

RESEARCH ARTICLE

An adaptive multilevel wavelet framework for scale-selective WENO reconstruction schemes

R. Maulik¹ | O. San¹  | R. Behera² 

¹School of Mechanical and Aerospace Engineering, Oklahoma State University, Oklahoma, USA

²Department of Mathematics and Statistics, Indian Institute of Science Education and Research Kolkata, Nadia, India

Correspondence

O. San, School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA.

Email: osan@okstate.edu

Funding information

National Science Foundation, Grant/Award Number: OCI1126330; Graduate College Summer Research Fellowship; Oklahoma State University; Science and Engineering Research Board (SERB); Department of Science and Technology, India, Grant/Award Number: MTR/2017/000417

Summary

We put forth a dynamic computing framework for scale-selective adaptation of weighted essential nonoscillatory (WENO) schemes for the simulation of hyperbolic conservation laws exhibiting strong discontinuities. A multilevel wavelet-based multiresolution procedure, embedded in a conservative finite volume formulation, is used for a twofold purpose. (i) a dynamic grid adaptation of the solution field for redistributing grid points optimally (in some sense) according to the underlying flow structures, and (ii) a dynamic minimization of the in-built artificial dissipation of WENO schemes. Taking advantage of the structure detection properties of this multiresolution 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. Therefore, 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 providing a scale-selective dissipation process without a compromise on shock capturing behavior for conservation laws, which would be a strong contender for dynamic implicit large eddy simulation approaches.

KEYWORDS

adaptive mesh refinement, compact orthogonal wavelets, finite volume framework, multiresolution analysis, scale-selective dissipation, WENO reconstruction

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,^{1,2} nonlinear optics,³ and geophysical applications.⁴⁻⁶ These pose considerable challenges to computational and mathematical frameworks due to the requirement of very high degrees of freedom and 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 lower degrees of resolution. 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. Unfortunately, the number of relevant degrees of freedom is finite in truncated field computations.^{7,8} While improved hardware and data-oriented strategies such as parallel computing have made great strides in the past few decades, memory costs and computational expenses associated with massively parallel uniform grid simulations represent a great motivation for utilizing adaptive grid strategies for 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, ie, being localized in both space and scale.^{9–11} Traditionally, mesh refinements have usually been constructed according to the presence of sharp gradients in the field^{12,13} or local residuals.¹⁴ However, a significant limitation to these approaches is the lack of reliable error control. Discrete wavelet-based techniques are commonly used for mesh refinement purposes^{15–22} and may be considered a similar approach to the framework proposed here.

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.²³ 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 3 main classes as follows: pure wavelet, multiresolution, and wavelet optimized methods. Pure wavelet schemes (either Galerkin or collocation) are either used directly to discretize the underlying governing law or their properties are used for grid adaptation, preconditioning, or inversion of the operators and fast interpolation. Furthermore, adaptive wavelet Galerkin methods²⁴ are used to solve problems in wavelet coefficient space, whereas adaptive wavelet collocation methods²⁵ are used to solve problems in physical space on an adaptive computational grid. We must note that this class of methods does not utilize finite difference or volume-type discretization. In contrast, the idea of the multiresolution method is simply to use a hierarchical data representation of the solution, and in essence, there is no grid adaptation, but computational costs are reduced through inexpensive flux calculations via interpolation across different levels of resolution rather than on a uniform grid. This choice of interpolation or explicit flux calculation is decided through the decay of wavelet coefficients, which yield information on the local regularity of the solution through an estimation of the truncation error. Hence, a given discretization on a uniform mesh can be accelerated as the number of costly flux evaluations is significantly reduced while maintaining the accuracy of the discretization.^{15–17} Finally, wavelet-optimized methods include algorithms based on classical discretizations (eg, by finite differences or finite volumes) but use wavelet analysis either to define adaptive meshes or to speed up linear algebra.^{19,26–30} In our investigation, we utilize the latter approach. We note here that multiresolution analysis is utilized for the latter 2 approaches, but wavelet-optimized methods actually adapt grids dynamically from information obtained through this analysis. In that context, a discrete wavelet transform is utilized to bypass a monitor function for the purpose of refinement. Further details of the classification of wavelet methods may be found in the comprehensive review of Schneider and Vasilyev.³¹ We emphasize that the wavelet-optimized methods are also referred to as adaptive multiresolution approaches.

In this investigation, we follow the methodologies given in other works^{26,32–35} 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 by Harten in his other works^{15–17} and further used in the works of Bihari³⁶ and Harten³⁷ for multiresolution computations for hyperbolic problems. Indeed, the same concept of generalized wavelets has been used in the work of Roussel et al²⁶ 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.^{21,38,39} 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 update⁴⁰). 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” that 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. 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 within the finite volume context. Interfacial state reconstructions for the conserved

variables within the finite volume framework are obtained using the classical fifth-order weighted essential nonoscillatory (WENO) schemes.⁴¹ 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.⁴² 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.²⁰

A limitation of WENO schemes arises from the fact that these are too dissipative for the accurate modeling of turbulent flow.⁴³ 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.^{44–46} We follow the weighted average methodology of “switching” between the linear upwind and fifth-order WENO schemes as given in the works of Ren et al⁴⁷ and Kim and Kwon,⁴⁸ 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 (ILES) 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 Li et al⁴⁹ where second-generation lifted wavelets are utilized in the method of finite differences to identify areas of high-resolution irregularity and to use fifth-order WENO solely at the highest levels of refinement, whereas the inverse wavelet transform is used to calculate stencil for finite difference approximations of gradients recursively at different levels of resolution.²¹ We note that several works in literature^{16,37,50} 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 6-point stencil. We reiterate that the choice of an adaptive multiresolution analysis provides us with a computational advantage over the approach utilized in the work of Harten³⁷ where transforms based on compact orthogonal wavelets are utilized purely for the purpose of reducing computation attributed to costly flux calculations but not for dynamic grid refinement. 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 the following sections. 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 for solving Euler equations of gas dynamics.⁵¹ Similar hyperbolic systems of equations that are tested include the Brio-Wu shock tube problem utilizing the magnetohydrodynamic (MHD) equations⁵² and the classical dam break problem on a wet bottom described by the shallow water equations.⁵³ 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 multilevel compact orthogonal wavelet-based finite volume method is implemented for the simulation of hyperbolic partial differential equations (PDEs).
- 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, and 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 and the proposed framework for conservative flux reconstruction at cell interfaces in Section 2. Section 3 introduces the multiresolution framework utilizes for adapting our finite volume grid for 1-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 1-dimensional hyperbolic system of equations in their conservative form as our underlying PDEs. 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 1-dimensional framework presented here may easily be extrapolated to multidimensional implementations.

2.2 | Finite volume framework

The semidiscrete 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 PDEs as a semidiscrete system of ordinary differential equations 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 aforementioned equation representing the combined effect of all the spatial derivatives in the governing equations. The aforementioned ordinary differential equation 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 scheme⁵⁴ 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 WENO approach, which was first introduced in the work of Liu et al.⁵⁵ The order of accuracy of these reconstructions depends on the length of the stencil chosen and affects 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 that are being reconstructed in this formulation. A modified implementation of the WENO-5 reconstruction⁵⁶ can be given by

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

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 $d_0 = 1/10$, $d_1 = 6/10$, and $d_2 = 3/10$ in Equation (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 (Jiang and Shu⁵⁶) scheme. Before proceeding, we must observe that there are many variants of the WENO reconstructions described (eg, in other works⁵⁷⁻⁶⁰), 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^{41,61} or reduced computational expense. The interested reader is directed to the work of Latini et al⁶² for an excellent discussion on the effect of the order of reconstruction on a classical 2-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 that may be detrimental in ILES 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 and l is an arbitrary level of local refinement. 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. Figure (1) graphically displays the scale-selective nature of our framework.

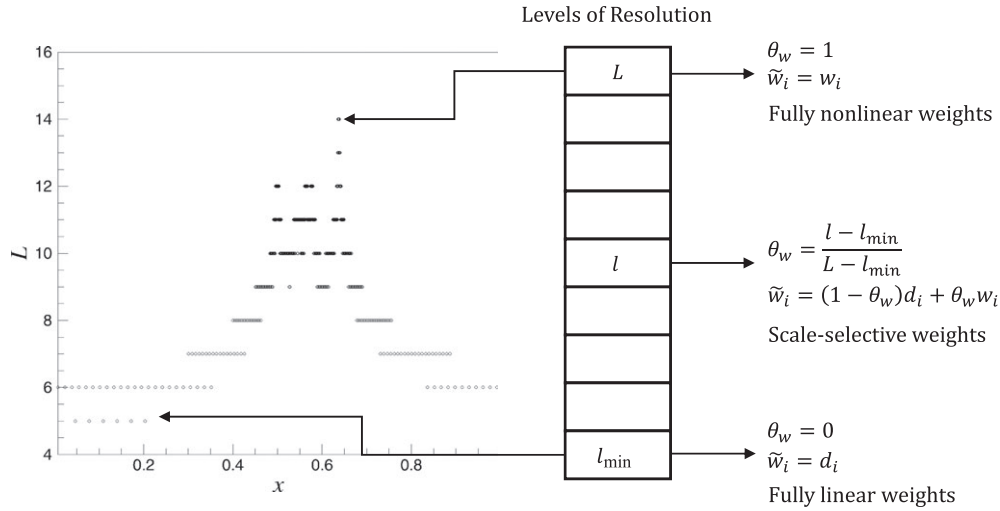


FIGURE 1 A schematic of the scale-selective adaptation of the nonlinear weights in the proposed framework. Notice how smoother areas are dissipated using linear fifth-order upwind stencils

2.5 | Riemann solver at interfaces

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 solver⁶³ utilizes information from the maximum local wave propagation speed to give us the following 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, (ie, $F^+ = F(q_{i+1/2}^+)$ 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, ie,

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

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

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

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 | ADAPTIVE MULTILEVEL WAVELET FRAMEWORK

3.1 | Compactly supported orthogonal wavelets

Here, we briefly recall the idea of compactly supported wavelets and multiresolution analysis. The idea of multiresolution analysis was first introduced by Mallat⁶⁴ and Meyer⁶⁵ in the context of wavelet analysis. This idea dealt with a general formalism for the construction of an orthogonal basis of wavelets, and indeed, multiresolution analysis is central to all constructions of wavelet basis. To avoid potential confusion, we reiterate here that the phrase “multiresolution analysis” deals with the mathematical framework for identifying scale content at different resolutions, whereas “adaptive multiresolution analysis” utilizes this mathematical formalism for adaptive mesh refinement. Our multiresolution analysis consists of a sequence $\{G^l : l \in \mathbb{Z}\}$ of embedded closed subspaces of $L_2(\mathbb{R})$ that satisfy the following axiom.⁶⁶

- $\dots \subset G^{-2} \subset G^{-1} \subset G^0 \subset G^1 \subset G^2 \subset \dots$
- $\bigcup_{l=-\infty}^{\infty} G^l$ is dense in $L_2(\mathbb{R})$, ie, $\overline{\bigcup_{l=-\infty}^{\infty} G^l} = L_2(\mathbb{R})$.

- $\bigcap_{l=-\infty}^{\infty} G^l = \{0\}$.
- $u(x) \in G^l$ if and only if $u(2x) \in G^{l+1}$, and for all $l \in \mathbb{Z}$, there exist a function $\phi \in G^0$, such that $\{\phi_i^0 = \phi(2x - i), i \in \mathbb{Z}\}$ is an orthonormal basis for G^0 . That is,

$$\|u\|^2 = \int_{-\infty}^{\infty} |u(x)|^2 dx = \sum_{i=-\infty}^{\infty} |\langle u, \phi_i^0 \rangle|^2, \quad \forall u \in G^0, \quad (15)$$

and it is also required that ϕ has unit area, ie, $\int_{-\infty}^{\infty} \phi(x) dx = 1$. Now, we define \mathcal{W}^l to be the orthogonal complement of G^l in G^{l+1} , ie, $G^l \perp \mathcal{W}^l$ and $G^{l+1} = G^l \oplus \mathcal{W}^l$. In addition, there exists a function called the scaling function $\phi(x) \in G^0$, such that the sequence $\phi_i^l(x) = 2^{l/2} \phi(2^l x - i)$ is an orthonormal basis for G^l and $\psi_i^l(x) = 2^{l/2} \psi(2^l x - i)$ is an orthonormal basis for \mathcal{W}^l , where $i, l \in \mathbb{Z}$, with l being the dilation parameter analogous to a certain level of refinement in resolution and i is a translation parameter. Mathematically, one can introduce at each level l , the subspace \mathcal{W}^l , defined as the orthogonal complement of G^l in G^{l+1} .

Since $\psi_i^l \in \mathcal{W}^l$, it follows that ψ_i^l is orthogonal to ϕ_i^l because $\phi_i^l \in G^l$ and $G^l \perp \mathcal{W}^l$. Moreover, because all \mathcal{W}^l are mutually orthogonal, it follows that the wavelets are orthogonal across scale. Therefore, we have the following relations:

$$\int_{-\infty}^{\infty} \phi_k^j(x) \phi_l^j(x) dx = \delta_{k,l}, \quad \int_{-\infty}^{\infty} \psi_k^i(x) \psi_l^j(x) dx = \delta_{i,j} \delta_{k,l}, \quad (16)$$

$$\int_{-\infty}^{\infty} \phi_i^k(x) \psi_j^l(x) dx = 0, \quad j \geq i, \quad (17)$$

where $i, j, k, l \in \mathbb{Z}$, and $\delta_{k,l}$ is the Kronecker delta defined as

$$\delta_{k,l} = \begin{cases} 0, & \text{if } k \neq l, \\ 1, & \text{if } k = l, \end{cases} \quad (18)$$

We note that in the previous equations that i, j, k, l are solely utilized for exhibiting the phenomenon for orthogonality across scales for wavelets and are not related to the prior definitions of scaling and translation. Furthermore, since $G^0 \subset G^1$, any function in G^0 can be expanded in terms of basis function G^1 . In particular, $\phi(x) = \phi_0^0(x)$. Since the scaling functions are supported in a compact sense, only finitely many a_k will be nonzero. We then have

$$\phi(x) = 2 \sum_{k=0}^{E-1} a_k \phi(2x - k), \quad (19)$$

where $a_k = \int_{-\infty}^{\infty} \phi(x) \phi_k^1(x) dx$. Equation (19) is fundamental for wavelet theory and is known as the dilation equation. Note that it is common to call a_0, a_1, \dots, a_{E-1} as the filter coefficients. E is an even positive integer called the wavelet genus. The scaling function is uniquely characterized by these coefficients. Similar to Equation (19), we can write a relation for the basis wavelet ψ . Since $\psi \in W^0$ and $W^0 \subset G^1$, we can expand ψ (called the wavelet equation) as

$$\psi(x) = 2 \sum_{k=0}^{E-1} (-1)^k b_k \phi(2x - k). \quad (20)$$

where $b_k = a_{E-1-k}$. Furthermore, an accuracy is specified by requiring that $\psi(x) = \psi_0^0(x)$ is satisfied for all $m = 0, 1, \dots, M-1$, where $M = \frac{E}{2}$ are the vanishing moments

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0. \quad (21)$$

Many of the functions $\phi(x)$ that are used in numerical analysis automatically satisfy a dilation equation.^{66,67} Let us consider $\phi(x)$ as a box function, ie,

$$\phi(x) = \begin{cases} 1 & -1 \leq x \leq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

$$(23)$$

Our justification for choosing a box function is to incorporate the finite volume approach for conservative flux calculation into our dynamically adapted grid. We note that the choice of $E = 2$ corresponds to the box scaling function. A Dirac delta function would be an implementation of the finite difference method but with associated issues of conservative approximations. Thus, the scaling function and wavelet function become

$$\phi(x) = \phi(2x) + \phi(2x - 1) \quad \text{and} \quad \psi(x) = \phi(2x) - \phi(2x - 1). \quad (24)$$

Now, we may use the concept of discrete multiresolution analysis and the associated terminology of Harten¹⁵ to construct an adaptive multilevel wavelet framework. Consider a set of nested volumes in increasing order of cell sizes as

$$\left\{ \{x_i^l\}_{i=1}^{N_l} \right\}_{l=L-1}^0, \quad x_i^l = i \cdot h_l, \quad h_l = 2^l h_L, \quad N_l = \frac{1}{h_l}, \quad (25)$$

and corresponding quantities

$$u_i^l = \langle u, \phi_i^l \rangle, \quad \phi_i^l(x) = \frac{1}{h_l} \phi\left(\frac{x - x_i^l}{h_l}\right), \quad 1 \leq i \leq N_l, \quad 0 \leq l \leq L, \quad (26)$$

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

$$\left\{ \{u_i^l\}_{i=1}^{N_l} \right\}_{l=L}^0, \quad (27)$$

provided the relationship given by Equation (19) holds. The interpretation of the aforementioned 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 level $l+1$ represent all larger scales l to 0, provided a suitable set of coefficients a_k are chosen, ie,

$$\left\langle u, \phi_i^l - \sum_k^{E-1} a_k \phi_{2i+k}^{l+1} \right\rangle = 0. \quad (28)$$

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

$$\langle R(x; u^l), \phi_i^l(x) \rangle = u_i^l, \quad 1 \leq i \leq N_l. \quad (29)$$

A continuous approximation of the errors in scale capture between 2 conservative levels may be obtained through

$$Q^l(x; u) = R(x; u^{l+1}) - R(x; u^l), \quad (30)$$

and its Euclidean inner product with the scaling function gives us

$$d_i^{l+1} = \langle Q^l(x; u), \phi_i^{l+1}(x) \rangle. \quad (31)$$

If R is an orthogonal projection into the linear span of $\phi_i^l(x)$, ie,

$$R(x; u^l) = \sum_{i=1}^{N_l} u_i^l \phi_i^l(x), \quad (32)$$

then our reconstruction procedure basically constitutes the transformation of the discrete samples to the compactly supported orthonormal wavelet bases of Daubechies⁶⁶ and d_i^l may be assumed to be the *discrete wavelet coefficients* at level of resolution l .

3.2 | Multilevel representation of cell averages

As mentioned in the introduction, in this section, we utilize a computational domain constructed through nested cell volumes. Utilizing the terminology of Harten³⁷ and Roussel et al,²⁶ let G^l , $0 \leq l \leq L$, be a set of nested dyadic grids

$$G^l = \{x_i^l\}_{i=1}^{N_l}, \quad h_l = 2^l, \quad N_l = 2^l, \quad l \in [0, L], \quad (33)$$

which corresponds to a nested set of finite volumes on a 1-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)

$$\bar{u}_i^l = \frac{1}{h_l} \int_{x_{i-1}^l}^{x_i^l} u(x) dx. \quad (34)$$

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 1-dimensional structured dyadic grid may be represented as

$$\bar{u}_i^l = (P_{l+1 \rightarrow l} \bar{U}^{l+1})_i = \frac{1}{2} (\bar{u}_{2i}^{l+1} + \bar{u}_{2i+1}^{l+1}). \quad (35)$$

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). Here we would like to clarify a modification in notation where the values of i previously existing in a range $[1, N_i]$ for an arbitrary l now have unique identifiers within the entire nested domain for ease of subsequent data structure description. 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 by³⁷

$$\begin{aligned} \hat{u}_{2i}^{l+1} &= \bar{u}_i^l + \sum_{m=1}^2 \gamma_m (\bar{u}_{i+m}^l - \bar{u}_{i-m}^l), \\ \hat{u}_{2i+1}^{l+1} &= \bar{u}_i^l - \sum_{m=1}^2 \gamma_m (\bar{u}_{i+m}^l - \bar{u}_{i-m}^l). \end{aligned} \quad (36)$$

Our coefficients in the chosen interpolation scheme are given by

$$\gamma_1 = -\frac{22}{128}, \quad \gamma_2 = \frac{3}{128}. \quad (37)$$

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

$$\bar{d}_i^l = \bar{u}_i^l - \hat{u}_i^l. \quad (38)$$

We must remark here that, in the 1-dimensional case, only one of the details belonging to the 2 children of each cell are independent. This is 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

$$|V_i^l| \bar{u}_i^l = \sum_{p=2i}^{2i+1} |V_p^{l+1}| \hat{u}_p^{l+1}, \quad (39)$$

where p is summed for all the children cells of \bar{u}_i^l at the resolution of $l + 1$ and $|V_i|$ 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 other works^{26,32,36} 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(<i>Tree</i>)
2: Given solution at locally refined leaves
3: $l = L - 1$
4: while $l \neq 0$ do
5: Project to find cell average of parents
6: $l = l - 1$
7: end while
8: end procedure | <p>▷ For calculating cell averages of parents</p> <p>▷ Starting at finer resolution</p> <p>▷ Using Equation (35)</p> <p>▷ Advance towards root</p> |
|--|--|
-

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 Equation (36)
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 orthogonal wavelets with compact support. We follow in the footsteps of other works^{26,32,50} to identify some terminology related to our multiresolution framework.

- The root is the starting position of the tree data structure. As demonstrated subsequently, our 1-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 structure 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 2 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 Equation (36).

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 predecided number of fixed neighbors. A question that arises here is for the coarsest levels of resolution, ie, when there are an insufficient number of 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 Figure (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 1-dimensional case).

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 that 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 and with the cell's parent and its multiple children. Integer quantities corresponding to the level of resolution and 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

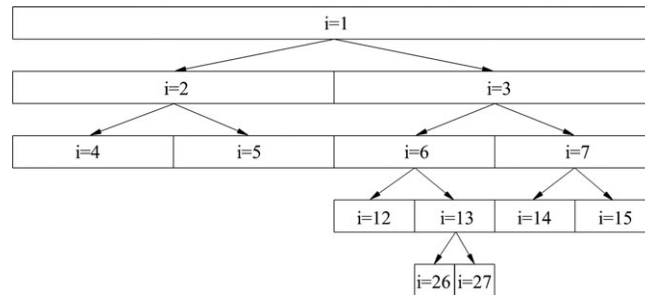


FIGURE 2 A schematic of the collocated finite volume adaptive mesh refinement framework

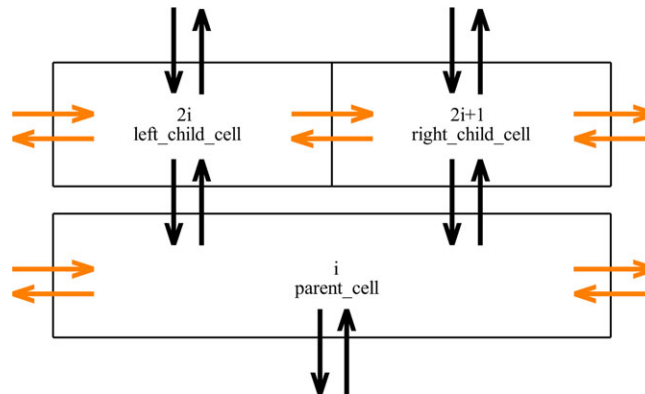


FIGURE 3 A schematic of the various pointers in an instance of the class object. Black pointers are parent-child connections across different resolutions, and the orange pointers are connected to the neighboring cells

content of the cell. Figure (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.

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. We remark that any cell at resolution l may be found through $2l$ steps, which point to $O(N \log N)$ complexity at the worst case. However, as mentioned by Roussel et al.,²⁶ in general, very few regions require the maximum amount of resolution, and the proposed algorithm is quite efficient. The Appendix contains the exact listings of our data structure and search algorithms.

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 the work of Cohen,⁶⁸ this threshold parameter is responsible for the control of the perturbation error introduced by the multiresolution procedure. Following the work of Harten,¹⁶ we define a resolution dependent (or local) threshold given by

$$\varepsilon_l = 2^{l-L} \varepsilon_g, \quad (40)$$

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.^{26,50,68} The pseudocode described in Algorithm 3 describes the thresholding procedure for our investigation for a 1-dimensional case with 1 conserved variable. This thresholding procedure is “predictive” in that it retains (or adds) new finite volume cells 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  $\bar{d}_i^l < \varepsilon_l$  then
6:       Delete children nodes  $\bar{u}_{2i}^{l+1}$  and  $\bar{u}_{2i+1}^{l+1}$                                 ▷ Since this level is not required
7:     else
8:       Keep nodes  $\bar{u}_{2i+q}^{l+1}$  where  $-K \leq q \leq K + 1$                                 ▷  $K = 1$  to capture advection
9:       if  $\bar{d}_i^l \geq 2^{2r} \varepsilon_l$  and  $l \neq L - 1$  then                                ▷  $r = 4$  for fifth-order polynomials
10:        Keep nodes given by  $\bar{u}_{2q}^{l+2}$  and  $\bar{u}_{2q+1}^{l+2}$ , where  $2i - K \leq q \leq 2i + 1 + K$ 
11:      end if
12:    end if
13:     $l = l - 1$                                                 ▷ Advance towards root
14:  end while
15: 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 particular level of resolution:

$$\bar{d}_i^l = \frac{|\bar{d}_i^l|_{L_1}}{\max_k |\bar{d}_k^l|_{L_1}}. \quad (41)$$

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 1 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

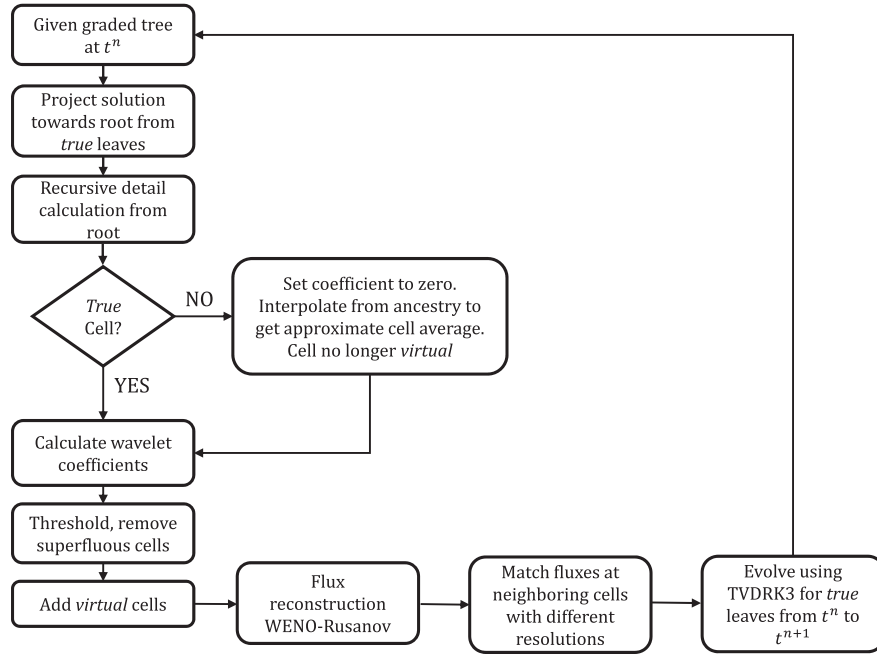


FIGURE 4 A flow-chart of processes involved in the dynamic adaptation of the grid in our framework. WENO, weighted essential nonoscillatory; TVDRK, total variation diminishing Runge-Kutta

to update the values in the virtual cell. A flowchart detailing the entire dynamic adaptive grid methodology is shown in Figure (4).

Algorithm 4 Virtual leaf addition

1: procedure VIRTUALADD(<i>Leaf</i>)	▷ 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 $\bar{u}_{i\pm n}^l$ do not exist then	▷ 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 $\bar{u}_{i\pm m}^{l+1}$	▷ 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	

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 (ie, it belongs to a prediction stencil),
- If it is required for the construction of interfacial fluxes for any leaf cells (ie, this cell is a virtually needed cell), and
- If a parent has 1 child that requires retention, all its children must be retained (ie, conservative retention).

4 | NUMERICAL EXPERIMENTS

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

We select a function given by

$$f(x) = \sin(2\pi(x - 0.5)) + e^{-10000(x-0.5)^2}, \quad (42)$$

for evaluating the grid adaptation ability of the proposed approach. As shown in Figure (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 Harten¹⁶ who notes that the 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 Figure (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 the work of Cohen et al.⁵⁰ We therefore propose the utilization of a *normalized* L_1 norm that 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 Figure (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 the works of Deiterding et al.⁶⁹ and Tenaud and Duarte.⁷⁰ 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). Figure (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 norm of the adaptive grid (ie, 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}), ie,

$$\text{RMA} = \frac{N_{\text{AMR}}}{N_{\text{UG}}} \times 100. \quad (43)$$

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 until the root of the domain, the reference uniform grid will assume a simple array-type data structure with the only the finest level of elements present in memory. The 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

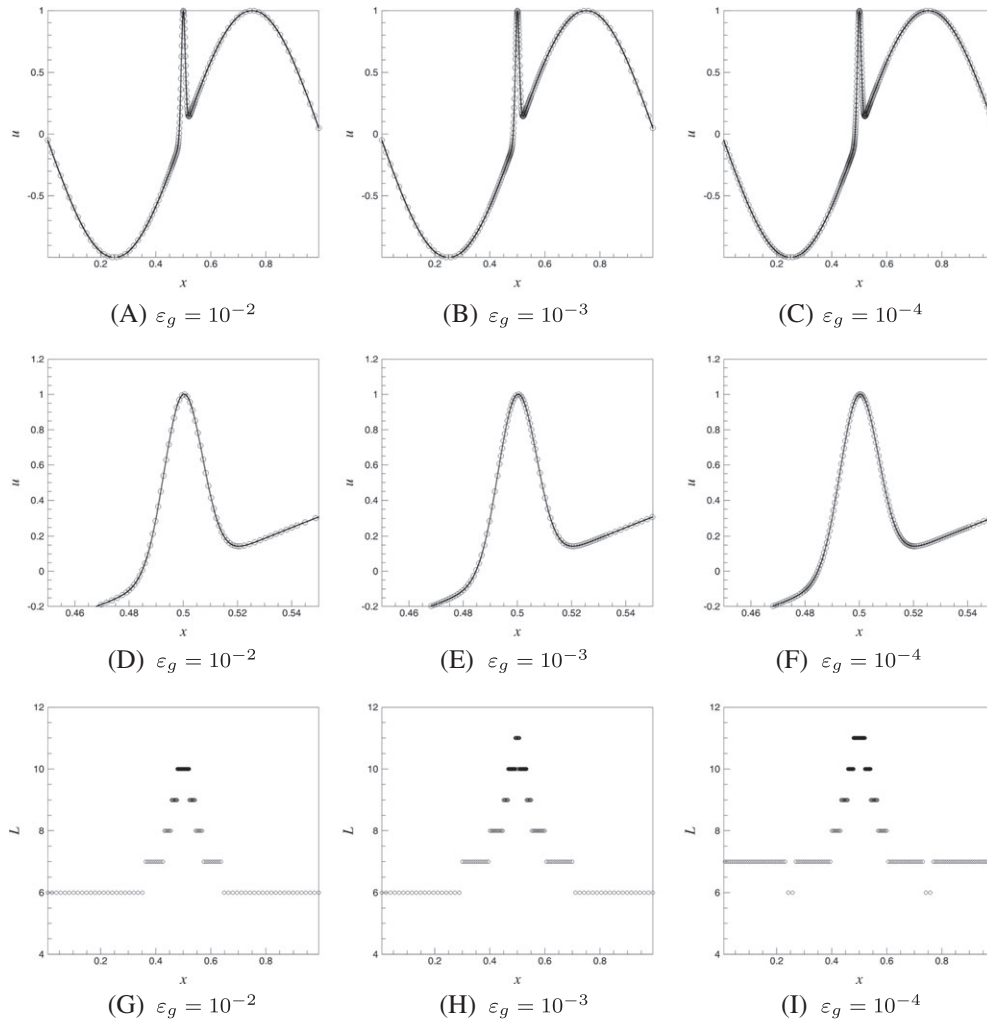


FIGURE 5 A visual representation of increasing refinement due to stricter thresholding. Fields for test function shown above, zoom in of sharp gradient at $x = 0.5$ shown in the middle, and corresponding levels of resolution shown below. The solid line is given by the exact solution. A, $\varepsilon_g = 10^{-2}$; B, $\varepsilon_g = 10^{-3}$; C, $\varepsilon_g = 10^{-4}$; D, $\varepsilon_g = 10^{-2}$; E, $\varepsilon_g = 10^{-3}$; F, $\varepsilon_g = 10^{-4}$; G, $\varepsilon_g = 10^{-2}$; H, $\varepsilon_g = 10^{-3}$; I, $\varepsilon_g = 10^{-4}$

grid. To clarify this particular metric, let us describe a sample case. An RMA of 20% for an adaptive multiresolution representation (AMR) simulation with maximum level $L = 12$ implies that

$$\begin{aligned} N_{UG} &= 2^{12} = 2048, \\ N_{AMR} &= \frac{\text{RMA}}{100} 2048 \approx 410. \end{aligned} \quad (44)$$

For our assessments with the static function, we obtain trends, which agree with the work of Roussel et al²⁶ that higher thresholds combined with lower maximum levels of resolution tend to result in high relative memory allocation, as shown in Figure (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.

4.2 | Nonlinear scalar conservation laws: Inviscid Burgers equation

Our first *dynamic* test case for the proposed framework is given by the hyperbolic inviscid Burgers equation given by the following initial and boundary conditions⁷¹:

$$u_t + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0, \quad x \in \mathbb{R}, \quad t \geq 0, \quad (45)$$

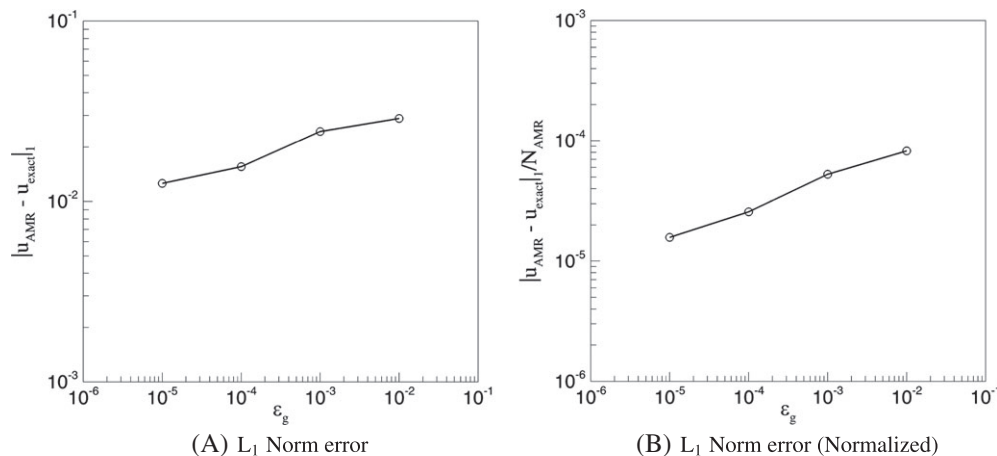


FIGURE 6 Varying L_1 norm errors (left) and L_1 norm errors normalized by the total number of points in the domain for the static function as a function of ϵ_g . On the right, we also have the linear fit of the thresholding criteria ϵ_g with the reduction in perturbation error

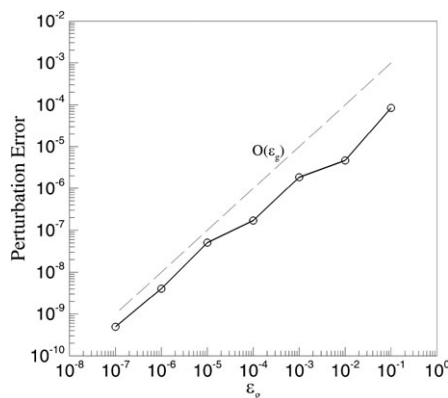


FIGURE 7 Perturbation error (with order of accuracy) for the mesh refinement strategy utilized in this investigation. One can observe a linear fit with ϵ_g

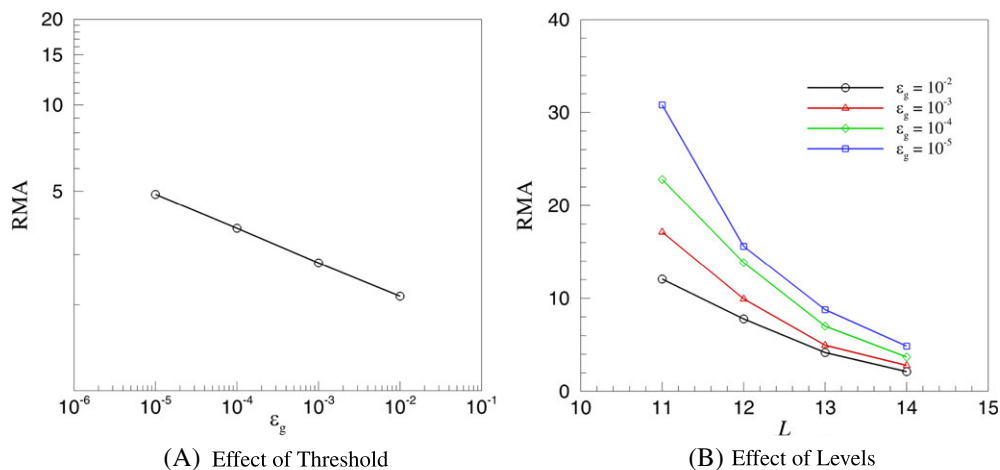


FIGURE 8 Memory requirements with different ϵ_g and levels of resolution. Observe how high thresholds and low maximum levels of resolution lead to infeasible memory gains. A, Effect of threshold; B, Effect of levels. RMA, relative memory allocation [Colour figure can be viewed at wileyonlinelibrary.com]

with initial conditions given by

$$u_0(x) = \begin{cases} 1, & \text{for } 0.5 \leq x < 0.6, \\ 0, & \text{otherwise.} \end{cases} \quad (46)$$

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

$$q = u \text{ and } F = \frac{u^2}{2}. \quad (47)$$

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

$$c_{i+1/2} = \max(|q_{i-2}|, |q_{i-1}|, |q_i|, |q_{i+1}|, |q_{i+2}|, |q_{i+3}|). \quad (48)$$

The described system of equations is solved for open boundary conditions in both positive and negative x directions until a nondimensional 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. Figure (9) details the

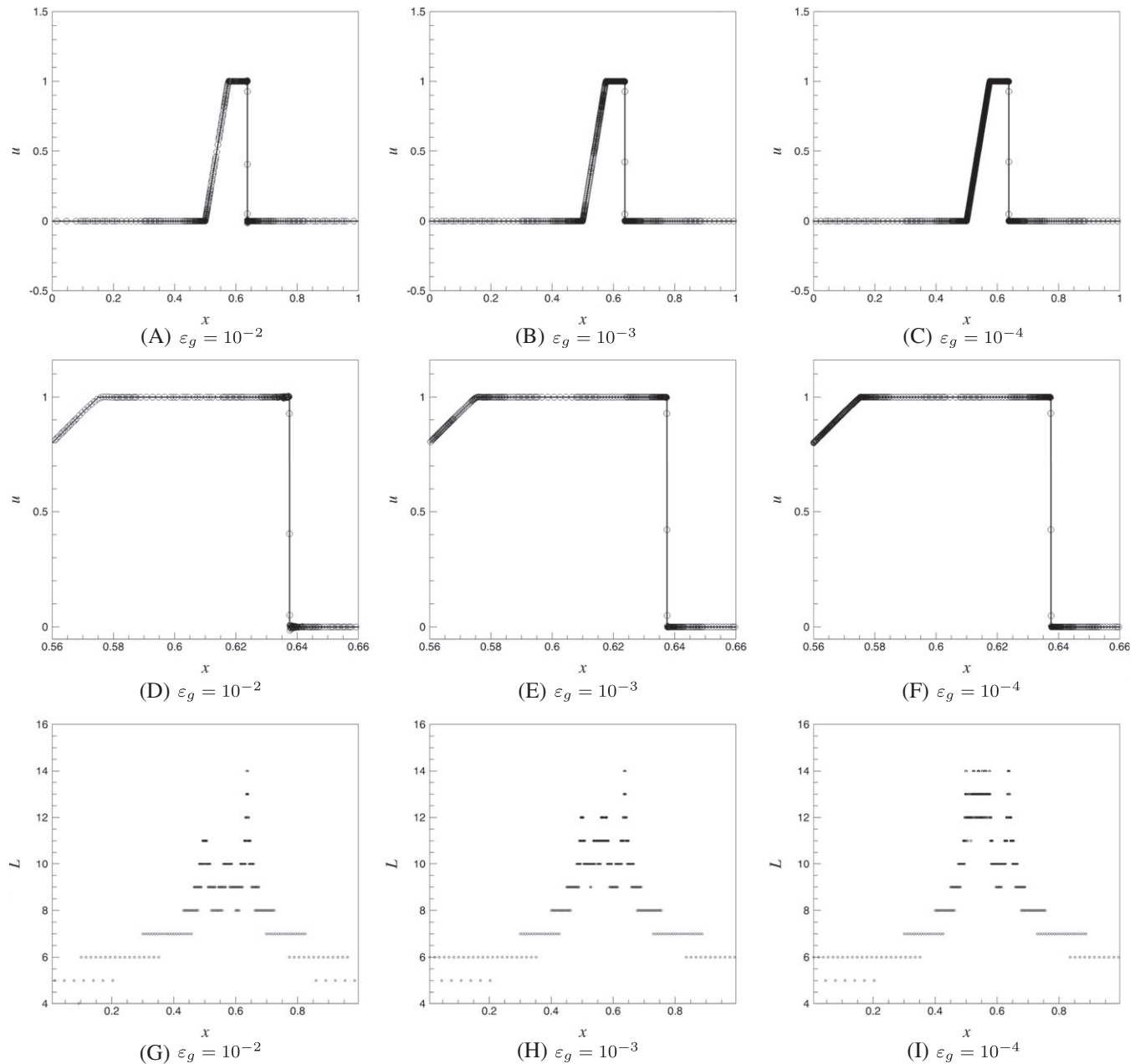


FIGURE 9 A visual representation of increasing refinement due to stricter thresholding. Fields for the inviscid Burgers' equation problem at $t = 0.075$ shown above, zoom ins of the moving shock front showing discontinuity resolution performance in the middle, and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65 536 degrees of freedom. A, $\epsilon_g = 10^{-2}$; B, $\epsilon_g = 10^{-3}$; C, $\epsilon_g = 10^{-4}$; D, $\epsilon_g = 10^{-2}$; E, $\epsilon_g = 10^{-3}$; F, $\epsilon_g = 10^{-4}$; G, $\epsilon_g = 10^{-2}$; H, $\epsilon_g = 10^{-3}$; I, $\epsilon_g = 10^{-4}$

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 cell around the vicinity of regions with sharp gradients. Our exact solution is computed using $N_{\text{UG}} = 65\,536$ 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 Figure (10), which details the time evolution of memory requirement ratios as a function of L and ε_g . As mentioned in the work of Roussel et al.,²⁶ 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:

$$\text{Gain} = \left(1 - \frac{t_{\text{AMR}}}{t_{\text{UG}}}\right) \times 100, \quad (49)$$

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

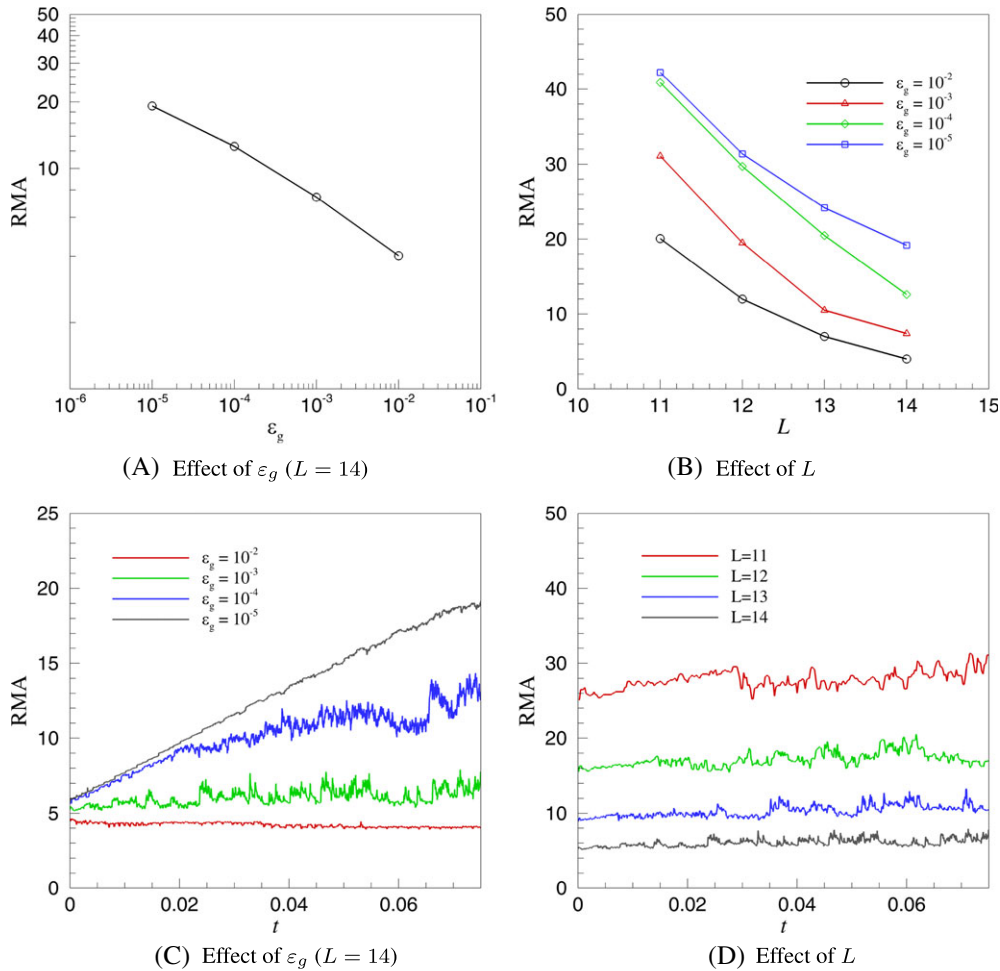


FIGURE 10 The effect of the choice of L and ε_g on the memory gain of the system with time. Higher relative memory allocations (RMAs) for lower max level of resolution L and stricter ε_g observed as expected. A, Effect of ε_g ($L = 14$); B, Effect of L ; C, Effect of ε_g ($L = 14$); D, Effect of L [Colour figure can be viewed at wileyonlinelibrary.com]

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

CPU Gain	$\epsilon_g=10^{-2}$	$\epsilon_g=10^{-3}$	$\epsilon_g=10^{-4}$	$\epsilon_g=10^{-5}$
$L = 11$	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

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 that 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 cluster may cause such slight discrepancies. We must remark here that the benefits we witness for the 1-dimensional problems we study here shall be amplified considerably when the proposed algorithms are extended to higher dimensions.

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 the work of Ambrosi.⁵³ The governing laws of the problem are given by the shallow water (or Saint-Venant) equations, which are given in their conservation form by

$$\mathbf{q} = \begin{bmatrix} \rho h \\ \rho h u \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho h u \\ \rho h u^2 + \frac{1}{2} \rho g h^2 \end{bmatrix}, \quad (50)$$

where ρ is the fluid density, h is the height of the fluid column in the computational domain, 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

$$r(A) = \max(|u|, |u - a|, |u + a|), \quad (51)$$

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

$$a = 2\sqrt{gh}. \quad (52)$$

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

$$h_0(x) = \begin{cases} 10, & \text{for } 0.0 \leq x \leq 0.5 \\ 1, & \text{for } 0.5 < x \leq 1.0. \end{cases} \text{ and } u_0(x) = 0. \quad (53)$$

As in the Burgers' equation, a reference solution computed on a uniform grid comprised of $N_{\text{UG}} = 65\,636$ finite volume cells is utilized for the purpose of numerical assessment. Figure (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 Figure (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 (ie, 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 $10^{-2} \leq \epsilon_g \leq 10^{-3}$ were feasible for excellent memory gains and 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 with the fine grid. For our results, here, we obtained a 40% benefit in terms of computational expense at $L = 14$ and $\epsilon_g = 10^{-2}$. The trends would suggest that increasing L beyond 14 would lead to further benefits in memory

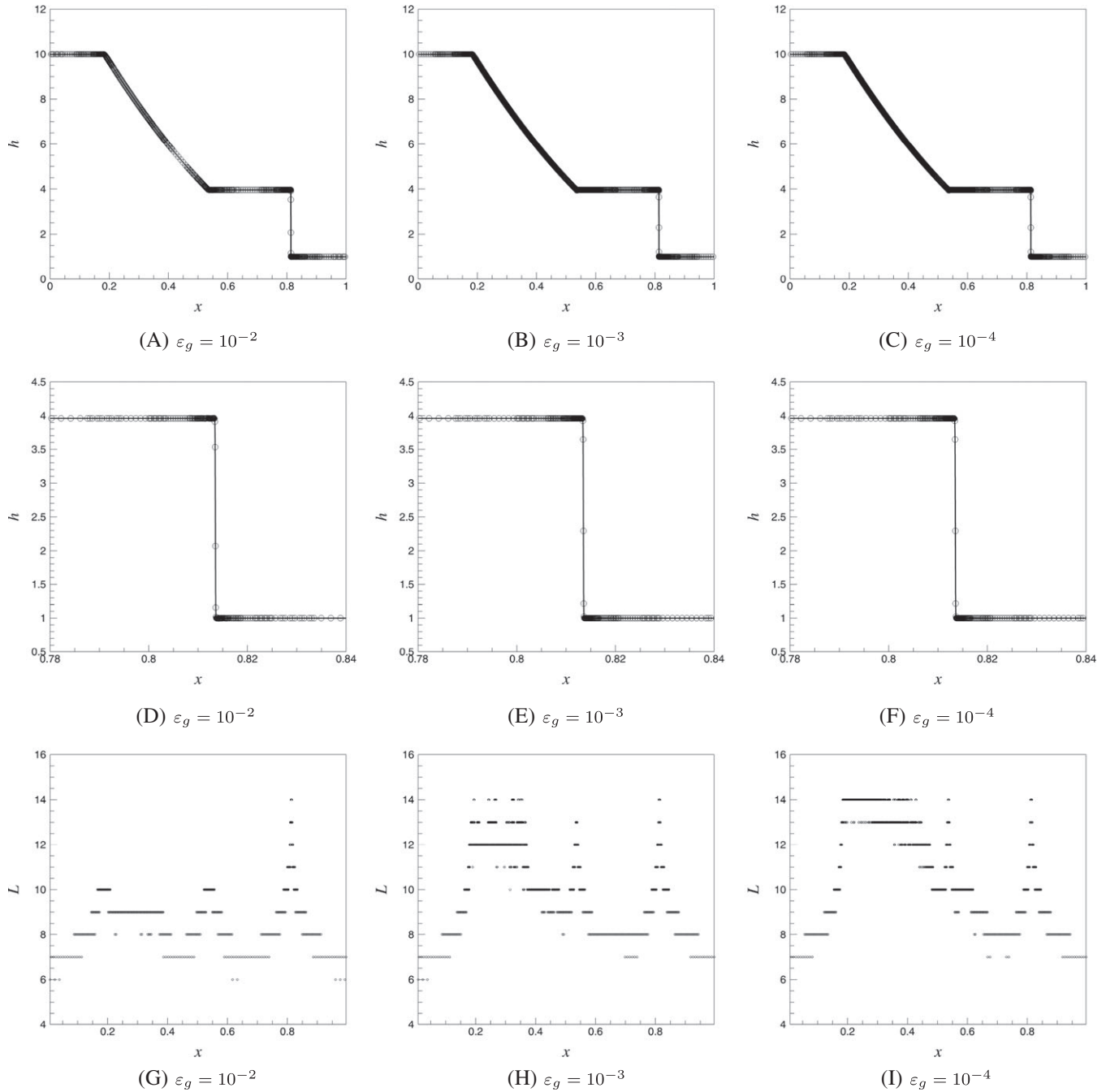


FIGURE 11 A visual representation of increasing refinement due to stricter thresholding. Fields for the dambreak problem at $t = 0.1$ shown above, zoom ins of the moving shock front showing discontinuity resolution performance in the middle, and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65 536 degrees of freedom. A, $\varepsilon_g = 10^{-2}$; B, $\varepsilon_g = 10^{-3}$; C, $\varepsilon_g = 10^{-4}$; D, $\varepsilon_g = 10^{-2}$; E, $\varepsilon_g = 10^{-3}$; F, $\varepsilon_g = 10^{-4}$; G, $\varepsilon_g = 10^{-2}$; H, $\varepsilon_g = 10^{-3}$; I, $\varepsilon_g = 10^{-4}$

compression and computational expense reduction as well. Indeed, such gains have motivated the use of similar adaptive approaches for solving different implementations of the shallow water equations.⁷²

4.4 | Gas dynamics: Sod shock tube

A dynamic characterization of our deviation from analytical solutions may be carried out using the 1-dimensional Euler equations configured to solve Sod's prototype shock tube problem for higher-dimensional gas dynamics applications. In

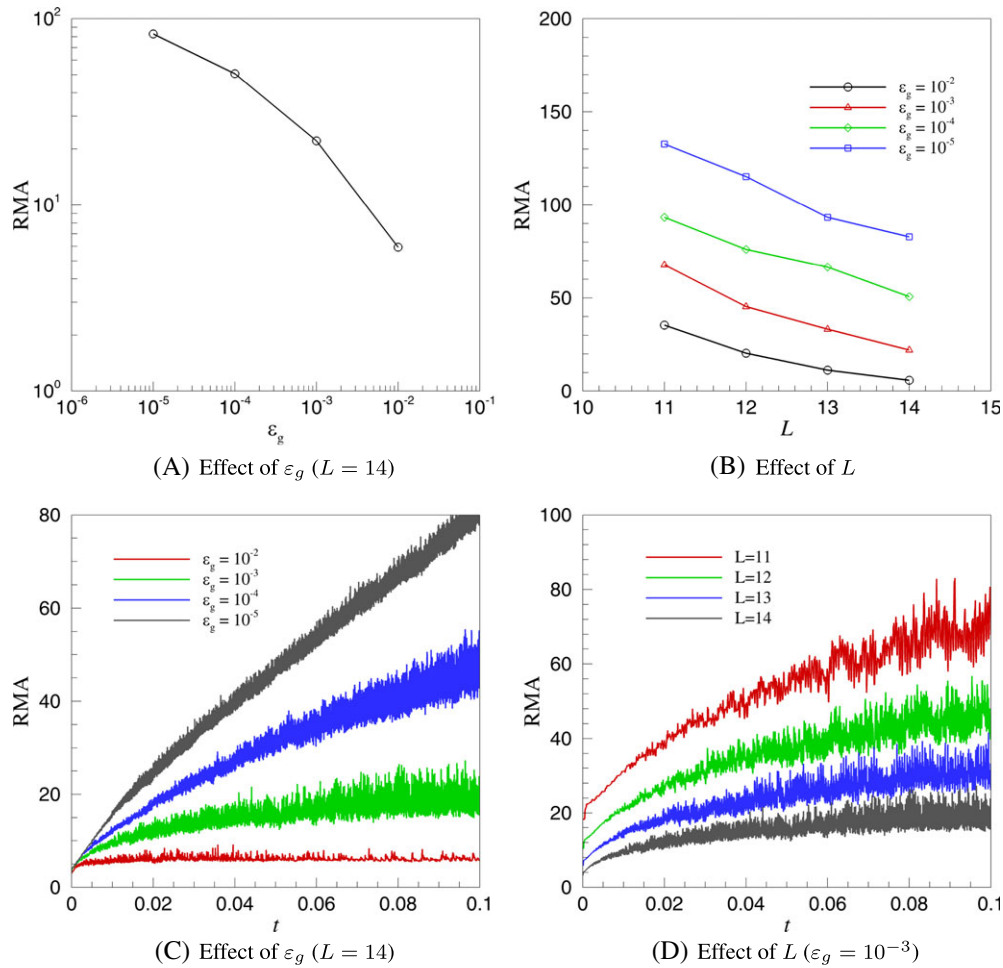


FIGURE 12 The effect of the choice of L and ϵ_g on the relative memory allocation (RMA) of the shallow water equations with time. Higher relative memory allocations for lower L and stricter ϵ_g observed as expected. A, Effect of ϵ_g ($L = 14$); B, Effect of L ; C, Effect of ϵ_g ($L = 14$); D, Effect of L ($\epsilon_g = 10^{-3}$) [Colour figure can be viewed at wileyonlinelibrary.com]

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

CPU Gain	$\epsilon_g=10^{-2}$	$\epsilon_g=10^{-3}$	$\epsilon_g=10^{-4}$	$\epsilon_g=10^{-5}$
$L = 11$	-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

terms of our hyperbolic framework, we have

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}, \quad (54)$$

with an equation of state given by

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho u^2 \right). \quad (55)$$

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

$$(\rho, u, p)_0 = \begin{cases} (1.0, 0.0, 1.0), & \text{for } 0.0 \leq x \leq 0.5 \\ (0.125, 0.0, 0.1), & \text{for } 0.5 < x \leq 1.0. \end{cases} \quad (56)$$

The spectral radius of the aforementioned system can be obtained through

$$r(A) = \max(|u|, |u - a|, |u + a|), \quad (57)$$

where a is the speed of sound given by

$$a = \sqrt{\gamma p / \rho}. \quad (58)$$

Figure (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

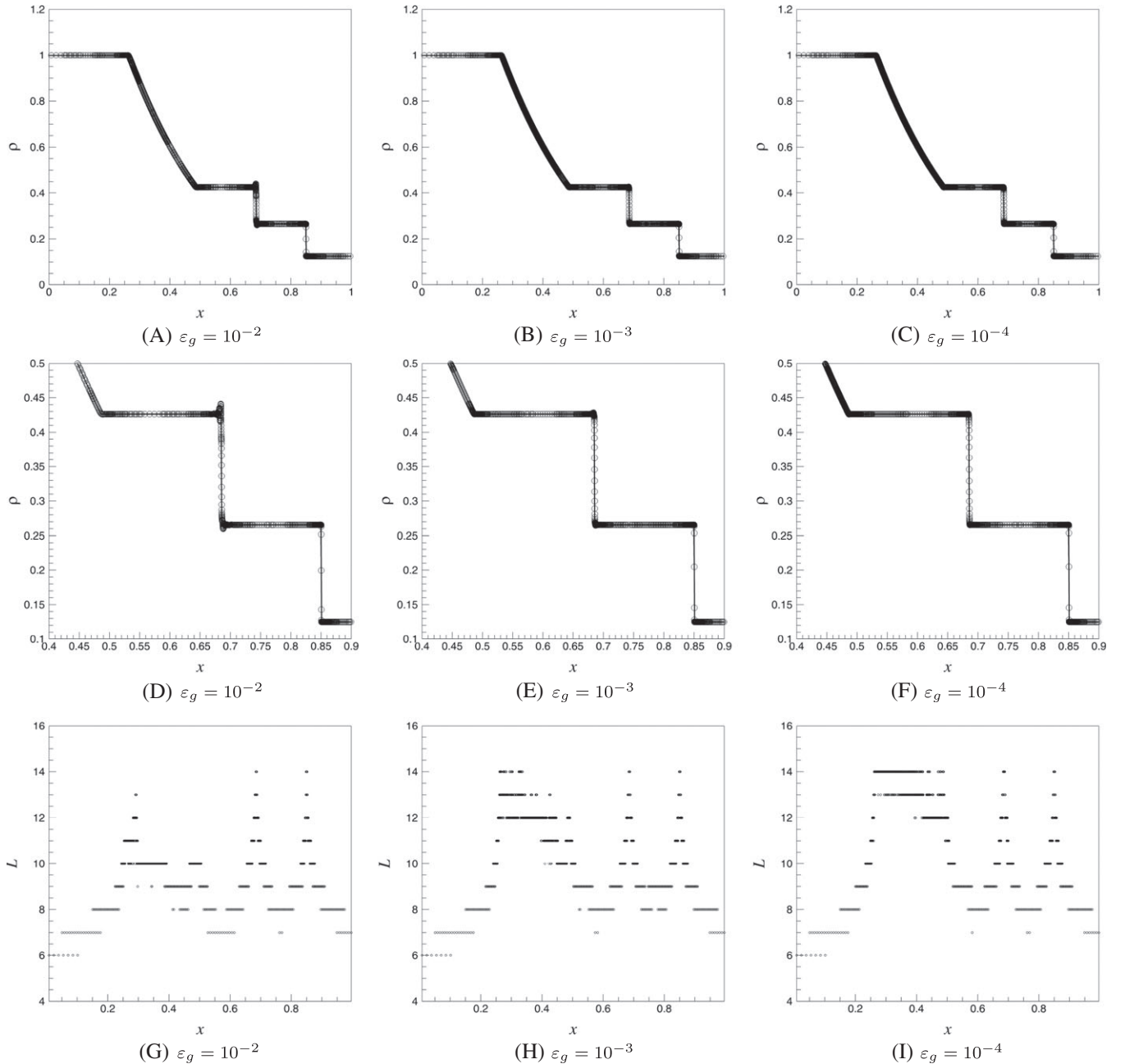


FIGURE 13 A visual representation of increasing refinement due to stricter thresholding. Fields for the shock tube problem at $t = 0.2$ shown above, zoom ins of the shock front showing discontinuity resolution performance in the middle, and corresponding levels of resolution shown below. The solid line is given by the exact solution. A, $\varepsilon_g = 10^{-2}$; B, $\varepsilon_g = 10^{-3}$; C, $\varepsilon_g = 10^{-4}$; D, $\varepsilon_g = 10^{-2}$; E, $\varepsilon_g = 10^{-3}$; F, $\varepsilon_g = 10^{-4}$; G, $\varepsilon_g = 10^{-2}$; H, $\varepsilon_g = 10^{-3}$; I, $\varepsilon_g = 10^{-4}$

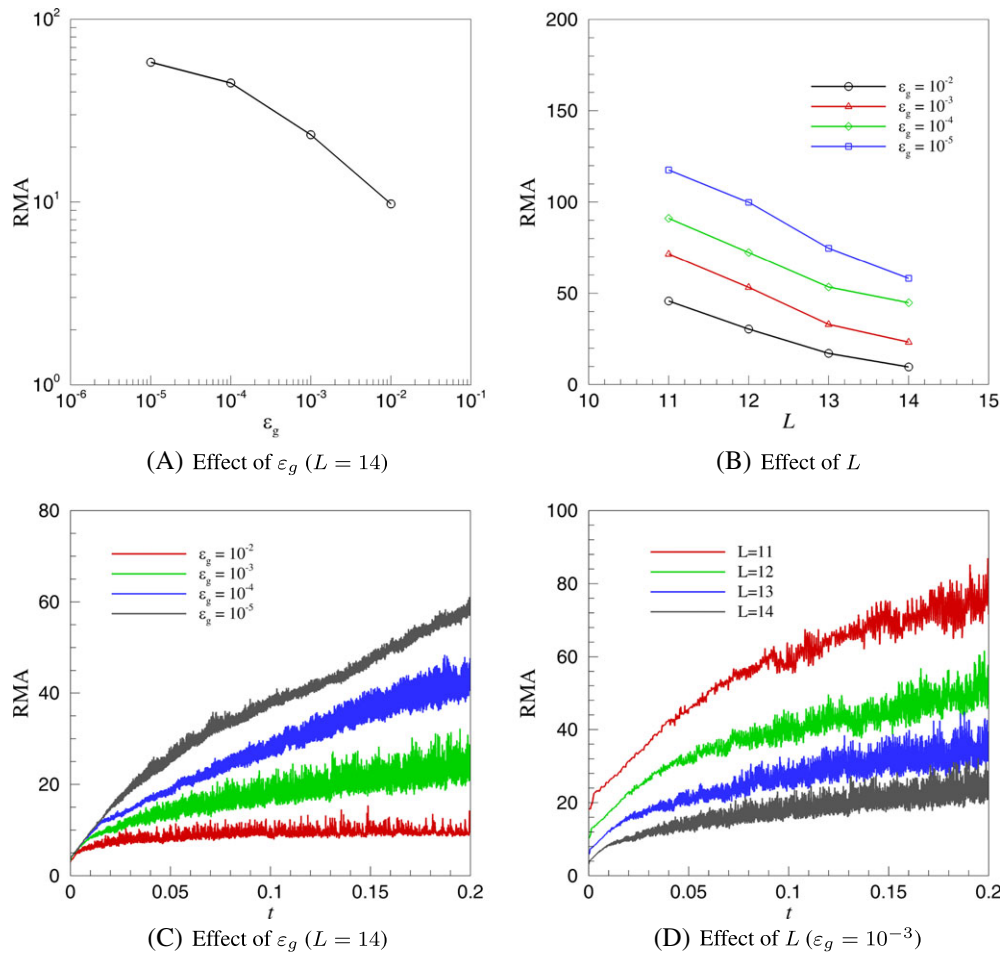


FIGURE 14 The effect of the choice of L and ε_g on the memory gain of the 1-dimensional Euler equations with time. Higher relative memory allocations (RMAs) for lower L and stricter ε_g observed as expected. A, Effect of ε_g ($L = 14$); B, Effect of L ; C, Effect of ε_g ($L = 14$); D, Effect of L ($\varepsilon_g = 10^{-3}$) [Colour figure can be viewed at wileyonlinelibrary.com]

due to the threshold. The case of $\varepsilon_g = 10^{-3}$ performs well in that no large scale oscillations are observed. Figure (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 1-dimensional shocktube problem lets us analyse errors of computation in the proposed framework through an evolution in time. Figure (15) 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

TABLE 3 Computational gain in percentages for our choice of combinations for the 1-dimensional Euler equations. Higher maximum levels of resolution and lower thresholds can be seen to provide greater benefits

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

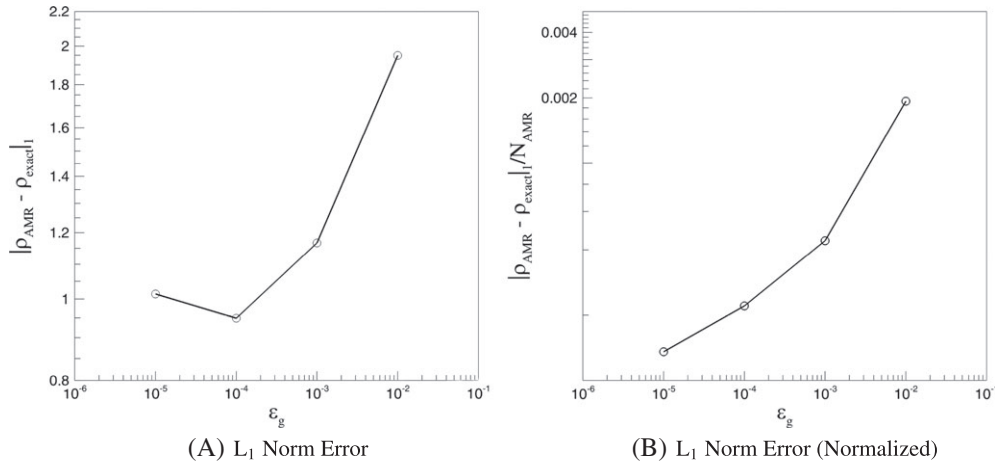


FIGURE 15 Varying L_1 norm error (left) and L_1 norm errors normalized by the total number of points in the computational domain at $t = 0.2$ for the Sod shock tube problem

can observe a steadily decreasing trend for both quantities. However, there is a slight increase in the (unnormalized) 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).

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,⁵² which can be implemented in our framework as

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ B_y \\ B_z \\ E \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p^* - B_x^2 \\ \rho uv - B_x B_y \\ \rho uw - B_x B_z \\ B_y u - B_x v \\ B_z u - B_x w \\ u(E + p^*) - B_x(B_x u + B_y v + B_z w) \end{bmatrix}, \quad (59)$$

where

$$p^* = p + \frac{1}{2} (B_x^2 + B_y^2 + B_z^2) \quad (60)$$

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

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2 + w^2) - \frac{1}{2} (B_x^2 + B_y^2 + B_z^2) \right). \quad (61)$$

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

$$(\rho, u, v, w, B_y, B_z, p)_0 = \begin{cases} (1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0), & \text{for } 0.0 \leq x \leq 0.5, \\ (0.125, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0), & \text{for } 0.5 < x \leq 1.0, \end{cases} \quad (62)$$

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

$$r(A) = \max (|u|, |u - c_f|, |u - c_a|, |u - c_s|, |u + c_f|, |u + c_a|, |u + c_s|), \quad (63)$$

where

$$\begin{aligned} c_f &= \sqrt{\frac{1}{2} \left[a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} + \sqrt{\left(a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} \right) - 4a^2 \frac{B_x^2}{\rho}} \right]}, \\ c_s &= \sqrt{\frac{1}{2} \left[a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} - \sqrt{\left(a^2 + \frac{B_x^2 + B_y^2 + B_z^2}{\rho} \right) - 4a^2 \frac{B_x^2}{\rho}} \right]}, \\ c_a &= \sqrt{\frac{B_x^2}{\rho}}, \end{aligned} \quad (64)$$

and the corresponding speed of sound given by

$$a = \sqrt{\gamma p / \rho}. \quad (65)$$

Figure (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 $N_{UG} = 65\,536$

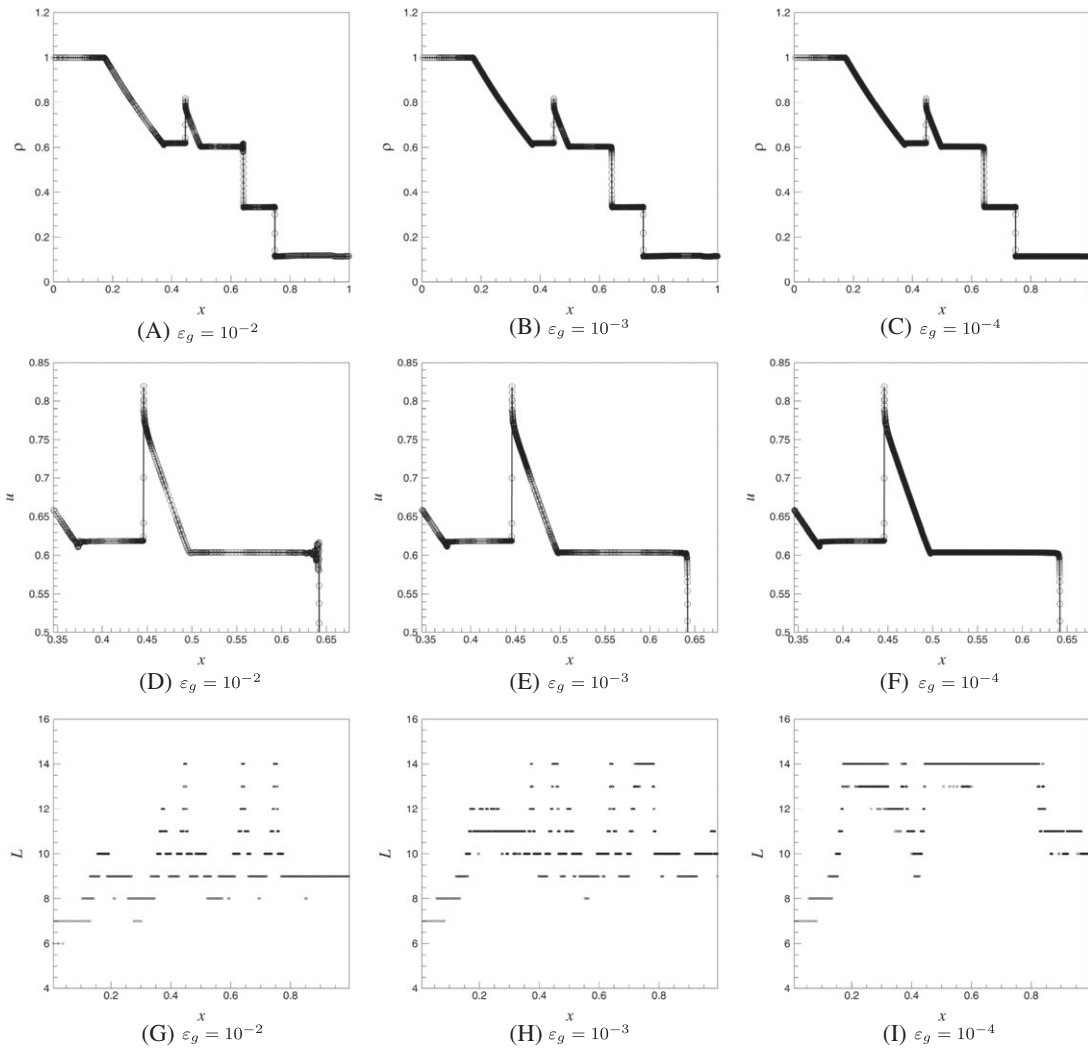


FIGURE 16 A visual representation of increasing refinement due to stricter thresholding. Fields for the Brio-Wu shock tube problem at $t = 0.2$ shown above, zoom ins of the shock front showing discontinuity resolution performance in the middle, and corresponding levels of resolution shown below. The solid line is given by a high fidelity uniform grid solution with 65 536 degrees of freedom. A, $\varepsilon_g = 10^{-2}$; B, $\varepsilon_g = 10^{-3}$; C, $\varepsilon_g = 10^{-4}$; D, $\varepsilon_g = 10^{-2}$; E, $\varepsilon_g = 10^{-3}$; F, $\varepsilon_g = 10^{-4}$; G, $\varepsilon_g = 10^{-2}$; H, $\varepsilon_g = 10^{-3}$; I, $\varepsilon_g = 10^{-4}$

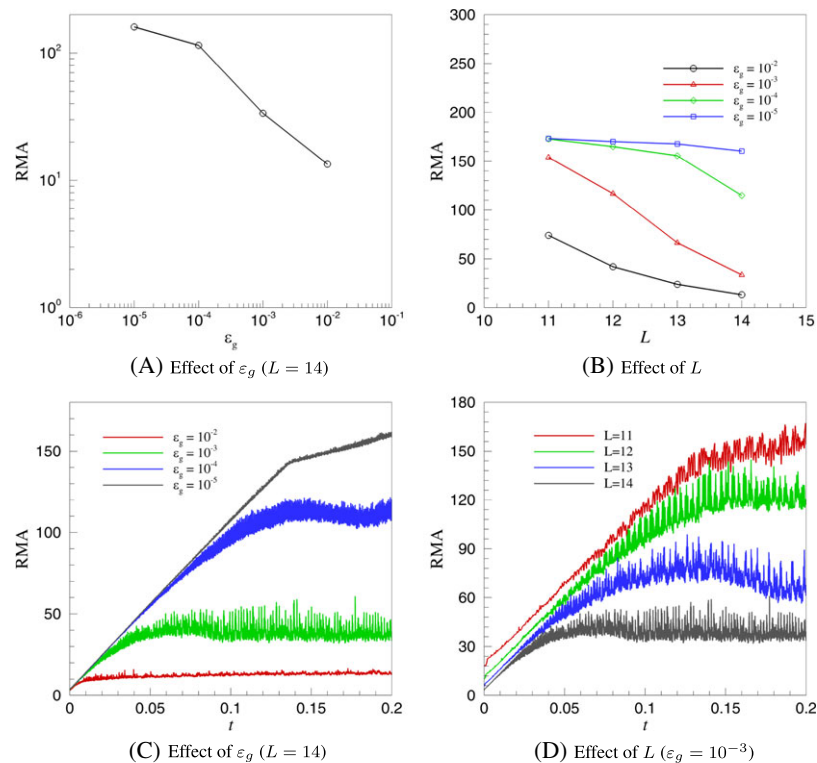


FIGURE 17 The effect of the choice of L and ϵ_g on the memory gain of the 1-dimensional magnetohydrodynamic equations with time. Higher relative memory allocations (RMAs) for lower L and stricter ϵ_g observed as expected. A, Effect of ϵ_g ($L = 14$); B, Effect of L ; C, Effect of ϵ_g ($L = 14$); D, Effect of L ($\epsilon_g = 10^{-3}$) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Computational gain in percentages for our choice of combinations for the 1-dimensional magnetohydrodynamic equations. Higher resolutions and lower thresholds can be seen to provide greater benefits

CPU Gain	$\epsilon_g=10^{-2}$	$\epsilon_g=10^{-3}$	$\epsilon_g=10^{-4}$	$\epsilon_g=10^{-5}$
$L = 11$	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

finite volume cells on a uniform grid) with good memory gain and computational benefits (at $L = 14$). Figure (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 $\epsilon_g = 10^{-3}$ and $L = 14$, we can see a 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.

5 | CONCLUSIONS AND PERSPECTIVES

In this investigation, we have implemented 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 non-linearity of the dissipative stencil of interfacial state reconstructions adaptively, ie, to have smooth regions with linear reconstructions with progressively increasing nonlinearity towards the finest resolutions present in the solution field.

For the purpose of testing this framework, we implement a 1-dimensional system of hyperbolic laws for different physical systems such as the 1-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 7-equation MHD conservation laws in the Brio-Wu shock tube configuration. The interfacial fluxes are reconstructed using the Rusanov Riemann solver, and 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 the discrete wavelet transform 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 that 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 1-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 the work of Roussel et al²⁶ 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.

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 represent a promising technique for procuring an enhanced accuracy at the different scales of a large eddy simulation without compromising on effective dissipation in the vicinity of extremely sharp gradients.

ACKNOWLEDGEMENTS

The computing for this project was performed at the OSU High Performance Computing Center at Oklahoma State University supported in part through the National Science Foundation grant OCI1126330. RM was supported through the Graduate College Summer Research Fellowship, Oklahoma State University, and RB was supported by the Science and Engineering Research Board (SERB), Department of Science and Technology, India, under the Grant No. MTR/2017/000417.

ORCID

O. San  <http://orcid.org/0000-0002-2241-4648>

R. Behera  <http://orcid.org/0000-0002-6237-5700>

REFERENCES

1. Weiss J. Applications of compactly supported wavelets to the numerical solution of partial differential equations. *Photonics Technologies for Robotics, Automation, and Manufacturing, International Society for Optics and Photonics*; 2004:106-117.

2. Whitham G. *Linear and Nonlinear Waves*. Vol 42. New York, NY: John Wiley & Sons; 2011.
3. Fujii M, Hoefer WJ. Interpolating wavelet collocation method of time dependent Maxwells equations: characterization of electrically large optical waveguide discontinuities. *J Comput Phys*. 2003;186(2):666-689.
4. Kumar P, Foufoula-Georgiou E. Wavelet analysis for geophysical applications. *Rev Geophys*. 1997;35(4):385-412.
5. Alam J, Lin J. Toward a fully Lagrangian atmospheric modeling system. *Mon Weather Rev*. 2008;136(12):4653-4667.
6. Luo J, Jameson L. A wavelet-based technique for identifying, labeling, and tracking of ocean eddies. *J Atmos Ocean Tech*. 2002;19(3):381-390.
7. Oruç Ö, Bulut F, Esen A. A Haar wavelet-finite difference hybrid method for the numerical solution of the modified Burgers' equation. *J Math Chem*. 2015;53(7):1592-1607.
8. Michlin T, Polyzou WN, Bulut F. Multiresolution decomposition of quantum field theories using wavelet bases. *Phys Rev D*. 2017;95(9):094501.
9. Bacry E, Mallat S, Papanicolaou G. A wavelet based space-time adaptive numerical method for partial differential equations. *ESAIM Math Model Numer Anal*. 1992;26(7):793-834.
10. Cai W, Wang J. Adaptive multiresolution collocation methods for initial-boundary value problems of nonlinear PDEs. *SIAM J Numer Anal*. 1996;33(3):937-970.
11. Vasilyev OV, Paolucci S. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J Comput Phys*. 1996;125(2):498-512.
12. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J Comput Phys*. 1989;82(1):64-84.
13. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *J Comput Phys*. 1984;53(3):484-512.
14. Sonar T, Süli E. A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. *Numer Math*. 1998;78(4):619-658.
15. Harten A. Discrete multi-resolution analysis and generalized wavelets. *Appl Numer Math*. 1993;12(1-3):153-192.
16. Harten A. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun Pure Appl Math*. 1995;48(12):1305-1342.
17. Harten A. Multiresolution representation of data: a general framework. *SIAM J Numer Anal*. 1996;33(3):1205-1256.
18. Jameson L. A wavelet-optimized, very high order adaptive grid and order numerical method. *SIAM J Sci Comput*. 1998;19(6):1980-2013.
19. Kumar V, Mehra M. Wavelet optimized finite difference method using interpolating wavelets for self-adjoint singularly perturbed problems. *J Comput Appl Math*. 2009;230(2):803-812.
20. Paolucci S, Zikoski Z, Wirasaet D. WAMR: an adaptive wavelet method for the simulation of compressible reacting flow. Part I. Accuracy and efficiency of algorithm. *J Comput Phys*. 2014;272:814-841.
21. Vasilyev OV, Bowman C. Second-generation wavelet collocation method for the solution of partial differential equations. *J Comput Phys*. 2000;165(2):660-693.
22. Liu WK, Chen Y. Wavelet and multiple scale reproducing kernel methods. *Int J Numer Meth Fluids*. 1995;21(10):901-931.
23. Hariharan G, Kannan K. Review of wavelet methods for the solution of reaction-diffusion problems in science and engineering. *Appl Math Model*. 2014;38(3):799-813.
24. Qian S, Weiss J. Wavelets and the numerical solution of partial differential equations. *J Comput Phys*. 1993;106(1):155-175.
25. Bertoluzza S, Naldi G. A wavelet collocation method for the numerical solution of partial differential equations. *Appl Comput Harmon Anal*. 1996;3(1):1-9.
26. Roussel O, Schneider K, Tsigulin A, Bockhorn H. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J Comput Phys*. 2003;188(2):493-523.
27. Alam JM, Walsh RP, Alamgir Hossain M, Rose AM. A computational methodology for two-dimensional fluid flows. *Int J Numer Methods Fluids*. 2014;75(12):835-859.
28. Castro DA, Gomes SM, Stolfi J. High-order adaptive finite-volume schemes in the context of multiresolution analysis for dyadic grids. *Comput Appl Math*. 2016;35(1):1-16.
29. Do S, Li H, Kang M. Wavelet-based adaptation methodology combined with finite difference WENO to solve ideal magnetohydrodynamics. *J Comput Phys*. 2017;339:482-499.
30. Schmidt AA, Kozakevicius ADJ, Jakobsson S. A parallel wavelet adaptive WENO scheme for 2D conservation laws. *Int J Numer Method H*. 2017;27(7):1467-1486.
31. Schneider K, Vasilyev OV. Wavelet methods in computational fluid dynamics. *Annu Rev Fluid Mech*. 2010;42:473-503.
32. Bramkamp F, Lamby P, Müller S. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *J Comput Phys*. 2004;197(2):460-490.
33. Duarte M, Massot M, Descombes S, et al. 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*. 2012;34(1):A76-A104.
34. Domingues MO, Gomes SM, Roussel O, Schneider K. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. *J Comput Phys*. 2008;227(8):3758-3780.
35. Domingues MO, Gomes SM, Roussel O, Schneider K. Space-time adaptive multiresolution methods for hyperbolic conservation laws: applications to compressible Euler equations. *Appl Numer Math*. 2009;59(9):2303-2321.

36. Bihari BL, Harten A. Multiresolution schemes for the numerical solution of 2-D conservation laws I. *SIAM J Sci Comput.* 1997;18(2):315-354.
37. Harten A. Adaptive multiresolution schemes for shock computations. *J Comput Phys.* 1994;115(2):319-338.
38. Sweldens W, Schröder P. Building your own wavelets at home. In: Klees R, Haagmans R, eds. *Wavelets in the Geosciences*. Berlin, Germany: Springer; 2000:72-107.
39. Vasilyev OV. Solving multi-dimensional evolution problems with localized structures using second generation wavelets. *Int J Comput Fluid D.* 2003;17(2):151-168.
40. Sweldens W. The lifting scheme: a construction of second generation wavelets. *SIAM J Math Anal.* 1998;29(2):511-546.
41. Balsara DS, Shu CW. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J Comput Phys.* 2000;160(2):405-452.
42. Grinstein F, Margolin L, Rider W, eds. *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge, UK: Cambridge University Press; 2007.
43. Martín MP, Taylor EM, Wu M, Weirs VG. A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence. *J Comput Phys.* 2006;220(1):270-289.
44. Adams NA, Shariff K. A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems. *J Comput Phys.* 1996;127(1):27-51.
45. Pirozzoli S. Conservative hybrid compact-WENO schemes for shock-turbulence interaction. *J Comput Phys.* 2002;178(1):81-117.
46. Hill DJ, Pullin DI. Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shocks. *J Comput Phys.* 2004;194(2):435-450.
47. Ren YX, Miao'er L, Zhang H. A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws. *J Comput Phys.* 2003;192(2):365-386.
48. Kim D, Kwon JH. A high-order accurate hybrid scheme using a central flux scheme and a WENO scheme for compressible flowfield analysis. *J Comput Phys.* 2005;210(2):554-583.
49. Li H, Do S, Kang M. A wavelet-based adaptive WENO algorithm for Euler equations. *Comput Fluids.* 2015;123:10-22.
50. Cohen A, Kaber S, Müller S, Postel M. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math Comput.* 2003;72(241):183-225.
51. Laney CB. *Computational Gasdynamics*. Cambridge, UK: Cambridge University Press; 1998.
52. Brio M, Wu CC. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J Comput Phys.* 1988;75(2):400-422.
53. Ambrosi D. Approximation of shallow water equations by Roe's Riemann solver. *Int J Numer Meth Fluids.* 1995;20(2):157-168.
54. Gottlieb S, Shu CW. Total variation diminishing Runge-Kutta schemes. *Math Comput.* 1998;67(221):73-85.
55. Liu XD, Osher S, Chan T. Weighted essentially non-oscillatory schemes. *J Comput Phys.* 1994;115(1):200-212.
56. Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. *J Comput Phys.* 1996;126(1):202-228.
57. Henrick AK, Aslam TD, Powers JM. Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *J Comput Phys.* 2005;207(2):542-567.
58. Ha Y, Kim C, Lee Y, Yoon J. An improved weighted essentially non-oscillatory scheme with a new smoothness indicator. *J Comput Phys.* 2013;232(1):68-86.
59. Kim CH, Ha YJ, Yoon J. Modified non-linear weights for fifth-order weighted essentially non-oscillatory schemes. *J Sci Comput.* 2016;67(1):299-323.
60. Huang C. WENO scheme with new smoothness indicator for Hamilton-Jacobi equation. *Appl Math Comput.* 2016;290:21-32.
61. Balsara DS, Garain S, Shu CW. An efficient class of WENO schemes with adaptive order. *J Comput Phys.* 2016;326:780-804.
62. Latini M, Schilling O, Don WS. Effects of WENO flux reconstruction order and spatial resolution on reshocked two-dimensional Richtmyer-Meshkov instability. *J Comput Phys.* 2007;221(2):805-836.
63. Rusanov VV. The calculation of the interaction of non-stationary shock waves with barriers. *Vysisl Mat I Mat Fiz.* 1961;1(2):267-279.
64. Mallat SG. Multiresolution approximation approximations and wavelet orthonormal bases of $L^2(R)$. *Trans Amer Math Soc.* 1989;315(1):69-87.
65. Meyer Y. *Analysis at Urbana I: Analysis in Function Spaces*. Cambridge, UK: Cambridge University Press; 1989.
66. Daubechies I. Orthonormal bases of compactly supported wavelets. *Commun Pure Appl Math.* 1988;41(7):909-996.
67. Strang G. Wavelets and dilation equations: a brief introduction. *SIAM Rev.* 1989;31(4):614-627.
68. Cohen A. Wavelet methods in numerical analysis. *Handb Numer Anal.* 2000;7:417-711.
69. Deiterding R, Domingues MO, Gomes SM, Roussel O, Schneider K. Adaptive multiresolution or adaptive mesh refinement? A case study for 2D Euler equations. *ESAIM: Proc.* 2009;29:28-42.
70. Tenaud C, Duarte M. Tutorials on adaptive multiresolution for mesh refinement applied to fluid dynamics and reactive media problems. *ESAIM: Proc.* 2011;34:184-239.
71. Baeza A, Martínez-Gavara A, Mulet P. Adaptation based on interpolation errors for high order mesh refinement methods applied to conservation laws. *Appl Numer Math.* 2012;62(4):278-296.

72. Reckinger SM, Vasilyev OV, Fox-Kemper B. Adaptive wavelet collocation method on the shallow water model. *J Comput Phys.* 2014;271:342-359.

How to cite this article: Maulik R, San O, Behera R. An adaptive multilevel wavelet framework for scale-selective WENO reconstruction schemes. *Int J Numer Meth Fluids.* 2018;87:239-269. <https://doi.org/10.1002/fld.4489>

APPENDIX

```

class Cell
{
public:
    double q[n_eq] = {0.0};    //Array for conserved variables
    int i;                      //Integer identifier

    double det[n_eq]={0.0};    //Wavelet coefficients

    int level;                  //Resolution
    double xx=0.0;              //Location
    int keep_flag = 1;          //Flag for cell retention
    double cell_length=0.0;
    int leaf = 0;               //if 1->leaf
    int new_cell = 0;           //if 1->this is a newly formed cell

    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
    Cell* right_level=nullptr;  //Pointer to right neighbor

    int virt = 0;               //Flag for virtual leaf

    //Fluxes
    double f_p[n_eq] = {0.0};  //Right face flux
    double f_m[n_eq] = {0.0};  //Left face flux

    //Runge Kutta storage
    double q_rhs[n_eq] = {0.0};
    double q_temp[n_eq] = {0.0};
    double q_1[n_eq] = {0.0};
    double q_2[n_eq] = {0.0};

    //Source Term
    double q_source[n_eq] = {0.0}; //If needed

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

```

```

Cell* find_cell(int i, unordered_map<int, Cell*> &map) //Use after
    verifying that cell exists
{
    int child_tracker[max_level];    //Setting up path array
    int level;

    if (i==1)                        //Root always remains in map
    {
        Cell* loc = map[1];
        return loc;
    }

    //Tracing unique path to cell index from root
    for (level=0; level<max_level; level++)
    {
        if (i
        {
            child_tracker[level]=1;
        }
        else if (i
        {
            child_tracker[level]=2;
        }

        i = i/2;

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

    //Following path to cell location in memory
    Cell* loc = map[1];    //Starting at root
    do{
        if (child_tracker[level]==1) //left child
        {
            loc = loc->left_child;
        }
        else if (child_tracker[level]==2) //right child
        {
            loc = loc->right_child;
        }
        level--;
    } while (level>=0);

    return loc;    //returning location of desired cell
}

```