process of recursive sampling to calculate wavelet coefficients. One can observe that the general trend shows an increase in gain as the value of L increases. We note that the somewhat inconsistent nature of the trends are due to the fact that simulations at lower L are extremely short (order of 10 seconds for $L = 11$) and the unequal load distribution on the shared HPC cluster may cause such slight discrepancies. We must remark here that the benefits we witness for the one-dimensional

CPU Gain	$\varepsilon_g = 10^{-2}$	$\varepsilon_g = 10^{-3}$	$\varepsilon_g = 10^{-4}$	$\varepsilon_g = 10^{-5}$
$L = 11$	0	-50	-42.86	-50
$L = 12$	-15.25	11.86	-23.73	-27.12
$L = 13$	39.00	21.16	9.12	-3.32
$L = 14$	47.26	51.34	40.29	45.57

Table 1 Computational gain in percentages for our choice of combinations for the Inviscid Burgers' equation. Higher resolutions and lower ε_g can be seen to provide greater benefits.

problems we study here shall be amplified considerably when the proposed algorithms are extended to higher dimensions.

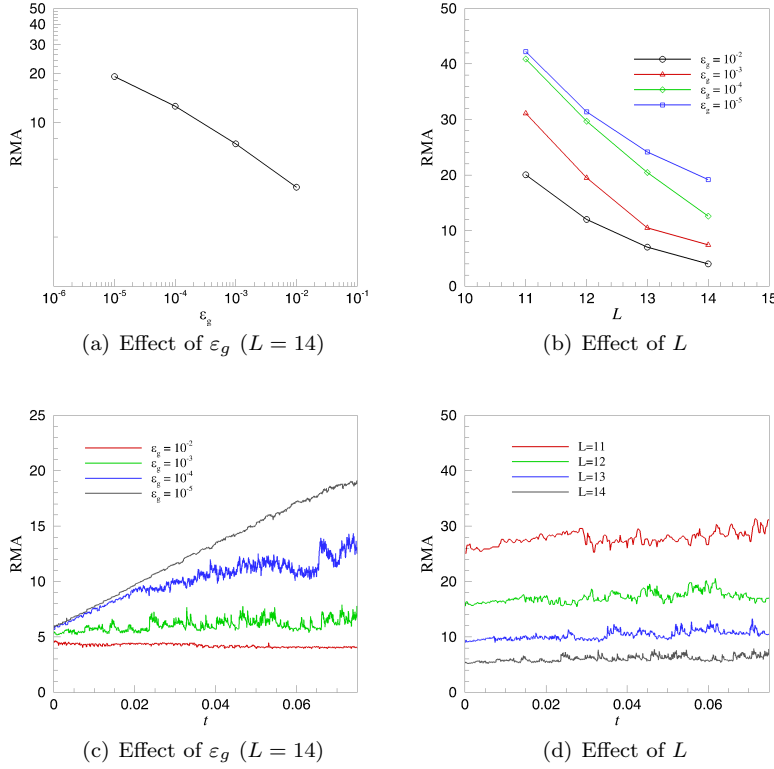


Fig. 10 The effect of the choice of L and ε_g on the memory gain of the system with time. Higher relative memory allocations for lower L and stricter ε_g observed as expected.

4.3 Shallow Water Equations: Dam Break Problem

Our next set of numerical tests utilize a modified form of the dam break problem given in [3]. The governing laws of the problem are given by the Shallow-Water (or Saint Venant) equations which are given in their conservation form by

$$\mathbf{q} = \begin{bmatrix} \rho h \\ \rho h u \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho h u \\ \rho h u^2 + \frac{1}{2} \rho g h^2 \end{bmatrix}. \quad (47)$$

where ρ is the fluid density, h is the height of the fluid column in the computational domain and u is the horizontal velocity of the fluid and g is the acceleration due to gravity. For the purpose of this investigation we set our $\rho = 1$ and $g = 1$. Wavespeed estimates at a particular cell for the Riemann solver are given through the calculation of spectral radii given by

$$r(A) = \max(|u|, |u - a|, |u + a|), \quad (48)$$

where the speed of propagation of information (analogous to the speed of sound) is given by

$$a = 2\sqrt{gh}. \quad (49)$$

Our interfacial wavespeed may then be reconstructed using Eq. (13). The initial conditions for this problem are given by

$$h_0(x) = \begin{cases} 10, & \text{for } 0.0 \leq x \leq 0.5 \\ 1, & \text{for } 0.5 < x \leq 1.0. \end{cases} \text{ and } u_0(x) = 0. \quad (50)$$

As in the Burgers' equation, a reference solution computed on a uniform grid comprised of $N_{UG} = 65,636$ finite volume cells is utilized for the purpose of numerical assessment. Fig. (11) shows the performance of our framework on the shallow water equations. It can be seen that cell volumes are bifurcated at regions of discontinuities such as the standing shock (also called the bore) and decreasing values of ε_g tend to higher amounts of clustering at areas with high gradients. We also perform a comparison of memory gains in Fig. (12) where once again one can observe greater memory reduction benefits at higher values of L and lower ε_g . Lower thresholds combined with lower choices of L may even lead to negative memory gains (i.e. $\text{RMA} > 100\%$) in which case the benefits of AMR are lost. For the shallow water equations tested here, we observed that a value of $L = 14$ and $10^{-2} \leq \varepsilon_g \leq 10^{-3}$ were feasible for excellent memory gains as well as CPU benefits as shown in Table 2. Once again the table details how a low selection of L and ε_g could lead to a slower computation as compared to the fine grid. For our results here we obtained a 40% benefit in terms of computational expense at $L = 14$ and $\varepsilon_g = 10^{-2}$. The trends would suggest that increasing L beyond 14 would lead to further benefits in memory compression and computational expense reduction as well.

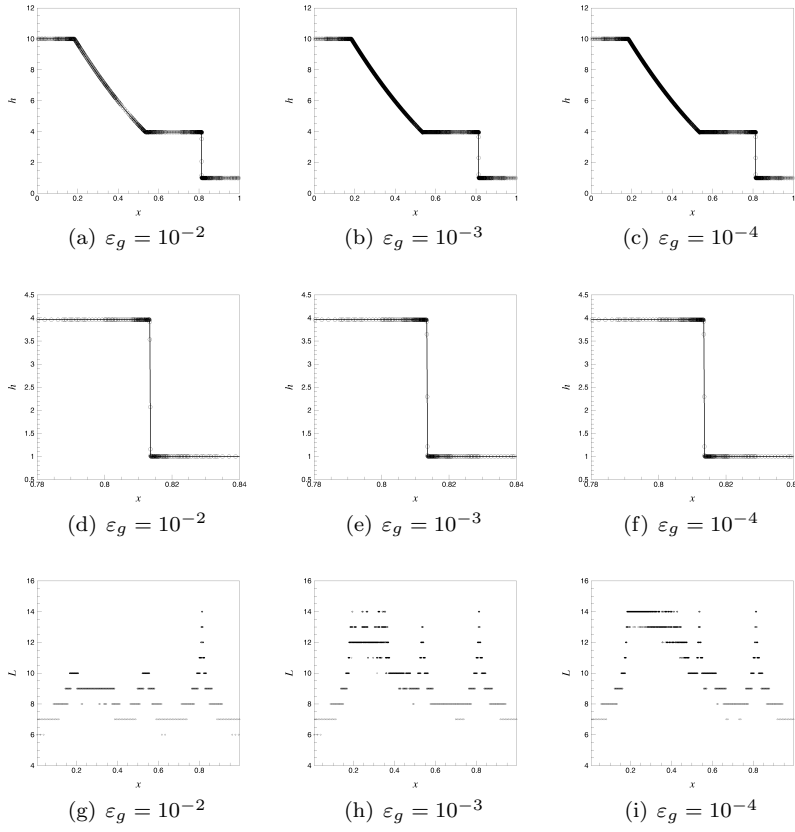


Fig. 11 A visual representation of increasing refinement due to stricter thresholding. Fields for the dambreak problem at $t = 0.1$ shown above, zoom-ins of the moving shock front showing discontinuity resolution performance in the middle and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65,536 degrees of freedom.

4.4 Gas Dynamics: Sod Shock Tube

A dynamic characterization of our deviation from analytical solutions may be carried out using the one-dimensional Euler equations configured to solve Sod's prototype shock tube problem for higher dimensional gas-dynamics applications. In terms of our hyperbolic framework we have

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}, \quad (51)$$

with an equation of state given by

$$p = (\gamma - 1)(E - \frac{1}{2}\rho u^2). \quad (52)$$

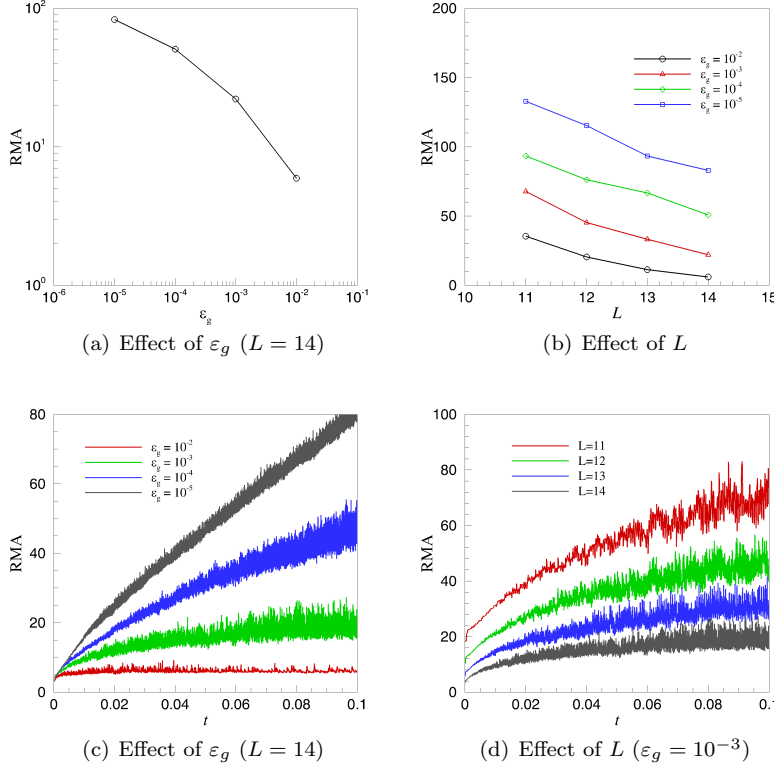


Fig. 12 The effect of the choice of L and ε_g on the relative memory allocation of the shallow water equations with time. Higher relative memory allocations for lower L and stricter ε_g observed as expected.

CPU Gain	$\varepsilon_g = 10^{-2}$	$\varepsilon_g = 10^{-3}$	$\varepsilon_g = 10^{-4}$	$\varepsilon_g = 10^{-5}$
$L = 11$	-9.80	-60.78	-88.24	-121.57
$L = 12$	17.16	-26.47	-74.02	-119.12
$L = 13$	27.55	-2.83	-28.29	-81.06
$L = 14$	39.60	20.43	-14.87	-44.45

Table 2 Computational gain in percentages for our choice of combinations for the shallow water equations. Higher resolutions can be seen to provide greater benefits and lower thresholds.

For our investigation we fix $\gamma = 7/5$ and the solution is evolved over the time interval of $t = 0$ to $t = 0.2$. Our initial conditions are given by

$$(\rho, u, p)_0 = \begin{cases} (1.0, 0.0, 1.0), & \text{for } 0.0 \leq x \leq 0.5 \\ (0.125, 0.0, 0.1), & \text{for } 0.5 < x \leq 1.0. \end{cases} \quad (53)$$

CPU Gain	$\varepsilon_g = 10^{-2}$	$\varepsilon_g = 10^{-3}$	$\varepsilon_g = 10^{-4}$	$\varepsilon_g = 10^{-5}$
$L = 11$	4.41	-22.06	-39.71	-42.65
$L = 12$	29.82	-4.00	-21.45	-33.82
$L = 13$	42.57	22.42	5.38	-13.67
$L = 14$	54.06	41.82	23.23	13.94

Table 3 Computational gain in percentages for our choice of combinations for the 1D Euler equations. Higher resolutions and lower thresholds can be seen to provide greater benefits.

The spectral radius of the aforementioned system can be obtained through

$$r(A) = \max(|u|, |u - a|, |u + a|) \quad (54)$$

where a is the speed of sound given by

$$a = \sqrt{\gamma p / \rho}. \quad (55)$$

Fig. (13) visually details the performance of the proposed framework for different thresholds at $t = 0.2$. A zoom-in of the shock front is also shown to describe the minimal dissipation of the framework. It can be seen that the $\varepsilon_g = 10^{-2}$ case proves to be slightly inaccurate as it allows for the build up of Gibbs' oscillations although numerical overflow is avoided due to the threshold. The case of $\varepsilon_g = 10^{-3}$ performs well in that no large scale oscillations are observed. Fig. (14) also details the memory gains from the implementation of the proposed framework. From the examination of the time evolution of the memory gains, it is apparent that ε_g values below 10^{-3} cause a lack of compressive gain due to excessive addition of the cells at comparatively lower levels of resolution. From our experiments, we discern that a combination of $L = 14$ and $\varepsilon_g = 10^{-3}$ gives us the maximum memory benefit. This is also reinforced in terms of computational benefit as shown in Table 3 where a 42% benefit is obtained at the same combination of parameters. An increase in the maximum level would improve these benefits significantly as well.

The presence of an analytical solution to this configuration of the one-dimensional shocktube problem lets us analyse errors of computation in the proposed framework through an evolution in time. Fig. (14) shows the L_1 norm and normalized L_1 norm of the error between analytical and exact solutions at $t = 0.2$ in the numerical simulation. One can observe a steadily decreasing trend for both quantities. However, there is a slight increase in the (un-normalized) L_1 norm due to the fact that a small error in a much larger total number of points prevents further reduction of this metric. The normalized version however, still shows a consistent decreasing trend quantifying to some extent the improved numerical accuracy of the proposed framework with stricter thresholds. We remind the reader that the scale-selective implementation of the WENO-5 state reconstruction stencil is not affected in terms of shock capturing ability and captures severe discontinuities comfortably (as seen here and previously).

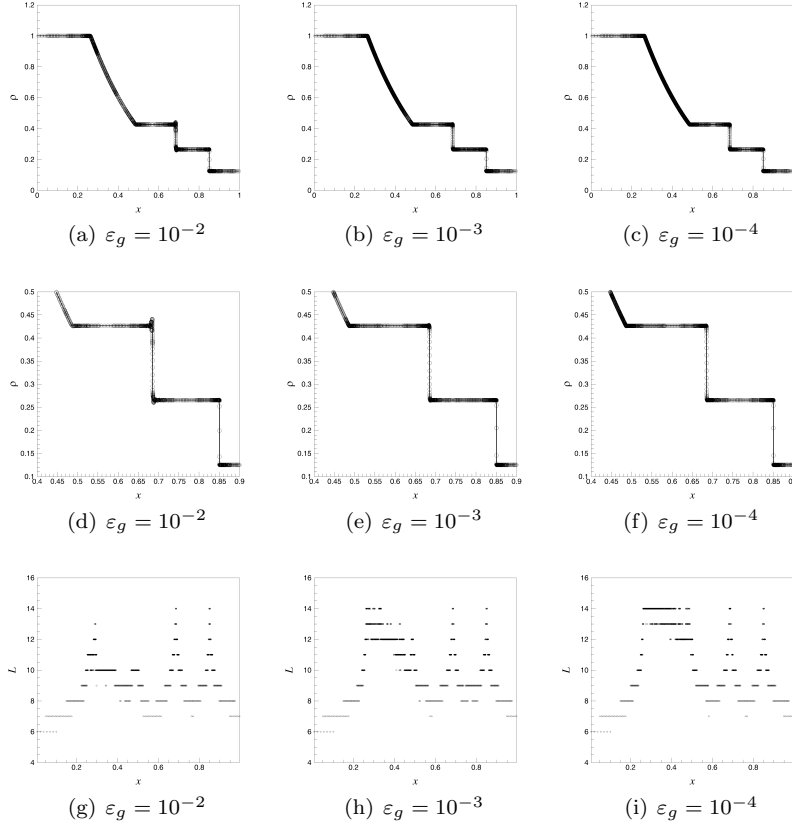


Fig. 13 A visual representation of increasing refinement due to stricter thresholding. Fields for the shock tube problem at $t = 0.2$ shown above, zoom-ins of the shock front showing discontinuity resolution performance in the middle and corresponding levels of resolution shown below. The solid line is given by the exact solution.

4.5 Magnetohydrodynamics: Brio-Wu Shock Tube

We conclude our numerical investigations by attempting to solve the 7 equation system of the MHD equations. The problem configuration is given by the famous Brio-Wu shock tube problem [12] which can be implemented in our framework as

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ B_y \\ B_z \\ E \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p^* - B_x^2 \\ \rho uv - B_x B_y \\ \rho uw - B_x B_z \\ B_y u - B_x v \\ B_z u - B_x w \\ u(E + p^*) - B_x(B_x u + B_y v + B_z w) \end{bmatrix}. \quad (56)$$

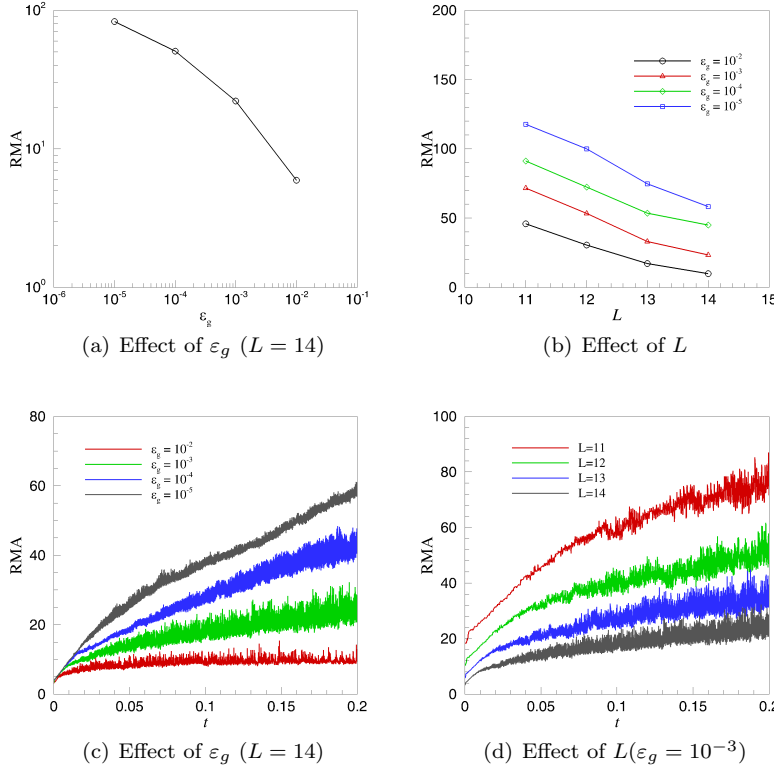


Fig. 14 The effect of the choice of L and ε_g on the memory gain of the 1D Euler equations with time. Higher relative memory allocations for lower L and stricter ε_g observed as expected.

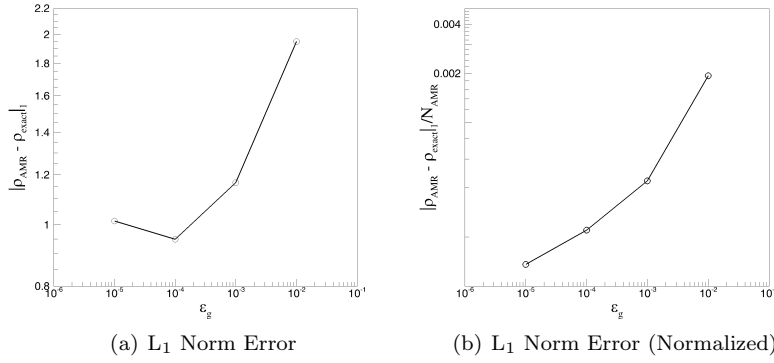


Fig. 15 Varying L_1 Norm error (left) and L_1 norm errors normalized by the total number of points in the computational domain at $t = 0.2$ for the Sod shock tube problem.

where

$$p^* = p + \frac{1}{2} (B_x^2 + B_y^2 + B_z^2) \quad (57)$$

is the total pressure and the pressure p is related to the conserved quantities through the equation of state

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2 + w^2) - \frac{1}{2} (B_x^2 + B_y^2 + B_z^2) \right). \quad (58)$$

Here we have presented cases for $\gamma = 7/5$ (a slight modification from the $\gamma = 2$ recommended in the original work but with no change to the conclusions). The solution is calculated from $t = 0.0$ to $t = 0.2$ with the initial conditions given by

$$(\rho, u, v, w, B_y, B_z, p)_0 = \begin{cases} (1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0), & \text{for } 0.0 \leq x \leq 0.5, \\ (0.125, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0), & \text{for } 0.5 < x \leq 1.0, \end{cases} \quad (59)$$

and $B_x = 0.75$ fixed as a constant value. The spectral radius of the aforementioned system can be obtained through

$$r(A) = \max \left(|u|, |u - c_f|, |u - c_a|, |u - c_s|, |u + c_f|, |u + c_a|, |u + c_s| \right), \quad (60)$$

where

$$\begin{aligned} c_f &= \sqrt{\frac{1}{2} \left[a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} + \sqrt{\left(a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} \right) - 4a^2 \frac{B_x^2}{\rho}} \right]}, \\ c_s &= \sqrt{\frac{1}{2} \left[a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} - \sqrt{\left(a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} \right) - 4a^2 \frac{B_x^2}{\rho}} \right]}, \\ c_a &= \sqrt{\frac{B_x^2}{\rho}}, \end{aligned} \quad (61)$$

and the corresponding speed of sound given by

$$a = \sqrt{\gamma p / \rho}, \quad (62)$$

Fig. (16) shows the development of the initial condition after 0.2 seconds have elapsed and its corresponding clustering of points. The presence of a significant number of sharp discontinuities leads to several areas of localized irregularity and corresponding clusters of cells. At $\varepsilon_g = 10^{-2}$ some oscillations are observed (similar to the Sod shock tube) and $\varepsilon_g = 10^{-3}$ provides us a good approximation of the high fidelity solution here (once again, computed with $N_{UG} = 65,536$ finite volume cells on a uniform grid) with good memory gain

and computational benefits (at $L = 14$). Fig. (17) is used to detail the evolution of the relative memory allocation of the system with time integration and it can immediately be seen that memory losses are incurred when very strict thresholds and low L are chosen. Indeed, the quantum of loss suggests that the presence of a large number of equations (with discontinuities at different locations of the computational domain) reduce the benefit of this AMR approach considerably. However, once again (as seen in Table 4) at $\varepsilon_g = 10^{-3}$ and $L = 14$, we can see an decrease of about 36% with respect to computational time and relative memory allocation ratios hovering predominantly around the 40% mark. Further increasing L would also lead to more benefits. The evidence here leads us to tentatively conclude that the true benefit of an AMR methodology is very large scale separation at localized areas of a solution field.

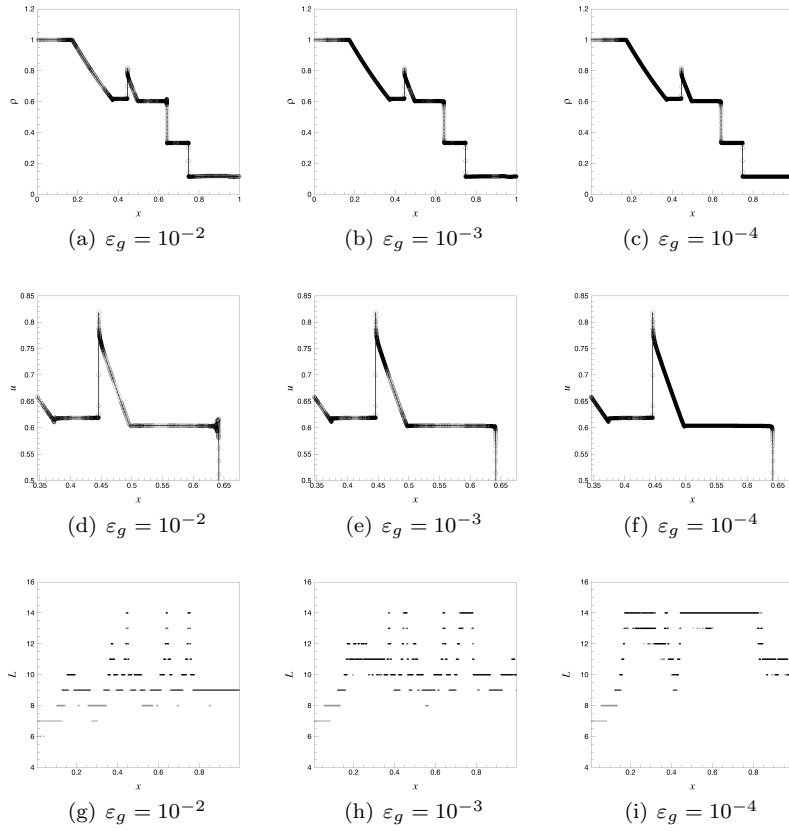


Fig. 16 A visual representation of increasing refinement due to stricter thresholding. Fields for the Brio-Wu shock tube problem at $t = 0.2$ shown above, zoom-ins of the shock front showing discontinuity resolution performance in the middle and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65,536 degrees of freedom.

CPU Gain	$\varepsilon_g = 10^{-2}$	$\varepsilon_g = 10^{-3}$	$\varepsilon_g = 10^{-4}$	$\varepsilon_g = 10^{-5}$
$L = 11$	5.53	-50.75	-69.85	-57.29
$L = 12$	32.82	-50.76	-78.49	-63.87
$L = 13$	51.96	-5.57	-63.49	-58.26
$L = 14$	61.12	36.02	-25.80	-48.39

Table 4 Computational gain in percentages for our choice of combinations for the 1D MHD equations. Higher resolutions and lower thresholds can be seen to provide greater benefits.

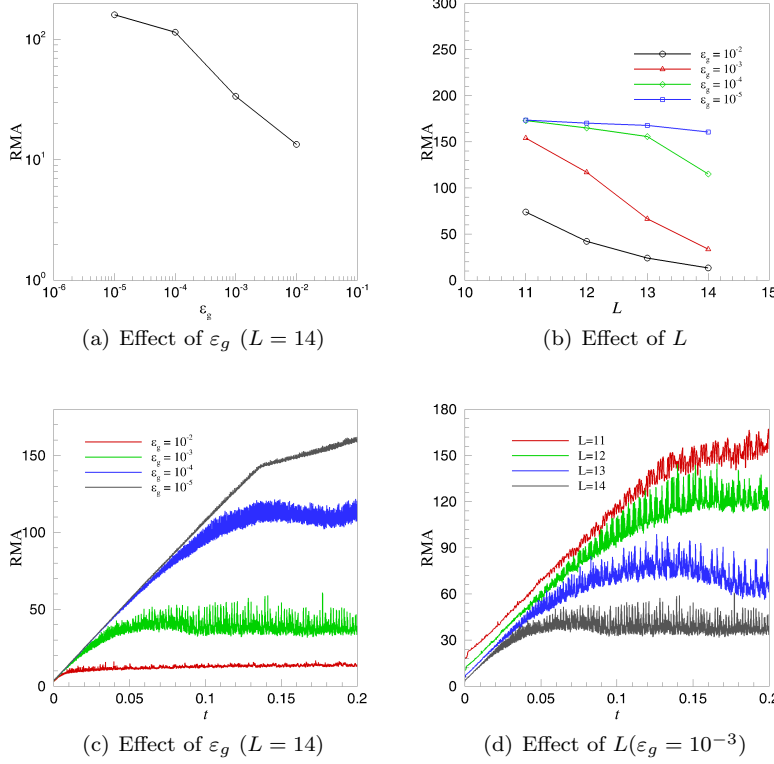


Fig. 17 The effect of the choice of L and ε_g on the memory gain of the 1D MHD equations with time. Higher relative memory allocations for lower L and stricter ε_g observed as expected.

5 Conclusions and Perspectives

In this investigation, we implement a scale-selective WENO-5 formulation utilizing an adaptive grid framework implemented using the method of finite volumes. The motivation behind this implementation is to modify the nonlinearity of the dissipative stencil of interfacial state reconstructions adaptively, i.e., to

have smooth regions with linear reconstructions with progressively increasingly nonlinearity towards the finest resolutions present in the solution field. For the purpose of testing this framework, we implement a one-dimensional system of hyperbolic laws for different physical systems such as the one-equation Inviscid Burgers' case with a moving shock, the dam break problem described by the shallow water equations, the famous Sod shock tube problem given by the Euler equations and the seven-equation Magnetohydrodynamic conservation laws in the Brio-Wu shock tube configuration. The interfacial fluxes are reconstructed using the Rusanov Riemann solver and a CFL=0.5 is utilized for the dynamic time integration of the adaptive finite volume grid. A total-variation diminishing third-order in time Runge-Kutta integrator is used for explicit time integration.

A graded tree approach is utilized for the progressive refinement of the domain from a root cell and each node of this tree possesses a unique integer identifier related to its level of resolution, neighbors, parent and children. This allows for $O(N \log N)$ look up of any particular cell in the tree. For the purpose of adaptation we utilize the concept of generalized wavelets within a collocated finite volume domain. Wavelet coefficients are calculated using fifth-order polynomial interpolation of cell-averaged quantities at different levels of resolution to determine a measure of the local irregularity. Through this, decisions are taken to refine areas which are in the vicinity of sharp gradients. The solution of the system is evolved only at the locally refined nodes of the domain (termed leaves) and this update in the solution is then projected towards the low resolution cell averages. For the purpose of flux calculation, virtual cells are added to leaves which do not have immediate neighbors for WENO reconstructions. These cells also serve the dual purpose of adapting the grid for admitting advective solutions. Conservation of fluxes are ensured through matching interfacial fluxes for neighbors at different levels of resolution. A dynamic memory implementation of this algorithm is utilized in the unordered map environment in C++ with a binary-tree data structure.

Our assessments on the various one-dimensional hyperbolic systems lead us to the following primary conclusions

- The utilization of the scale-selective WENO-5 implementation retains the mesh-refinement properties observed in [44] wherein higher gains are observed for lower thresholds and larger choices for the maximum level of resolution.
- The shock capturing behavior of the WENO-5 approach remains viable and all our numerical systems show well resolved standing and moving discontinuities with minimal shock front dissipation.
- Systems showing a large number of sharp discontinuities are more prone to reduced memory and CPU benefits. This can be inferred from the cases comparing the MHD and Euler shock tube problems. This behavior can be offset by further refining the level of resolution or reducing the global threshold for wavelet coefficient truncation.

In terms of the bigger picture, this investigation forms a benchmark framework for the proposed scale-selective AMR methodology and provides a preliminary validation of the concepts put forward. The next step in our long-term strategy is to incorporate this framework for the validation of multidimensional hyperbolic problems and to finally simulate the Navier-Stokes equations to quantify the statistical benefits of the scale selective dissipation from an ILES point of view. It is our expectation that the localized dissipation through nonlinear stencils would enhance scaling law capture for compressible flows exhibiting sharp gradients on coarser grids. From that perspective, the conclusions of this primary investigation suggest that the hybrid scale-selective WENO-5 and AMR methodology put forth represents a promising technique for procuring an enhanced accuracy at the different scales of a large eddy simulation without compromising on monotonic dissipation in the vicinity of extremely sharp gradients.

A Cell Class Object

```

class Cell
{
public:
    //Physical Quantities
    double q[n_eq] = {0.0}; //Conserved variables; n_eq is the
        number of conserved variables
    double xx=0.0; //Location in physical space -
        initialized at zero
    double cell_length=0.0; //Cell size - initialized at zero
    //Face fluxes
    double f_p[n_eq] = {0.0}; //Right Face flux
    double f_m[n_eq] = {0.0}; //Left Face flux
    //Third-Order Runge-Kutta storage
    double q_rhs[n_eq] = {0.0}; //Right hand side
    double q_temp[n_eq] = {0.0}; //A temporary array of conserved
        variables
    double q_1[n_eq] = {0.0}; //Runge-Kutta Substage 1 storage
    double q_2[n_eq] = {0.0}; //Runge-Kutta Substage 2 storage

    //MRA based storage
    int i; //Spatial index
    int level; //Level of resolution
    double det[n_eq]={0.0}; //Detail - initialized at zero

    //MRA based flags
    int keep_flag = 1; //if 1->keep instance else delete
        from memory
    int leaf = 0; //if 1->leaf else has children
    int new_cell = 0; //if 1->this is a newly formed cell
    int virt = 0; //if 1->this is a virtual cell

    //Pointers to other class objects - initialized as null pointers
    Cell* parent=nullptr; //Pointer to parent

    Cell* left_child=nullptr; //Pointer to left child
    Cell* right_child=nullptr; //Pointer to right child

    Cell* left_level=nullptr; //Pointer to left neighbor at the
        same level of resolution
    Cell* right_level=nullptr; //Pointer to right neighbor at the
        same level of resolution

    //Constructor
    Cell(){};
    //Destructor
    ~Cell(){};
};

```

Listing 1 Cell Class Object

B Searching binary tree for Cell

```

Cell* findcell(int i, unordered_map<int, Cell*> &Tree)
{
    //Note that first element of our unordered_map has to be the root.
    int child_tracker[max_level]; //max_level is the
    //predetermined maximum level of resolution in the computation
    int level; //Defining a integer
    //value to track levels

    if (i==1) //Root given by i==1
    {
        //always remains in Tree
        Cell* loc = Tree[1];
        return loc; //Return the pointer
        //to the cell class object
    }

    for (level=0; level<max_level; level++) //Increment after body
    { //of loop executes
        if (i%2==0) //left child
        {
            child_tracker[level]=1;
        }
        else if (i%2!=0) //right child
        {
            child_tracker[level]=2;
        }

        i = i/2;

        if (i==1) //We have reached root
        {
            break;
        }
    }

    Cell* loc=Tree[1]; //Start search from
    //root

    do{
        if (child_tracker[level]==1) //Go to left child
        {
            loc = loc->left_child;
        }
        else if (child_tracker[level]==2) //Go to right child
        {
            loc = loc->right_child;
        }
        level--;
    }while (level>=0);

    return loc; //Return the pointer
    //to the cell class object
}

```

Listing 2 Implementation of findcell function

References

1. N. A. Adams and K. Shariff. A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems. *J. Comput. Phys.*, 127(1):27–51, 1996.
2. J. M. Alam and J. C. Lin. Toward a fully Lagrangian atmospheric modeling system. *Mon. Weather Rev.*, 136(12):4653–4667, 2008.
3. D. Ambrosi. Approximation of shallow water equations by Roe’s Riemann solver. *Int. J. Numer. Meth. Fl.*, 20(2):157–168, 1995.
4. E. Bacry, S. Mallat, and G. Papanicolaou. A wavelet based space-time adaptive numerical method for partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 26(7):793–834, 1992.
5. A. Baeza, A. Martínez-Gavara, and P. Mulet. Adaptation based on interpolation errors for high order mesh refinement methods applied to conservation laws. *Appl. Numer. Math.*, 62(4):278–296, 2012.
6. D. S. Balsara, S. Garain, and C. Shu. An efficient class of WENO schemes with adaptive order. *J. Comput. Phys.*, 326:780–804, 2016.
7. D. S. Balsara and C. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.*, 160(2):405–452, 2000.
8. M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82(1):64–84, 1989.
9. M. J. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53(3):484–512, 1984.
10. B. L. Bihari and A. Harten. Multiresolution schemes for the numerical solution of 2-D conservation laws I. *SIAM J. Sci. Comput.*, 18(2):315–354, 1997.
11. F. Bramkamp, P. Lamby, and S. Müller. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *J. Comput. Phys.*, 197(2):460–490, 2004.
12. M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J. Comput. Phys.*, 75(2):400–422, 1988.
13. W. Cai and J. Wang. Adaptive multiresolution collocation methods for initial-boundary value problems of nonlinear PDEs. *SIAM J. Numer. Anal.*, 33(3):937–970, 1996.
14. A. Cohen. Wavelet methods in numerical analysis. *Handb. Numer. Anal.*, 7:417–711, 2000.
15. A. Cohen, S. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comput.*, 72(241):183–225, 2003.
16. I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. Pure. Appl. Math.*, 41(7):909–996, 1988.
17. R. Deiterding, M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. Adaptive multiresolution or adaptive mesh refinement? a case study for 2D Euler equations. In *ESAIM: Proceedings*, volume 29, pages 28–42. EDP Sciences, 2009.
18. M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent. New resolution strategy for multiscale reaction waves using time operator splitting, space adaptive multiresolution, and dedicated high order implicit/explicit time integrators. *SIAM J. Sci. Comput.*, 34(1):A76–A104, 2012.
19. M. Fujii and W. J. Hoefer. Interpolating wavelet collocation method of time dependent Maxwells equations: characterization of electrically large optical waveguide discontinuities. *J. Comput. Phys.*, 186(2):666–689, 2003.
20. S. Gottlieb and C. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comput.*, 67(221):73–85, 1998.
21. F. F. Grinstein, L. G. Margolin, and W. J. Rider. *Implicit large eddy simulation: computing turbulent fluid dynamics*. Cambridge university press, 2007.
22. Y. Ha, C. H. Kim, Y. J. Lee, and J. Yoon. An improved weighted essentially non-oscillatory scheme with a new smoothness indicator. *J. Comput. Phys.*, 232(1):68–86, 2013.
23. A. Harten. Discrete multi-resolution analysis and generalized wavelets. *Appl. Numer. Math.*, 12(1-3):153–192, 1993.
24. A. Harten. Adaptive multiresolution schemes for shock computations. *J. Comput. Phys.*, 115(2):319–338, 1994.

25. A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun. Pure. Appl. Math.*, 48(12):1305–1342, 1995.
26. A. Harten. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.*, 33(3):1205–1256, 1996.
27. A. K. Henrick, T. D. Aslam, and J. M. Powers. Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *J. Comput. Phys.*, 207(2):542–567, 2005.
28. D. J. Hill and D. I. Pullin. Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shocks. *J. Comput. Phys.*, 194(2):435–450, 2004.
29. C. Huang. WENO scheme with new smoothness indicator for Hamilton-Jacobi equation. *Appl. Math. Comput.*, 290:21–32, 2016.
30. L. Jameson. A wavelet-optimized, very high order adaptive grid and order numerical method. *SIAM J. Sci. Comput.*, 19(6):1980–2013, 1998.
31. G. Jiang and C. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202–228, 1996.
32. C. H. Kim, Y. Ha, and J. Yoon. Modified non-linear weights for fifth-order weighted essentially non-oscillatory schemes. *J. Sci. Comput.*, 67(1):299–323, 2016.
33. D. Kim and J. H. Kwon. A high-order accurate hybrid scheme using a central flux scheme and a WENO scheme for compressible flowfield analysis. *J. Comput. Phys.*, 210(2):554–583, 2005.
34. V. Kumar and M. Mehra. Wavelet optimized finite difference method using interpolating wavelets for self-adjoint singularly perturbed problems. *J. Comput. Appl. Math.*, 230(2):803–812, 2009.
35. C. B. Laney. *Computational gasdynamics*. Cambridge university press, 1998.
36. M. Latini, O. Schilling, and W. S. Don. Effects of WENO flux reconstruction order and spatial resolution on reshocked two-dimensional Richtmyer-Meshkov instability. *J. Comput. Phys.*, 221(2):805–836, 2007.
37. H. Li, S. Do, and M. Kang. A wavelet-based adaptive weno algorithm for euler equations. *Computers & Fluids*, 123:10–22, 2015.
38. X. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.
39. J. Luo and L. Jameson. A wavelet-based technique for identifying, labeling, and tracking of ocean eddies. *J. Atmos. Ocean. Tech.*, 19(3):381–390, 2002.
40. M. P. Martín, E. M. Taylor, M. Wu, and V. G. Weirs. A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence. *J. Comput. Phys.*, 220(1):270–289, 2006.
41. S. Paolucci, Z. J. Zikoski, and D. Wirasaet. WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. part I. Accuracy and efficiency of algorithm. *J. Comput. Phys.*, 272:814–841, 2014.
42. S. Pirozzoli. Conservative hybrid compact-weno schemes for shock-turbulence interaction. *J. Comput. Phys.*, 178(1):81–117, 2002.
43. Y. Ren, L. Miao'er, and H. Zhang. A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.*, 192(2):365–386, 2003.
44. O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.*, 188(2):493–523, 2003.
45. V. Rusanov. The calculation of the interaction of non-stationary shock waves with barriers. *Vysisl. Mat. i Mat. Fiz.*, 1:267–279, 1961.
46. K. Schneider and O. V. Vasilyev. Wavelet methods in computational fluid dynamics. *Annu. Rev. Fluid Mech.*, 42:473–503, 2010.
47. T. Sonar and E. Süli. A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. *Numer. Math.*, 78(4):619–658, 1998.
48. W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1998.
49. W. Sweldens and P. Schröder. Building your own wavelets at home. *Lect. Notes. Earth Sci.*, pages 72–107, 2000.
50. C. Tenaud and M. Duarte. Tutorials on Adaptive multiresolution for mesh refinement applied to fluid dynamics and reactive media problems. In *ESAIM: Proceedings*, volume 34, pages 184–239. EDP Sciences, 2011.

-
51. O. V. Vasilyev. Solving multi-dimensional evolution problems with localized structures using second generation wavelets. *Int. J. Comput. Fluid. D.*, 17(2):151–168, 2003.
 52. O. V. Vasilyev and C. Bowman. Second-generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.*, 165(2):660–693, 2000.
 53. O. V. Vasilyev and S. Paolucci. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.*, 125(2):498–512, 1996.
 54. J. E. Weiss. Applications of compactly supported wavelets to the numerical solution of partial differential equations. In *Photonics Technologies for Robotics, Automation, and Manufacturing*, pages 106–117. International Society for Optics and Photonics, 2004.
 55. G. B. Whitham. *Linear and nonlinear waves*, volume 42. John Wiley & Sons, 2011.