



# WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. Part I. Accuracy and efficiency of algorithm

Samuel Paolucci\*, Zachary J. Zikoski, Damrongsak Wirasaet

University of Notre Dame, Notre Dame, IN 46556, United States

## ARTICLE INFO

### Article history:

Received 20 August 2012

Received in revised form 8 October 2013

Accepted 16 January 2014

Available online 6 February 2014

### Keywords:

Wavelet

Adaptive

Compressible reacting flow

Newtonian fluid

Detailed kinetics

Multicomponent diffusion

## ABSTRACT

The Wavelet Adaptive Multiresolution Representation (WAMR) method provides a robust method for controlling spatial grid adaptation — fine grid spacing in regions where a solution varies greatly (i.e., near steep gradients, or near-singularities) and a much coarser grid where the solution varies slowly. Subsequently, a wide range of spatial scales, often demanded in challenging continuum physics problems, can be efficiently captured. Furthermore, the wavelet transform provides a direct measure of local error at each collocation point, effectively producing automatically verified solutions.

The method is applied to the solution of unsteady, compressible, reactive flow equations, and includes detailed diffusive transport and chemical kinetics models. Accuracy and performance of the method are examined on several test problems. The sparse grids produced by the WAMR method exhibit an impressive compression of the solution, reducing the number of collocation points used by factors of many orders of magnitude when compared to uniform grids of equivalent resolution.

© 2014 Elsevier Inc. All rights reserved.

## 1. Introduction

For numerical simulations to be used as truly predictive tools, solutions of computational physics models must adequately capture all relevant scales embodied in them in an accurate and consistent manner. However, this becomes particularly challenging for problems having a strong multiscale character, where spatial and temporal scales may span several orders of magnitude. Such multiscale problems are encountered in a number of fields of practical interest such as astrophysics, material sciences, meteorology, and combustion.

According to a 2006 Department of Energy Office of Basic Energy Sciences' workshop [1], combustion science has a single grand challenge — the development of a “validated, predictive, multiscale combustion modeling capability to optimize the design and operation of evolving fuels in advanced engines used for transportation.” Simulation of combustion is particularly challenging due to its typically multidimensional nature and the wide range of spatial and temporal scales present. In addition to the challenges associated with chemical reactions, this work is primarily concerned with the simulation of *high-speed* reactive flows, where compressibility effects and the presence of phenomena such as shock waves is important. The intricate coupling between the fluid mechanics and chemistry results in detonations being particularly difficult to simulate accurately. For detailed chemistry models in a hydrogen/oxygen mixture, the smallest spatial scales have been estimated to

\* Corresponding author.

E-mail addresses: paolucci@nd.edu (S. Paolucci), zzikoski@alumni.nd.edu (Z.J. Zikoski), dwirasae@nd.edu (D. Wirasaet).

be of the order of  $0.1 \mu\text{m}$  [2]. This estimate is based purely on the resolution required to capture reaction zone structures, and is of the same order as the mean free path for the mixture. Diffusive effects, like chemical reactions, occur on molecular scales, and the resolution required to capture viscous effects would be of a similar size. Thus, spatial scales span seven or more orders of magnitude. Using a standard finite difference approximation on a uniform grid would require  $O(10^{7d})$  grid points for a simulation in  $d$  spatial dimensions. Similarly, time scales can range from those associated with chemical reactions (nano-seconds) up to macro time scales (of the order of seconds). Such physical/chemical demands require the use of advanced computational approaches along with modern computational resources.

These multiscale problems are generally impractical to solve (in terms of computer time and memory required) on a fixed grid. However, in many problems of practical interest, small scales only occur in limited regions of the computational domain. Adaptive methods take advantage of this general knowledge by providing fine resolution only in areas where it is needed and coarser resolution elsewhere. There are a number of adaptive methods which have been developed and applied to the solution of partial differential equations (PDEs), including adaptive mesh refinement (AMR), finite element methods (FEM), multigrid, and wavelet methods.

One of the earliest and most popular method is AMR, originally proposed by Berger and Oliger [3]. In AMR, the domain is composed of cells in an initially coarse grid. Individual cells are then flagged for adaption, according to criteria depending on the local solution. Subsequently, a great deal of computational savings can be realized from two sources: the first is the reduction of the total number of grid points or cells required in the solution, and second, all grid points are advanced according to the time step based on the level of refinement.

AMR schemes require a criterion for controlling grid refinement. Originally, Richardson extrapolation was suggested as a means to estimate the error between grid resolutions. In this way, an error tolerance could be set, and where the local error estimate exceeded the tolerance, those cells would be refined. However, Richardson extrapolation is only accurate in regions where the solution is sufficiently smooth; the error estimate is not accurate across steep gradients or near-discontinuities, which is where refinement would be expected. Heuristic methods are also commonly applied, using gradients of the solution to indicate where refinement should occur. Because the actual error introduced by AMR is not well understood, the effectiveness of an adaption criteria requires *a posteriori* error estimates [4].

In FEM the solution in each element is represented by combinations of basis functions, which may take a variety of forms, though polynomials are common. The representation of the solution inside elements provides additional options for adaptivity, beyond mesh refinement. Adaptive FEM may contain any combination of three main types of adaptivity:  $h$ -,  $p$ -, and  $r$ -adaptation [5]. In  $h$ -adaptation, individual elements may be subdivided into several smaller elements, improving mesh resolution locally. An initially coarse mesh is recursively refined according a refinement criterion or local error estimate [6]. In  $p$ -adaptivity, solution accuracy is improved by increasing the order of the basis functions used in an element. Using  $p$ -adaptation proceeds from a fixed mesh and using low-order basis functions. Gui and Babuska develop much of the theory behind  $h$ - and  $p$ -adaptation, as well as the combination of the two  $hp$ -adaptation schemes in [7] and [8]. It has been found that the combination of geometric refinement and higher-order basis functions in  $hp$ -adaptation can result in spectral convergence rates, at the cost of a more complex stiffness matrix and subsequent implementation [9]. The third type of adaptivity found in FEM is  $r$ -adaptation, the movement or relocation of elements. This type of adaptation iteratively relocates elements to minimize discretization errors while maintaining a fixed number of degrees of freedom. Such adaptivity is also often coupled with  $h$ -adaptation [10], though perhaps not explicitly identified in general  $h$ - and  $hp$ -adaptation methods.

Criteria used to control the various methods of adaptation in FEM are, like in the case of AMR, limited generally to *a posteriori* or heuristic methods [11–13]. For hyperbolic or parabolic problems, iterative creation of the mesh would be quite costly, as features may evolve and move throughout the domain [14].

Multigrid methods use the residual of the solution to determine where refinement is needed. In this manner, a solution to a discretized PDE can be found to a prescribed error tolerance, while reducing the total number of computations needed [15,16]. Being a relaxation method, adaptive multigrid methods are typically used to solve elliptic problems. However, an adaptive multigrid method has been applied to the solution of the Euler equations by using a space–time discretization [17].

Wavelets are mathematical functions with compact support in both location and scale. Wavelet representation of signals have been noted for their compression properties, and have found significant use in signal processing and data analysis. In the adaptive solution of PDEs, wavelets have been applied in a number of methods such as Galerkin formulations, collocation, and finite-difference methods [18].

Wavelets were first applied as basis functions in Galerkin formulations, using thresholding of wavelet amplitudes to substantially reduce the number of degrees of freedom [19]. However, treatment of non-linear terms and evaluations of integrals of wavelet values arising in the method offset potential reductions in execution time. To improve computational efficiency, techniques borrowed from pseudo-spectral methods are used, where nonlinear terms are computed in real space and transformed into wavelet space. Liandrat et al. [20] develop a dynamically adaptive method for unsteady problems using a Petrov–Galerkin formulation. Many approaches have been developed for handling irregular domains and non-periodic boundary conditions [21,22].

Wavelet collocation methods solve for function values at collocation points rather than for wavelet amplitudes as in wavelet-Galerkin methods. This difference circumvents many of the difficulties associated with non-linear terms. Many of the collocation methods also require far fewer operations per collocation point for both linear and non-linear terms, as well as allow the use of general boundary conditions on finite domains [23,24]. Dynamically adaptive algorithms for initial value

problems use the sparse set of collocation points at a previous time step to generate an updated set of points for the next time step, effectively tracking fine resolution features as they develop in the domain [25].

Wavelet finite-difference methods use wavelet transforms of intermediate solutions to determine where collocation points are needed, and use finite-difference approximations on the sparse grid. Application of finite-differences requires much less computational effort, as well as making application of boundary conditions trivial. This approach also makes use of fast wavelet transforms for mapping functional values to wavelet amplitudes and vice-versa, by taking advantage of the relationship between interpolating wavelets and the interpolating subdivision scheme.

Wavelet methods are particularly well-suited to adaptive solution of PDEs. Wavelet amplitudes reflect the local regularity of a transformed function. Wavelet amplitudes will be large in rapidly varying regions and small where the solution is well-behaved. By thresholding the wavelet transform and omitting points with small amplitudes, a sparse grid can be generated which can be dynamically adapted to evolving features of a solution. Wavelets provide a methodology for controlling grid adaptivity without requiring *ad hoc* error estimates or heuristics.

A number of multiresolution methods, some of which utilize the wavelet multiscale decomposition, have been developed and demonstrated on a variety of test problems in one- and two-dimensions, such as the heat equation, Burgers' equation [26,27], wave equation, reaction–diffusion equation [28], Euler equation [29], as well as various boundary-value problems. The wavelet method of Rastigejev et al. [30,31] is used to simulate a one-dimensional, viscous, detonation in a hydrogen–oxygen–argon mixture using detailed chemistry. The same method is also applied to two- and three-dimensional simulations of incompressible flow, including the lid-driven cavity and differentially-heated cavity problems. The method is also used to simulate incompressible flow around a tandem of cylinders in two-dimensions using a Brinkman penalization method to represent solid obstacles [32]. Griebel and Koster [33] use a variation of a Petrov–Galerkin and a finite-difference formulation to solve the incompressible Navier–Stokes equations in two-dimensions for a vortex interaction problem and in both two- and three-dimensions for a mixing layer. A wavelet-collocation method using a lifting scheme is applied to simulations of a two-dimensional flame-vortex interaction, and flow past a cylinder using a penalization method [34], as well as extended for use on elliptic problems [35]. Reckinger et al. [36] apply the method to simulating Rayleigh–Taylor instabilities. Farge [37] discusses the use of wavelets for analyzing and modeling turbulence, and Farge et al. [38,39] use a wavelet basis in a Petrov–Galerkin method to simulate turbulent fields.

Though this work is focused on the simulation of reactive flows, the method developed here is equally applicable to many similar multiscale problems encountered in other disciplines. The amplitudes in the wavelet transform used in the method provide a direct measure of the local error at each grid point. Subsequently, spatial grid adaption can be driven by removing points with small wavelet amplitudes resulting in a minimal loss of accuracy while greatly reducing the number of unknowns. Coupled with the efficient use of widely available distributed memory parallel architectures, the method provides a tool for producing solutions of multidimensional problems using an adaptive verified numerical simulation (VNS).

In this first part of the work, we critically examines many detailed aspects of the adaptive wavelet-based collocation algorithm and its accuracy for the simulation of compressible, reactive flows. Section 2 describes the general set of governing equations used for reactive, compressible flow problems, and the general form of common initial and boundary conditions used. Section 3 describes the numerical method and provides implementation algorithms. In this section we also demonstrate the convergence properties of the method through error analyses by utilizing the method of manufactured solutions. In Section 4 we discuss the roles that real and numerical diffusions play in obtaining accurate solutions. In Section 5 specific initial and boundary conditions are given along with the description of problems that are used for code verification. This is followed by Section 6 where we demonstrate the algorithmic performance of the WAMR method, and Section 7 where we make concluding remarks. In the second part [40] we discuss additional computational aspects necessary for the development of an efficient parallel implementation of the method.

The present work is generally distinct in that 1) we solve balance equations that are fairly complete and faithful to continuum physics and chemistry at all scales, and 2) such equations are solved using an adaptive multiscale wavelet collocation scheme for which i) we provide specific algorithms associated with required operations and ii) provide evidence of the scheme's ability to provide verified solutions.

## 2. Governing equations

The numerical simulation of physical problems consists of a set of governing equations which model the physics and chemistry, along with associated initial and boundary conditions, and the numerical method used. These equations are complemented by constitutive equations that include detailed multicomponent diffusion, Sorét and Dufour effects, state dependent transport properties, and detailed chemical kinetics.

### 2.1. Balance equations

The unsteady, compressible, reactive balance equations for mass, momentum, and energy are used to model flows in this work. The system for a  $d$ -dimensional problem of a  $K$ -component mixture in conservation form is

$$\frac{\partial \rho}{\partial t} = - \frac{\partial}{\partial x_j} (\rho u_j), \quad (1)$$

$$\frac{\partial \rho u_i}{\partial t} = -\frac{\partial}{\partial x_j}(\rho u_i u_j + p \delta_{ij}) + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2)$$

$$\frac{\partial \rho E}{\partial t} = -\frac{\partial}{\partial x_j}[(\rho E + p)u_j] - \frac{\partial}{\partial x_j}(q_j - u_i \tau_{ij}), \quad (3)$$

$$\frac{\partial \rho Y_k}{\partial t} = -\frac{\partial}{\partial x_j}(\rho u_j Y_k) - \frac{\partial j_{j,k}}{\partial x_j} + M_k \dot{\omega}_k, \quad k = 1, \dots, K-1. \quad (4)$$

This set of equations describes conservation of mass (1), linear momentum (2), and total energy (3), and includes  $K-1$  evolution equations for species' partial densities (4). The independent variables are the spatial coordinates  $x_j$ ,  $j = 1, \dots, d$  and time  $t$ ,  $\delta_{ij}$  is the Kronecker delta, and the dependent variables in primitive form are: density  $\rho$ , velocity vector component  $u_i$ , pressure  $p$ , total specific energy  $E$ , and mass fraction of the  $k$ -th specie  $Y_k$ . The total specific energy is defined as  $E = e + \frac{1}{2}u_i u_i$ , where  $e$  is the specific internal energy. The sum of species partial densities must equal the mixture density, so the system is also subject to the constraint

$$\sum_{k=1}^K Y_k = 1, \quad (5)$$

which is used to determine the mass fraction of the remaining specie, typically chosen to be the largest initial mass fraction – often an inert diluent [41].

## 2.2. Constitutive equations

The balance equations are closed by additionally providing equations of state based on thermodynamic relations and constitutive relationships for the viscous stress tensor  $\tau_{ij}$ , the heat flux vector  $q_i$ , the mass flux vector  $j_{i,k}$ , and the chemical source term  $\dot{\omega}_k$ , providing the net production rate of species  $k$ . Below we provide brief descriptions. Detailed equations are given in Appendix A.

We use thermal and caloric equations of state for a perfect gas mixture obeying Dalton's law.

Often, diffusive terms are neglected using the *a priori* assumption that small scale effects do not significantly affect the solution. In this work, diffusive transport terms are retained and detailed models for mass, momentum, and heat diffusions are used. We use the viscous stress tensor  $\tau_{ij}$  for a Newtonian fluid with shear viscosity and the bulk viscosity of the mixture that are generally functions of temperature and composition. The heat flux vector  $q_i$  contains a standard Fourier's law term and a Sorét effect term accounting for energy transport due to mass diffusion. The species mass flux vector  $j_{i,k}$  contains ordinary multicomponent mass diffusion and mass transport due to thermal gradients, known as the DuFour effect. The multicomponent mass diffusion allows for species to diffuse to varying extents into other species due to both concentration gradients, similar to Fick's law, and pressure gradients.

Similar to the diffusive terms in the previous section, often the reaction model is simplified to ease computational demands. Reduced models with one- or two-step kinetics are used and contain free parameters which are tuned to match specific physical properties of interest, such as detonation velocities, heat release, etc. However, the quantitative results of the simulation can depend strongly on the dynamics of the chemical kinetics model, which are lost when simplified models are used. The present work uses a detailed chemical kinetics mechanism based on the law of mass action and the Arrhenius form of reaction rate temperature dependence.

## 3. Wavelet Adaptive Multiresolution Representation (WAMR)

Consider the set of spaces  $\{V_j\}_{j \in \mathbb{Z}} \in L_2(\mathbb{R})$ , where the spaces  $V_j$  are nested,  $V_j \subset V_{j+1}$ . Each space may then be decomposed as  $V_{j+1} = V_j \oplus W_j$ , where  $W_j$  is the complement of  $V_j$  in  $V_{j+1}$ . Applying the decomposition recursively to  $V_J$  up to some coarsest scale  $J_0$  provides the wavelet decomposition of  $V_J$ . We denote bases of the spaces  $V_j$  as *scaling functions*,  $\phi_{j,k}$ , and bases of  $W_j$  as *wavelets*,  $\psi_{j,k}$  [42]. The scaling functions and wavelets are translations and dilatations of special functions,  $\phi(x)$  and  $\psi(x)$ . This work uses a special scaling function  $\phi(x)$  which is the autocorrelation of the Daubechies scaling function  $\varphi_p(x)$  [43]. If  $\varphi_p(x)$  is of order  $p$  ( $p \in \mathbb{N}$ , even,  $p$  coefficients in the dilatation equation and corresponding to  $p/2$  vanishing moments) then  $\phi(x)$  is of order  $p$ . Additionally, linear combinations of the scaling functions can exactly reproduce polynomials up to order  $p-1$  [44–49]. The special wavelets are defined as dilatations and translations of our special scaling function,  $\psi(x) = \phi(2x-1)$ , with modified boundary wavelets used near the edges of a finite domain. For this wavelet basis, the algorithm for finding wavelet expansion coefficients reduces to the use of simple Lagrange interpolation [44,46].

Using these bases, the one-dimensional wavelet transform of a continuous function  $f(x)$  has two parameters: a scale level  $j$  and a location  $k$ . Together, these functions form a multiscale basis, representing  $f(x)$  with scaling functions on a coarsest level,  $J_0$ , and contributions from wavelets on finer levels up to level  $J-1$  with wavelet amplitudes  $d_{j,k}$ . Due to the scaling functions satisfying the interpolation property, we use a collocation procedure where the scaling function amplitudes are given by function values at associated spatial locations,  $f_{J_0,k} = f(x_{J_0,k})$ . The coarsest level is set as  $J_0 = 0$ , and a new

parameter  $N_0 \geq p - 1$  is defined as the maximum index of the coarsest grid level. Each basis function is related to a single point in a regular grid  $V_J$  with locations  $\{x_k = k(N_0 2^J)^{-1} : k \in k_J^0\}$ , with  $k_J^0 = \{0, \dots, N_0 2^J\}$ . This creates a dyadic grid with spacing decreasing by a factor of two at each refinement level. More details on the bases and decomposition are given elsewhere [31,32]

The wavelet amplitudes provide a direct measure of the local approximation error at each grid point. By introducing a thresholding parameter  $\varepsilon$ , the wavelet transform of the function can be rewritten, collecting in  $R_\varepsilon^J$  wavelets with amplitudes smaller than the threshold,

$$f^J(x) = f_\varepsilon^J(x) + R_\varepsilon^J(x), \quad (6)$$

where

$$f_\varepsilon^J(x) = \sum_k f_{0,k} \phi_{0,k}(x) + \sum_{j=0}^{J-1} \sum_{\{k: |d_{j,k}| \geq \varepsilon\}} d_{j,k} \psi_{j,k}(x), \quad (7)$$

$$R_\varepsilon^J(x) = \sum_{j=0}^{J-1} \sum_{\{k: |d_{j,k}| < \varepsilon\}} d_{j,k} \psi_{j,k}(x). \quad (8)$$

Discarding  $R_\varepsilon^J$  yields the sparse wavelet representation (SWR) of  $f$ ,  $f_\varepsilon^J$ , with the local approximation error being of the order of  $\varepsilon$  at any point. In this manner a function can be represented using a minimal number of basis functions while maintaining a prescribed accuracy for a resolution  $J$  sufficiently large.

Subsequently, by also discarding the collocation points associated with wavelet amplitudes less than  $\varepsilon$ , an irregular sparse collocation grid  $\mathcal{V}_E$  of *essential* points can be defined

$$\mathcal{V}_E = \left\{ x_{0,k} \oplus \bigcup_{j>0} x_{j,k} : k \in \kappa_j^0 \right\}, \quad \kappa_j^0 = \{k \in k_j : |d_{j,k}| \geq \varepsilon\}. \quad (9)$$

Collocation points on the coarsest level,  $x_{0,k}$ , are associated only with scaling functions and are always in  $\mathcal{V}_E$ . We define  $N_E$  as the total number of essential points in  $\mathcal{V}_E$ :  $N_E = \dim \mathcal{V}_E$ .

There also exists a fast wavelet transform (FWT) on the irregular collocation grid which maps functional values of  $f_\varepsilon^J(x)$  to and from wavelet amplitudes. The FWT consists of hierarchical application of Lagrange interpolating polynomials. Individual wavelet amplitudes are calculated as the difference between the actual value at a point and its interpolated value using only information on the next coarser grid level,

$$d_{j,k} = f_{j,k} - \sum_{r=1}^p h_r f_{j,k_r^j}, \quad (10)$$

where  $h_r$  are the interpolation coefficients and the  $\varepsilon$  and  $J$  subscript and superscript are suppressed for clarity of explanations [32]. Details regarding the FWT, its inverse, as well as the computation of  $h_r$  are given elsewhere [31]. To perform the adaptive fast wavelet transform (AFWT) and its inverse requires each point to have complete interpolation stencils, which we call a *minimum index set*. However, the sparse grid generated from the SWR will often not meet the minimum index set requirements, so the sparse grid of essential points is expanded to include *non-essential* points which are required for performing interpolations. Non-essential points are assigned wavelet amplitudes of zero, which is equivalent to a purely interpolated value.

The one-dimensional interpolating wavelet and AFWT are easily extended to multiple dimensions. The multidimensional basis functions are formed from tensor products of the one-dimensional scaling functions and wavelets. The location index  $\mathbf{k} \in \mathbf{k}_j = \{0, \dots, N_0 2^j\}^d$  is now a  $d$ -component vector. The base grid dimensions  $\mathbf{N}_0$  is also a  $d$ -component vector with possibly different values in each direction. An additional parameter,  $\mathbf{e} \in \{0, 1\}^d \setminus \mathbf{0}$ , identifies the basis type. Each collocation point has a scale associated with its index in each coordinate direction, and as in the one-dimensional case, is said to belong to the minimum grid level on which the point first exists – identical to the maximum of the scale levels associated with each index. Since points are associated with varying resolution levels in each direction, the basis functions also reflect this difference, with wavelets in directions with the maximum level, and scaling functions in directions at coarser levels. The parameter  $\mathbf{e}$  reflects this distinction, with  $e_i = 1$  for each direction on the maximum grid level, and 0 otherwise.

The AFWT in  $d$ -dimensions consists of  $d$  applications of the one-dimensional transform. Identical to the one-dimensional case, wavelets and their associated collocation points are discarded to generate a sparse irregular grid of essential points  $\mathcal{V}_E$ , the sparse grid is enlarged by adding non-essential points needed to satisfy the minimum index set. Algorithm 1 shows the pseudocode for performing the forward AFWT on an irregular grid. Note that the wavelet amplitudes are written into the same memory location as the function values; the inverse AFWT uses the same algorithm with  $j$  proceeding from 0 up to  $J - 1$ , and the subtraction in line 8 becoming an addition.

**Algorithm 1** AFWT.

---

```

1: Given function values  $\{f_{j,k}\}_{k \in \kappa_j^0}$ 
2: Require:  $\{\mathbf{x}_{j,k}\}_{k \in \kappa_j^0}$  be a grid with minimum index set
3: for  $j = J - 1$  down to 0 do
4:   for  $n = 1$  to  $d$  do
5:     for all  $(\mathbf{e}, \mathbf{k}) \in \mathcal{V}_j$  such that  $e_n = 1$  do
6:        $\mathbf{q} = \mathbf{k}$ 
7:       for  $r = 1$  to  $p$  do
8:          $f_{j,k} = f_{j,k} - h_r f_{j,q}$ 
9:       end for
10:    end for
11:  end for
12: end for

```

---

**Algorithm 2** dfill( $\mathbf{x}_{j,k}$ ),  $j \geq 0$ .

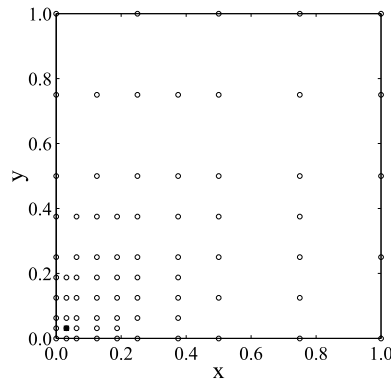
---

```

1: Given point  $\mathbf{x}_{j,k}$ ,  $j \geq 0$ 
2: for  $n = 1$  to  $d$  do
3:   if  $e_n = 1$  then
4:      $\mathbf{q} = \mathbf{k}$ 
5:     for  $r = 1$  to  $p$  do
6:        $q_n = k_r^j$ 
7:       if  $\mathbf{x}_{j,q} \notin \mathcal{V}_j$  then
8:         Enlarge grid  $\mathcal{V}_j = \mathcal{V}_j \cup \{\mathbf{x}_{j,q}\}$ 
9:          $q_n = k_r^j$ 
10:      end if
11:      if  $\mathbf{x}_{j,q} \notin \mathcal{V}_0$  then
12:        call dfill( $\mathbf{x}_{j,q}$ )
13:      end if
14:    end for
15:  end if
16: end for

```

---



**Fig. 1.** The irregular grid ( $p = 4$ ,  $\mathbf{N}_0 = \{4, 4\}$ ), satisfying the minimum index set condition, obtained from the augmentation of the grid by the single point  $\mathbf{x}_{2,(1,1)}$ : ■ represents  $\mathbf{x}_{2,(1,1)}$  and ○ denote augmented points.

An irregular collocation grid  $\mathcal{V}_j$  is said to fulfill the minimum index set if, for every grid point  $\mathbf{x}_{j,k} \in \mathcal{V}_j$ , the points

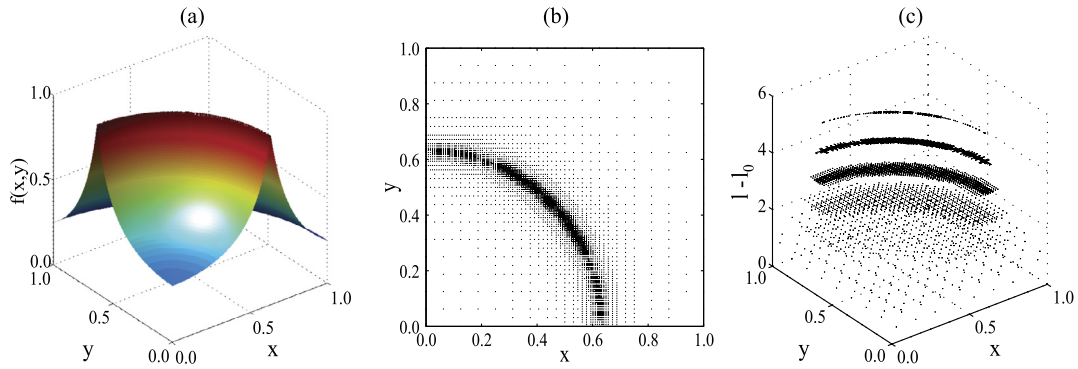
$$\mathbf{x} = (x_1, \dots, x_d) \in \mathbf{X}_{j,k}^{\mathbf{e}} = X_{j,k_1}^{e_1} \times \dots \times X_{j,k_d}^{e_d}, \quad (11)$$

where

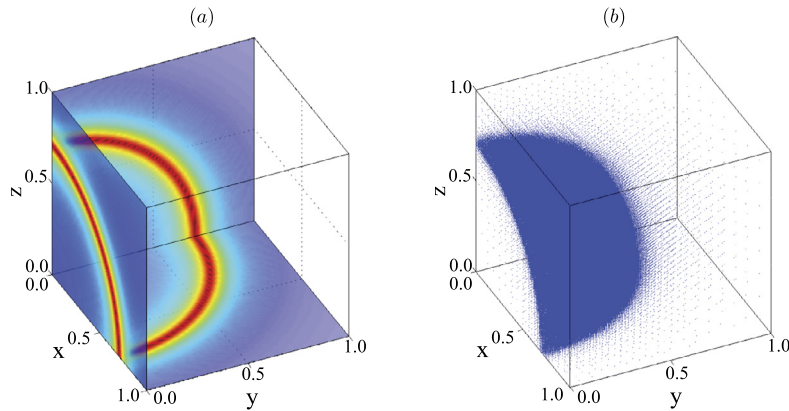
$$X_{j,k}^0 \equiv \{x_{j,k_1}\} \quad \text{and} \quad X_{j,k}^1 \equiv \bigcup_{r=1}^p x_{j,k_r^j}, \quad (12)$$

are also in the irregular grid. Algorithm 2 outlines the procedure for satisfying minimum index set requirement given a single grid point. The routine adds any missing points in the interpolation stencil and progresses recursively, expanding the grid to include the required support for the newly added points as well. Fig. 1 illustrates the minimum index set fulfilled by using Algorithm 2 for a single grid point. It should be noted that the support for each individual basis function is compact





**Fig. 2.** (a) Test function  $f(x, y) = 0.2/(|0.4 - x^2 - y^2| + 0.2)$ , (b) collocation points, and (c) their distribution at each level corresponding to  $\varepsilon = 5 \times 10^{-3}$  for  $p = 6$  and  $\mathbf{N}_0 = \{8, 8\}$ .



**Fig. 3.** (a) Sections of test function  $f(x, y, z) = 1/(|0.5 - x^2 - y^2 - z^2| + 0.1)$  and (b) collocation points corresponding to  $\varepsilon = 5 \times 10^{-3}$  for  $p = 6$  and  $\mathbf{N}_0 = \{8, 8, 8\}$ .

(i.e.,  $\text{supp } \phi_{j,k} = O(2^{-j})$ ); however, due to the hierarchical dependence on coarser levels, the total support for a single point can be quite large.

The SWR of a function is seen to concentrate points automatically in regions where the function varies rapidly. Figs. 2, and 3 show the resultant irregular grid generated from the SWR of a function with a cusp along a spherical surface in 2-, and 3-dimensions (2-, and 3-D), respectively. When the sparse grid is viewed in scale-location space, a distinctive cone-shaped region is generated around singularities or near-singularities.

For a sufficiently smooth normalized function, there is some level  $J$  such that all wavelet amplitudes on levels  $j > J$  are below the threshold value, indicating the solution is well represented by the SWR on  $J$ . From this, the maximum approximation error in  $f_\varepsilon^J$  can be estimated, and is found to be proportional to the threshold parameter  $\varepsilon$ :

$$\|f - f_\varepsilon^J\|_\infty \leq C_1 \varepsilon, \quad (13)$$

where  $C_1$  is a constant depending on  $f$ ,  $d$ , and  $p$  [50]. Furthermore, the number of basis functions used in the SWR to represent a function to within an accuracy of  $\varepsilon$  is [47,51]

$$N_E \leq C_2 \varepsilon^{-d/p}, \quad (14)$$

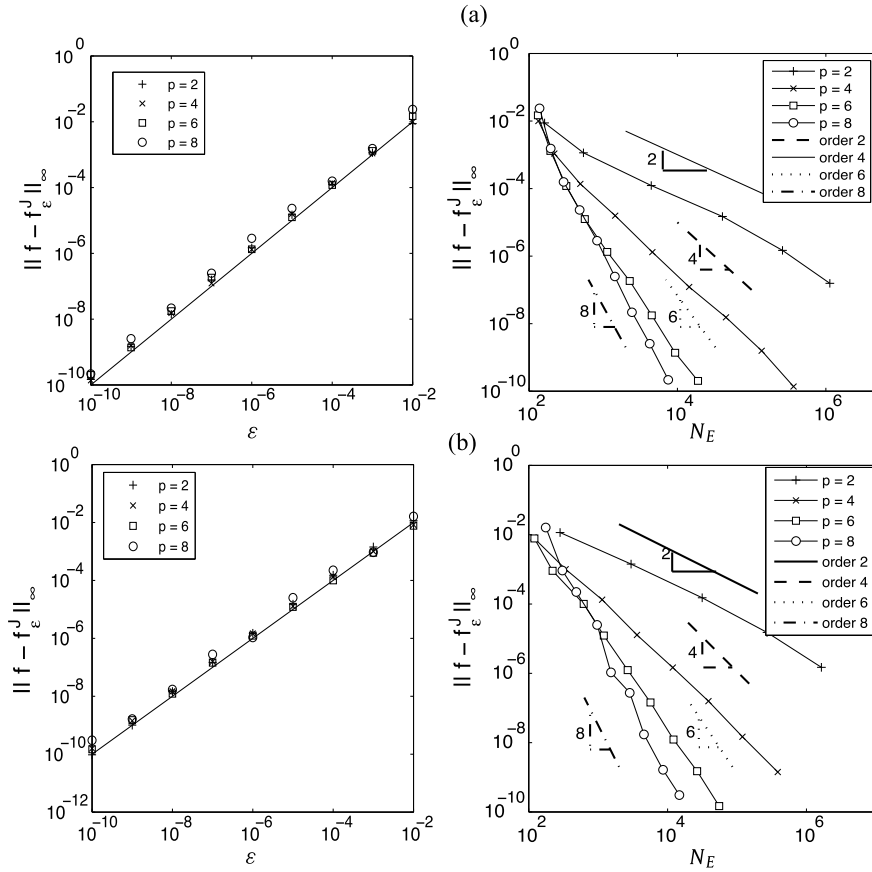
where  $C_2$  is a constant depending on  $f$ . Clearly, (13) and (14) imply the following bound:

$$\|f - f_\varepsilon^J\|_\infty \leq C_3 N_E^{-p/d}. \quad (15)$$

In the above inequalities,  $C_1$ ,  $C_2$  and  $C_3$  are  $O(1)$  constants. A more (or less) accurate approximation can be obtained by appropriately decreasing (or increasing) the value of  $\varepsilon$ , with the total number of unknowns changing accordingly. For a uniform grid,  $O(2^{dJ})$  points are required for the same accuracy; the SWR provides a substantial reduction in the total number of unknowns, with even greater benefit occurring as problem dimensionality increases.

The above estimates are verified numerically by considering the SWR of the following 2-D test functions:

$$f(\mathbf{x}) = \frac{0.05(x_1 + 0.05)}{(x_1 + 0.05)^2 + (x_2 - 0.4)^2} \quad (16)$$



**Fig. 4.** (a) Test function (16) and (b) test function (17); relationships between the approximation error  $\|f(\mathbf{x}) - f_\epsilon^J(\mathbf{x})\|_\infty$  as a function of (left)  $\epsilon$ , and (right)  $N_E$  as a results of varying  $\epsilon$  for the SWR for different orders  $p$ .

and

$$f(\mathbf{x}) = \exp\left\{-200\left[\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2\right]\right\} + \frac{1}{5} \sin(2\pi x_1) \sin(2\pi x_2). \quad (17)$$

Fig. 4 shows in the left column the maximum approximation error of  $f_\epsilon^J$  as the threshold  $\epsilon$  is varied for various orders  $p$ , and agrees with the estimate (13). The right column shows the total number of points required as the threshold is varied for several values of  $p$ . Each line has a slope of approximately  $-2/p$ , in agreement with the estimate (14).

The method used in [50] and the present work takes advantage of the conical structure formed around near-singularities in  $f$  (as seen in scale-location space) to construct the SWR gradually starting from the coarsest level. Algorithm 3 summarizes the construction of the SWR. Beginning from function values at points in the base grid  $V_0$ , the wavelet amplitudes at all points on level 1 can be determined. Performing an AFWT and thresholding identifies any points in level 1 which should be retained. Next, nearby neighbors in space and scale of those retained on level 1 are added, and the AFWT and thresholding procedure is repeated. The process continues until no additional points are added.

The region of neighboring points  $\mathcal{V}_B$  in location and scale is given by

$$\mathcal{V}_B = \left\{ \bigcup_{\{j, \mathbf{k}_j\} \in \mathcal{V}_E} \mathcal{N}_{j, \mathbf{k}_j} \right\} \setminus \mathcal{V}_E, \quad (18)$$

with the total number of neighboring points defined as  $N_B = \dim \mathcal{V}_B$ . The neighbors  $\mathcal{N}_{j, \mathbf{k}_j}$  of a point  $\mathbf{x}_{j, \mathbf{k}_j}$  are given by

$$\mathcal{N}_{j, \mathbf{k}_j} = \{x_{j+n_s, k_i-n_l} 2^{j+n_s}, \dots, x_{j+n_s, k_i+n_l} 2^{j+n_s}\}^d, \quad (19)$$

where  $n_s$  and  $n_l$  are the extent of neighbors in scale and location, respectively. The refinement strategy (19) creates a  $d$ -dimensional uniform refinement about an essential point with spacing according to scale level  $j + n_s$  and extending to  $n_l$  points on scale level  $j + n_s$  in all directions. Note from Algorithm 3 that new essential points can only be added to the SWR if they are first neighboring points. The definition of the neighboring region, therefore, is quite important in terms of maintaining the accuracy of the representation as solution features advect or grow during solution of evolution equations.



**Algorithm 3** COMPRESS.

---

Given continuous function  $f$   
 $AFWT\{f(\mathbf{x}_{j,k})\}_{\mathbf{k} \in \mathbf{k}_1^0} \rightarrow \{\{f_{0,k}\}, \{d_{1,k}\}\}$   
 $i = 1$   
 $\mathcal{V}_i = \{\mathbf{x}_{i,k} : |d_{i,k}| \geq \varepsilon\}, \mathcal{V}_0 = \mathcal{V}_1$   
**while**  $\mathcal{V}_i \neq \emptyset$  **do**  
 $i = i + 1$   
Determine the neighboring region  $\mathcal{V}_i = \bigcup_{\mathbf{x}_{i,k} \in \mathcal{V}_0} \mathcal{N}_{i,k}$   
Set  $\mathcal{V}_1 = \mathcal{V}_0 \cup \mathcal{V}_i$   
 $AFWT\{\{f(\mathbf{x}_{0,k})\}_{\mathbf{k} \in \mathbf{k}_0^0}, \{f(\mathbf{x}_{j,k})\}_{\mathbf{x}_{j,k} \in \mathcal{V}_1}\} \rightarrow \{\{f_{0,k}\}_{\mathbf{k} \in \mathbf{k}_0^0}, \{d_{j,k}\}_{\mathbf{x}_{j,k} \in \mathcal{V}_1}\}$   
 $\mathcal{V}_i = \{\mathbf{x}_{i,k} : \mathbf{x}_{i,k} \in \mathcal{V}_1 \text{ and } |d_{i,k}| \geq \varepsilon\}$   
Threshold  $\mathcal{V}_0 = \{\mathbf{x}_{j,k} : \mathbf{x}_{j,k} \in \mathcal{V}_1 \text{ and } |d_{j,k}| \geq \varepsilon\}$   
**end while**  
**Results:**  
 $\mathcal{V}_A = \mathcal{V}_0$   
 $\{\{f_{0,k}\}_{\mathbf{k} \in \mathbf{k}_0^0}, \{d_{j,k}\}_{\mathbf{x}_{j,k} \in \mathcal{V}_\varepsilon}\}$

---

Note:  $\mathcal{N}_{j,k}$  is the set of neighboring points residing on the same and next level(s) of resolution.

---

The sparse collocation grid  $\mathcal{V}_A$  consists of the union of essential points and neighboring points:

$$\mathcal{V}_A = \mathcal{V}_E \cup \mathcal{V}_B.$$

The solution of equations is advanced only at points *active* in  $\mathcal{V}_A$ . The total number of active points is given by  $N_A = N_E + N_B$ . To complete stencils needed in various operations on the sparse grid, extra non-essential points are included and their value is determined purely from interpolation. The number of non-essential points is  $N_N$ , making the total number of points  $N_T = N_A + N_N$ .

For the solution of PDEs on a sparse grid, it is necessary to calculate spatial derivatives of a function  $f$ . The wavelet basis can be differentiated; however, with varying support sizes on different levels, it is computationally expensive, requiring  $O(p(J - J_0)N_A)$  operations. As an alternative, finite differences based on the associated Lagrange interpolation polynomial are used to approximate the derivatives on the sparse grid, requiring only  $O(N_A)$  operations. For each point  $\mathbf{x} = \mathbf{x}_{J,k} \in \mathcal{V}_A$  the  $i$ th derivative of  $f$  with respect to direction  $x_r$  is

$$\left. \frac{\partial^i f}{\partial x_r^i} \right|_{\mathbf{x}} \approx D_{x_r}^{(i)} f_\varepsilon^J = \frac{1}{h^i} \sum_{l \in \mathcal{S}_{i,\mathbf{x}}^{x_r}} a_J^{r,\mathbf{x}} f(x_{J,(k_1, \dots, k_r+l, \dots, k_d)}), \quad (20)$$

where  $k_r$  is the  $r$ th-component of  $\mathbf{k}$ ,  $h$  is the grid spacing of the stencil,  $a_J^{r,\mathbf{x}}$  are finite difference coefficients, and  $\mathcal{S}_{i,\mathbf{x}}^{x_r}$  is the set of indices for the  $n$ th-order accurate finite-difference stencil at  $\mathbf{x}$  [31]. The stencils are taken to be centered differences or equally accurate one-sided differences at or near boundaries. The pointwise error in the derivative approximation is estimated to be bounded by

$$\|\partial^i f / \partial x^i - D_{x_r}^{(i)} f_\varepsilon^J\|_{\mathcal{V}_A, \infty} \leq C_4 N_A^{-\min((p-i), n)/d}, \quad \|f\|_{\mathcal{V}_A, \infty} \equiv \max_{\mathbf{x} \in \mathcal{V}_A} |f(\mathbf{x})|. \quad (21)$$

From the above estimate, the error is bounded by the least accurate of either the sparse representation of  $f$  on the stencil,  $O(N_A^{-(p-i)/d})$ , or the finite difference approximation on  $\mathcal{V}_A$ ,  $O(N_A^{-n/d})$ . As such, the order of the finite difference stencil chosen should be commensurate with the order of wavelets used,  $n = p - \max(i)$ , where  $\max(i)$  is the highest-order derivative present in the governing equations. Fig. 5 shows the maximum error in the first and second derivative of the SWR of (16) for different combinations of  $\varepsilon$ ,  $p$ , and  $n$ , and the results agree reasonably well with the estimate (21).

### 3.1. Time integration

The complete algorithm for solving a system of time-dependent PDEs,

$$\frac{\partial u}{\partial t} = \mathbf{F}(t, u, u_{\mathbf{x}}, u_{\mathbf{xx}}, \dots),$$

with initial conditions

$$u(\mathbf{x}, 0) = u^0(\mathbf{x})$$

and given boundary conditions, can be summarized as follows. Apply the wavelet transform to the initial conditions. Subsequently, by thresholding the obtained wavelet amplitudes we have an initial irregular grid. In addition, apply the wavelet

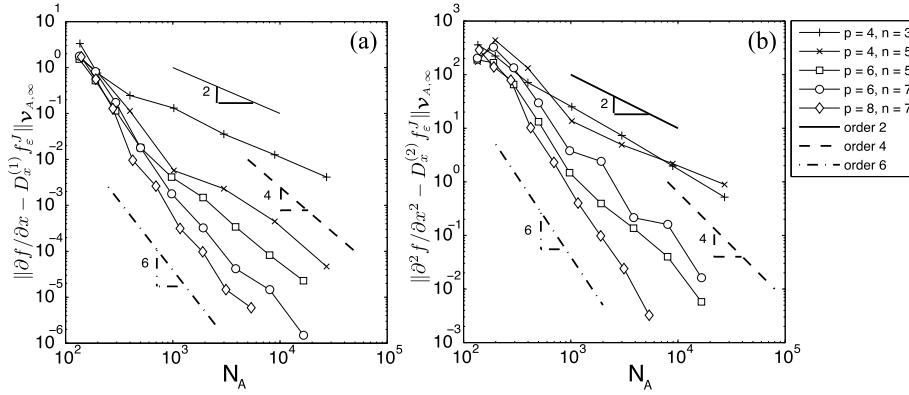


Fig. 5.  $L_{V_{A,\infty}}$ -error of derivative approximations on irregular grids,  $\|\partial^i f / \partial x^i - D_x^{(i)} f_\varepsilon^J\|_{V_{A,\infty}}$ , of the test function (16) as functions of  $N_A$  for different values of  $p$ ,  $n$ , and  $\varepsilon$ ; (a) first derivative and (b) second derivative.

---

**Algorithm 4** Dynamically adaptive solver for time-dependent problem.

---

Given  $\varepsilon$  and  $t_{final}$

Construct the initial irregular grid  $V_A^0$  by thresholding  $u^0$

**for**  $m = 0$  to  $t_{final}$  **do**

Estimate  $\Delta t$  from CFL condition

Solve problem (22) for  $u^{m+1}$  based on  $u^m, u^{m-1}, \dots, u^{m-q}$  on grid  $V_A^m$

# find essential grid points

Perform  $AFWT(u^{m+1}) \rightarrow \{u_{j_0, \mathbf{k}}^{m+1}, \{d_{l, \lambda}^{m+1}\}_{(l, \lambda) \in \Lambda^m}\}$

Gather  $\Lambda_s = \{(l, \lambda) \in \Lambda^m: |d_{l, \lambda}^m| \geq \varepsilon\}$

# add neighboring points to grid of essential points

Determine the neighbor region  $\Lambda_{N_B} = \bigcup_{(l, \lambda) \in \Lambda_s} \mathcal{N}_{l, \lambda}$  and establish  $\Lambda^{m+1} = \Lambda_s \cup \Lambda_{N_B}$

# new irregular grid

Construct  $V_A^{m+1} = \{u_{j_0, \mathbf{k}}^{m+1}, \{u_{l, \lambda}^{m+1}\}_{(l, \lambda) \in \Lambda^{m+1}}\}$

# compute  $u(n+1), \dots, u(n-q-1)$  with respect to new grid

**for**  $r = 0$  to  $q-1$  **do**

Establish  $\tilde{D} = \{u_{j_0, \mathbf{k}}^{m+1-r}, \{\tilde{d}_{l, \lambda}\}_{(l, \lambda) \in \Lambda^{m+1}}\}$  with

$$\tilde{d}_{l, \lambda} = \begin{cases} d_{l, \lambda}^{m+1-r} & (l, \lambda) \in \Lambda^{m+1} \cap \Lambda^m \\ 0 & (l, \lambda) \in \Lambda^{m+1} \setminus \Lambda^m \end{cases}$$

Perform  $AIWT(\tilde{D}) \rightarrow u^{n+1-r}$

**end for**

**end for**

---

transform to the PDEs and boundary conditions to obtain the following system of ordinary differential equations (ODEs) for  $u(t) = \{u(\mathbf{x}_{j, \mathbf{k}}, t)\}$ :

$$\frac{du}{dt} = \mathbf{G}(t, \mathbf{u}). \quad (22)$$

Eq. (22) is subsequently integrated in time, and the process is repeated until a final user-selected time is reached. Details on the adaptive procedure are provided in Algorithm 4. In fact, the WAMR method is designed as a method of lines algorithm [52], providing dynamic spatial grid adaptation and discretization, and the time integration handled by any appropriate ODE integrator. Schemes with error control should be used. These schemes take an initial time step as input, and will refine the time step as needed to satisfy a user-supplied error tolerance. The initial time step that we use is based on the CFL condition. This also corresponds to the time step for which the collocation grid is adapted. While a number of time integration schemes have been used with WAMR, all results presented here and in the next part have been obtained using a standard explicit 4th/5th-order Runge–Kutta–Fehlberg (RKF45) scheme [53]. The RKF45 scheme is a 5th-order Runge–Kutta method with an embedded 4th-order method, allowing one to make an estimate of the truncation error. From the estimate, the time step is altered accordingly to satisfy prescribed relative and absolute error tolerances; the step-size is refined and repeated if the error tolerances are not satisfied or lengthened for the next step, otherwise.

#### 4. Real and numerical diffusion

Real diffusion plays an important role in physics. Without it, momentum does not dissipate, heat does not conduct, and gases do not diffuse and potentially react. Numerical diffusion, on the other hand, is an artificial entity arising in numerical approximations of differential equations. Indeed, quite often, in problems whose solutions numerically exhibit discontinuities, without the proper amount of numerical diffusion, methods would lead to unphysical behavior and at worst would result in numerical instability.

It is noted that there are two types of discontinuities that one encounters; discontinuities that are present in the analytical problem that the numerical solution tries to approximate, and apparent discontinuities that are not present in the analytical problem, but occur in the numerical problem due to insufficient numerical resolution. In the latter case, the insufficient resolution may be intentional or unintentional depending on the aims of the numerical modeler. In any case, generally the numerical diffusion is due to discrete approximations of many terms occurring in a model; most, if not all, of these terms are not related to physically dissipative effects, and thus the resulting numerical forms of dissipation may potentially violate physical requirements (e.g., the 2nd law of thermodynamics, frame-invariance, etc.). Numerical diffusion that arises naturally as part of derivative approximations is called implicit, while numerical diffusion that is purposely added is called explicit.

The presence of real or numerical diffusion sets a lower bound on the smallest grid size necessary for resolution of the dissipative effects. The solution of the inviscid problem using a monotone scheme effectively includes a numerical viscosity to numerically stabilize the solution in the presence of a shock wave (which is a true discontinuity in the inviscid problem) [54]. The subsequent artificial kinematic viscosity near a shock in 1-D is

$$\nu_a \approx \frac{1}{2} \Delta x \|u + c\|,$$

where  $\Delta x$  is the local grid spacing, and  $c$  the frozen speed of sound. However, in problems that we address in this work, real diffusion is always present. Nevertheless, it should be pointed out that when the grid size is sufficiently large, the numerical diffusion determines the smallest length scale present in a problem, and subsequently one would get the same result when solving the analytical non-dissipative or dissipative equations [16]. Because of this fact, some researchers unfortunately draw the wrong conclusion that it is not necessary to solve the Navier–Stokes equations since the Euler equations yield the same results (e.g. [55]). This is not the case, as clearly demonstrated by Mazaheri et al. [56]. The Euler equations contain no small scale cutoff so that as one increases the resolutions (and thus reduces the artificial diffusion) the solution will always keep changing and remain unresolved. However, when solving dissipative equations, the physical diffusion imposes a small scale cut-off so that the solution subsequently will not change after sufficient refinement, i.e., when the numerical diffusion becomes much smaller than the physical diffusion.

Thus, whenever solving a problem where real diffusion is important, proper resolution is necessary to ensure that all scales are resolved. This is imperative in problems where instabilities occur and physical diffusion plays an important role in determining unstable modes, or involving physical mass diffusion and reactions since the degree of reaction and the specific reaction that would occur in a detailed chemical mechanism is determined by the local concentration of different species [57]. Adaptive methods like WAMR make it possible to use grids sufficiently fine to guarantee that the numerical diffusion is much smaller than the physical diffusion. Obviously, if one is not interested in describing the behavior of all species participating in mass diffusion and reaction, then a one- or two-step reaction model is sufficient and this degree of resolution is not necessary. Nevertheless, one should be mindful of the role that the numerical diffusion plays in the solution of such problems.

#### 5. Code verification

To quote P. Roache [58], “The identification, elimination, and engineering ‘proof’ or demonstration of the absence of coding errors are the concern of Code Verification.” This section demonstrates the verification of the WAMR code by comparing solutions for several test cases to analytic or accurate reference solutions. For each problem considered in the following subsections, the thresholding parameter  $\varepsilon$  is noted, and the finest resolution level used is determined automatically.

##### 5.1. Advection–diffusion–reaction problem

In order to test the adaptive algorithm for time dependent problem, we consider the following 2-D model problem:

$$\begin{aligned} \frac{\partial u}{\partial t} + (\mathbf{V} \cdot \nabla)u &= \nu \nabla^2 u + f(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times [0, +\infty), \quad \Omega = (0, 1)^2, \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega \text{ and } u(\mathbf{x}, 0) = q(\mathbf{x}), \end{aligned} \quad (23)$$

where  $\mathbf{V} = (0 \ 1)^T$  and  $\nu = 1/100$ . The source term  $f(\mathbf{x}, t)$ , specified Dirichlet conditions  $g(\mathbf{x}, t)$ , and the initial condition  $q(\mathbf{x})$  are chosen so that they satisfy the following exact solution:

$$u_e(\mathbf{x}, t) = \frac{0.05(x_1 + 0.05)}{(x_1 + 0.05)^2 + (x_2 - t)^2}. \quad (24)$$

The calculation is performed until  $t = 1$  is reached. Small relative and absolute errors in the time integration are enforced so as to minimize the error in the time discretization, i.e. the error in the approximate solution is dominated by that of the spatial discretization in order to clearly demonstrate such errors. Different combinations of  $p$  and  $n$  are used in the simulations. The maximum allowable resolution  $J - j_0$  is set to a large value so that the accuracy of the approximate solution is controlled by the threshold parameter  $\varepsilon$ . Note that the coarsest scale  $j_0$  is set to 3 ( $9 \times 9$  for the coarsest grid). Threshold values are varied from  $5 \times 10^{-3}$  to  $1 \times 10^{-7}$ . Fig. 6 shows snapshots of the approximate solution and irregular grid produced dynamically by the adaptive algorithm at different times. Results shown in the figure are those computed with  $p = 4$ ,  $n = 4$ , and  $\varepsilon = 5 \times 10^{-4}$ . It can be noticed that the computational grid at any time has fine resolution in the proximity of the near-singularity and coarse resolution in the smooth areas. This figure clearly indicates that the adaptive algorithm is able to follow the moving structure of the solution properly. The number of active grid points demanded by the adaptive algorithm during the course of the simulation with different threshold values is plotted in Fig. 7. Note that the integral of the exact solution as a function time is symmetric around  $t = 0.5$ . We thus expect the number of grid points generated by the adaptive algorithm to show some symmetry with respect to this specific time. Indeed, it can be observed that the plots appear fairly symmetric around  $t = 0.5$ . Note also that as the threshold parameter  $\varepsilon$  is decreased the number of grid points demanded by the adaptive algorithm increases automatically. Fig. 8 depicts the error in the discrete maximum norm  $L_{\mathbf{V}_{A,\infty}}$  of the approximate solution obtained with the adaptive method with  $p = 4$  and  $n = 4$  during the course of the simulation. It can be observed that the error in the approximate solution stays almost constant as a function of time (after the first few integration steps).

Fig. 9 shows the log–log plots of the error in  $L_{\mathbf{V}_{A,\infty}}$  in the approximate solution at  $t = 1/2$  and 1 versus  $N_A$ , the number of active collocation points required by the adaptive algorithm, for different combinations of  $p$  and  $n$  as a result of varying the threshold parameter  $\varepsilon$ . The log–log plots of  $N_A$  versus  $\varepsilon$  are included in the right column of the figure. The slopes of the log–log plots indicates that, at  $t = 1/2$ , the number of grid points behave like  $N_A = O(\varepsilon^{-2/c_1})$ , where  $c_1$  is approximately 4.4 for  $p = 4$  and  $n = 4$ , 7.1 for  $p = 6$  and  $n = 4$ , and 10.9 for  $p = 8$  and  $n = 6$ . At  $t = 1$ , the number of grid points  $N_A$  is proportional to  $O(\varepsilon^{-2/c_1})$ , where  $c_1$  is approximately 4.5 for  $p = 4$  and  $n = 4$ , 7.4 for  $p = 6$  and  $n = 4$ , and 9.4 for  $p = 8$  and  $n = 6$ . Note that the values of  $c_1$  are slightly larger than those predicted by (14). The constant  $c_2$  from a least square fit in the form  $\|u_e - u_\varepsilon\|_{\mathbf{V}_{A,\infty}} = O(N_A^{-c_2/2})$  at  $t = 1/2$  is approximately 3 for  $p = 4$  and  $n = 4$ , 4.2 for  $p = 6$  and  $n = 4$ , and 9 for  $p = 8$  and  $n = 6$ . At  $t = 1$ , the constant  $c_2$  is approximately 3.2 for  $p = 4$  and  $n = 4$ , 4.1 for  $p = 6$  and  $n = 4$ , and 7.9 for  $p = 8$  and  $n = 6$ . Such rates of convergence for the adaptive algorithm with  $p = 4$  and  $p = 8$  are larger than  $O(N_A^{-\min(p-2,n)/2})$  predicted by (21). For  $p = 6$ , the rate of convergence is in good agreement with the prediction. Note that the constant  $c_3$  from a least square fit in the form  $\|u_e - u_\varepsilon\|_{\mathbf{V}_{A,\infty}} = O(\varepsilon^{c_3})$  at  $t = 1/2$  is approximately 0.69 for  $p = 4$  and  $n = 4$ , 0.59 for  $p = 6$  and  $n = 4$ , and 0.83 for  $p = 8$  and  $n = 6$ . At  $t = 1$ , such constant equals approximately 0.70 for  $p = 4$  and  $n = 4$ , 0.55 for  $p = 6$  and  $n = 4$ , and 0.81 for  $p = 8$  and  $n = 6$ . Note that for  $p = 4$  and  $n = 6$ , errors in the discrete maximum norm are larger than the threshold values as the threshold values are decreased (as indicated by the fact that the constants  $c_3$  for  $p = 4$  and  $p = 6$  are smaller than unity). For the range of threshold values used, the errors in the approximate solution for  $p = 8$  are of the same order of the threshold values (despite the fact that  $c_3$  for  $p = 8$  is smaller than unity).

## 5.2. Sod problem

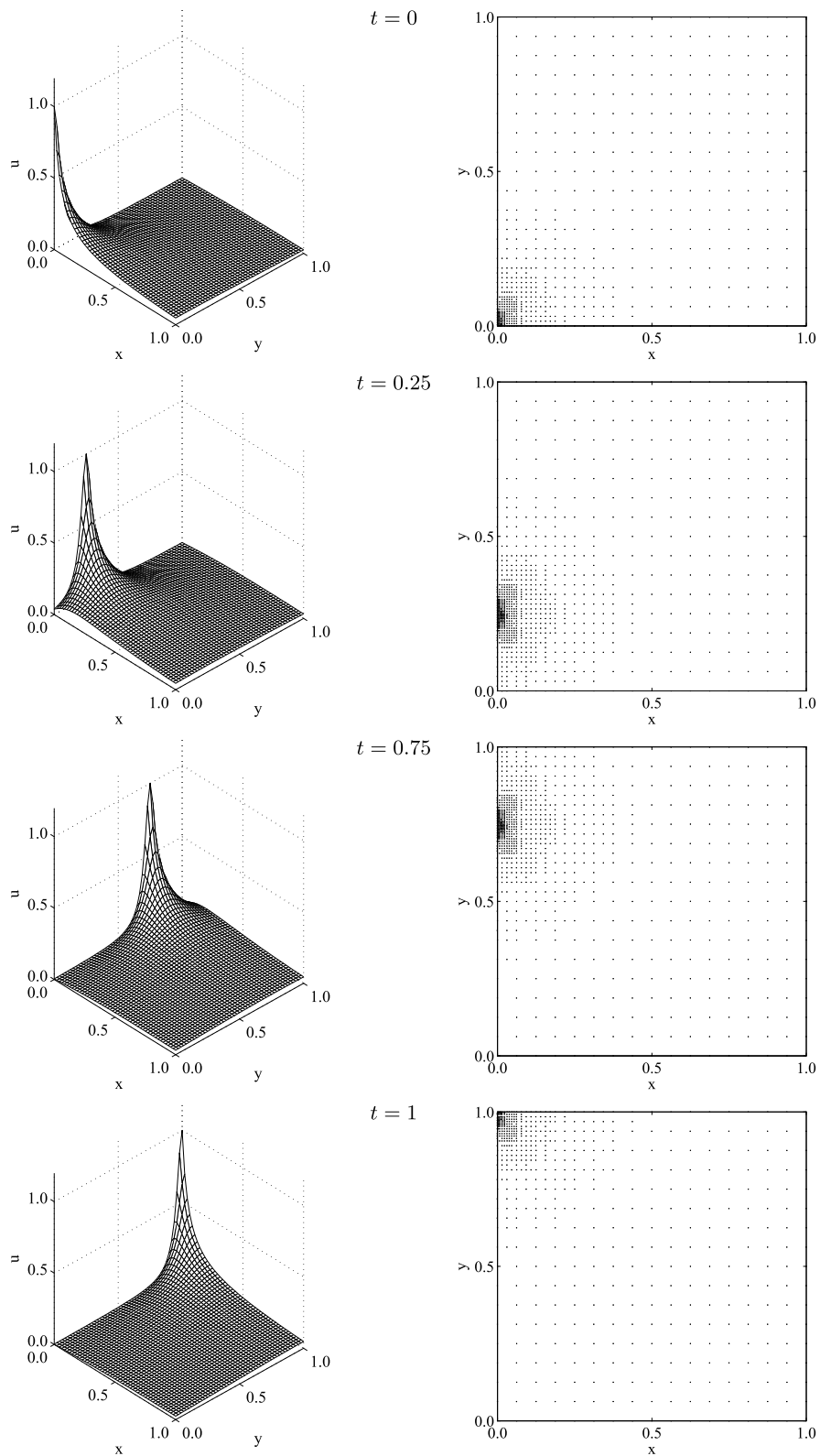
The Sod shock tube problem [59] was originally constructed to test numerical methods for the solution of hyperbolic conservation laws. The problem is classically modeled by the Euler equations and has an analytic solution. This has led to its continued use as a test problem for both compressible hydrodynamics codes and for hyperbolic equation solvers [60].

The problem represents the approximation of a one-dimensional shock tube filled with an initially quiescent, isothermal gas. The tube is separated into two equal regions by a diaphragm, with a 10:1 pressure ratio between the two regions. At the start, the diaphragm is removed, and a shock wave propagates into the low-pressure side followed by a contact discontinuity moving at a slower speed, and a rarefaction fan spreads in the opposite direction. The exact solution to this problem is for the non-dissipative problem and the shock and contact surfaces are represented as true discontinuities. The numerical problem that we solve using the WAMR method is the more physically accurate dissipative, multi-component shock tube problem of which the Sod problem approximates. That is, the full governing equations described in Section 2 are used, as opposed to the Euler equations.

The domain of the Sod shock tube problem is normalized to  $x \in [0, 1]$  with the diaphragm located at  $x_0 = 0.5$ . The low pressure side is at roughly atmospheric conditions with pressure,  $p_L = 10^6$  dyne/cm<sup>2</sup>, and temperature,  $T_0 = 300$  K. The high-pressure side is at the same temperature with a 10:1 pressure ratio,  $p_H = 10^7$  dyne/cm<sup>2</sup>. The initial pressure distribution is smoothed by a hyperbolic tangent function given as

$$p = \frac{1}{2} \left[ (p_H + p_L) - (p_H - p_L) \tanh\left(\frac{x - x_0}{\delta}\right) \right], \quad (25)$$

with  $\delta = 10^{-3}$ . The gas is a uniform, inert, three-component mixture of 78N<sub>2</sub>:21O<sub>2</sub>:Ar to represent air. At both ends of the tube reflecting, impermeable, adiabatic wall boundary conditions are used. A coarsest level of  $N_0 = 10$  is used with error threshold of  $\varepsilon = 10^{-3}$  and wavelet parameters  $p = 6$  and  $n = 4$ .



**Fig. 6.** Approximate solution (left) and the irregular grid (right) generated by the adaptive method with  $p = 4$ ,  $n = 4$ ,  $\varepsilon = 5 \times 10^{-4}$ .

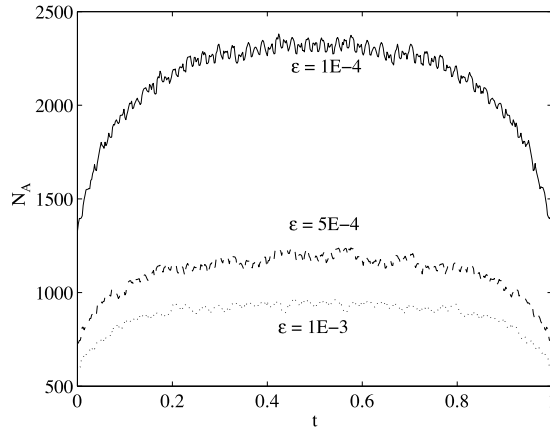


Fig. 7. Number of active points as a function of time using  $p = 4$ ,  $n = 4$  and different  $\varepsilon$ .

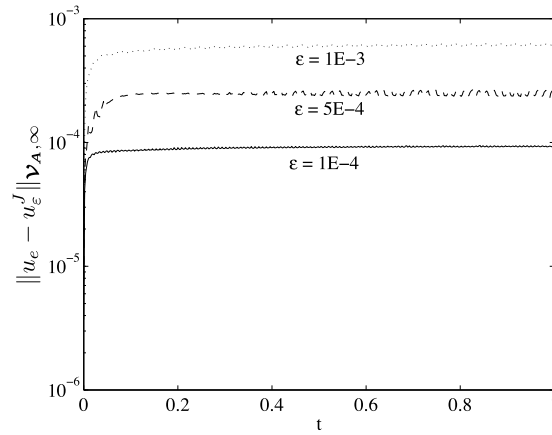
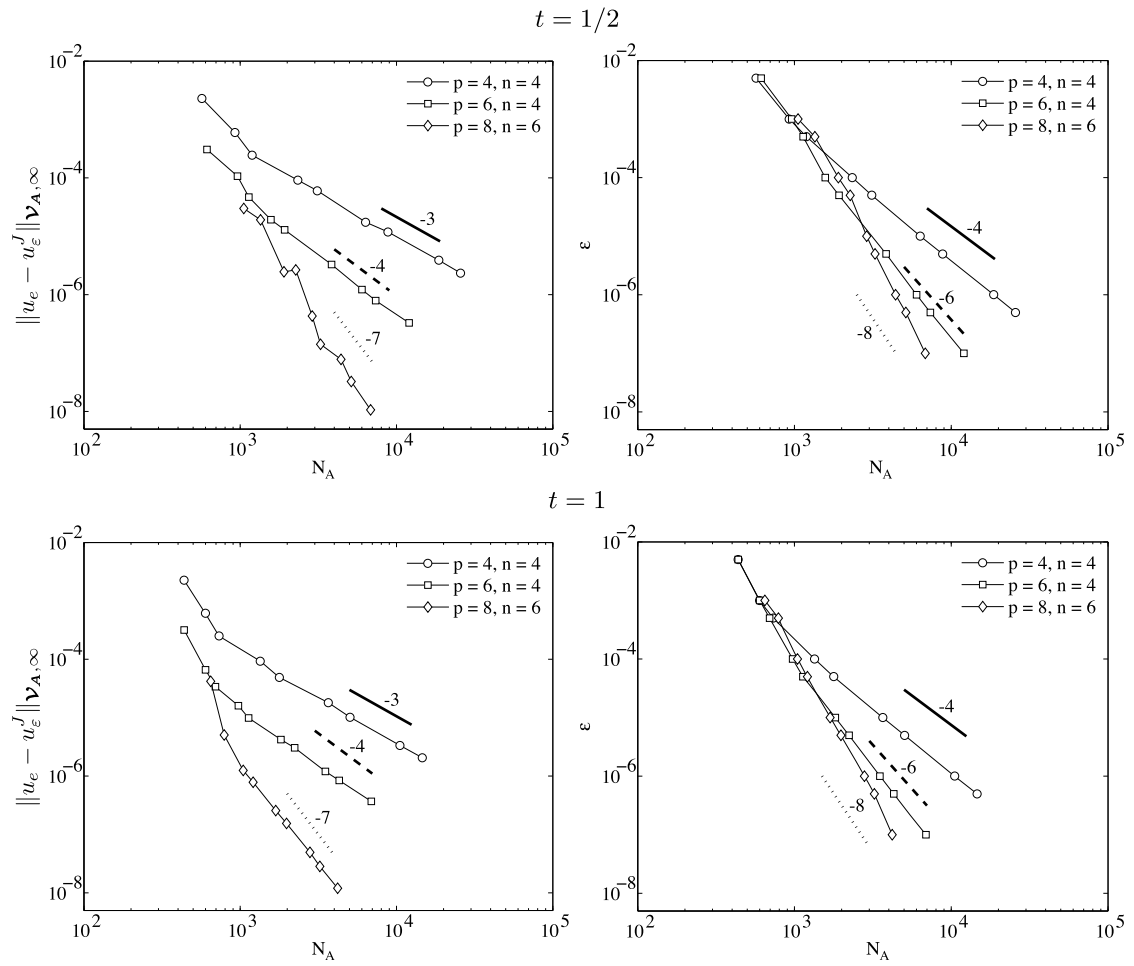


Fig. 8.  $L_{\infty}$  error as a function of time for  $p = 4$ ,  $n = 4$  and different  $\varepsilon$ .

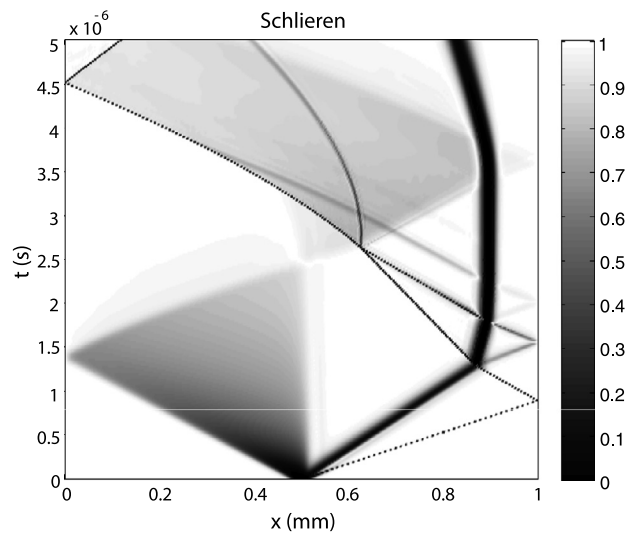
Fig. 10 shows the numerical Schlieren image (magnitude of the density gradient) in the  $x$ - $t$  plane. A shock wave, followed by a contact discontinuity, moves toward the low-pressure region on the right, while the head of a sonic rarefaction wave moves toward the high-pressure region on the left (the tail of the rarefaction wave is stationary). The figure shows very clearly the complex interactions between the transmitted and reflected contributions of the shock wave and the contact discontinuity and (later) the reflected rarefaction wave. It should be noted that the contact discontinuity never reaches the end of the tube – its motion is slowed down and eventually reversed by the reflected shock wave contributions. Fig. 11 shows the density fields from the inviscid (or non-dissipative) analytical and viscous (or dissipative) numerical solution of the Sod problem at a time of  $0.5 \mu\text{s}$ . The numerical solution shows excellent agreement with the exact solution. The subplots in Fig. 11 show close up views of the contact and shock layers. In the close-up views, the effects of physical dissipation in the solution is clearly visible, with both waves having a smooth, continuous profile rather than a discontinuous jump. Note that no monotone-preserving scheme is used in the computation. The viscous profiles of the waves are well-defined by the concentration of collocation points within each structure with each symbol corresponding to a single collocation point. Fig. 12 shows in the top row, comparisons of the pressure, velocity, and temperature fields, which also exhibit very good agreement with the exact (inviscid) solution. The bottom row of Fig. 12 shows the mass fractions of each species. The inclusion of baro- and thermo-diffusive terms in the mass flux creates inhomogeneities in the species fields which do not occur otherwise. It is noted from (A.9) and (A.10) that the mass flux is related to the gradients of temperature and pressure, and small spikes occur where the gradients of one or both is large. Fig. 13 shows a detail plot of the mass fraction of  $\text{O}_2$  near the contact and shock layers. The spike is larger and sharper near the shock, where there is a strong gradient in both the temperature and pressure. At the contact layer, the amplitude is smaller with a weaker gradient occurring only in the temperature. The direction of diffusion of a specie, i.e., from high to low temperature or vice-versa, in a mixture is a complicated function of the mixture composition, molecular masses and diameters, and collision characteristics as can be seen in the system defining the thermal and multi-component diffusion coefficients (see [41,61,62] for more details concerning calculations of multi-component and thermal diffusion coefficients).

Since we solve the diffusive problem, there is no analytical solution to compare with and thus provide convergence measures of true error. Subsequently, to demonstrate convergence rates, we performed a calculation using the same algorithm

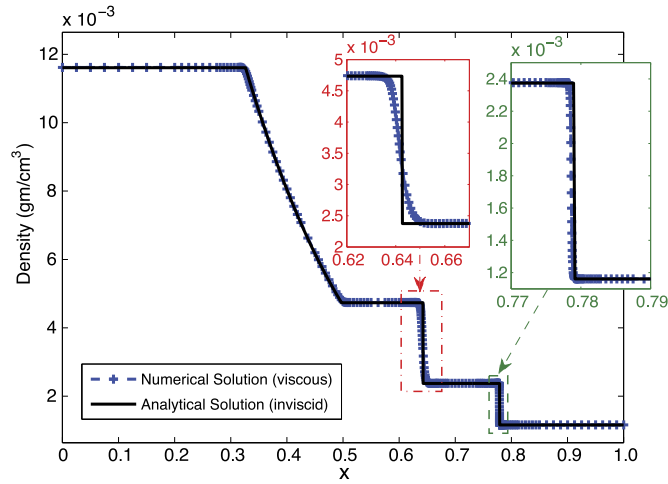




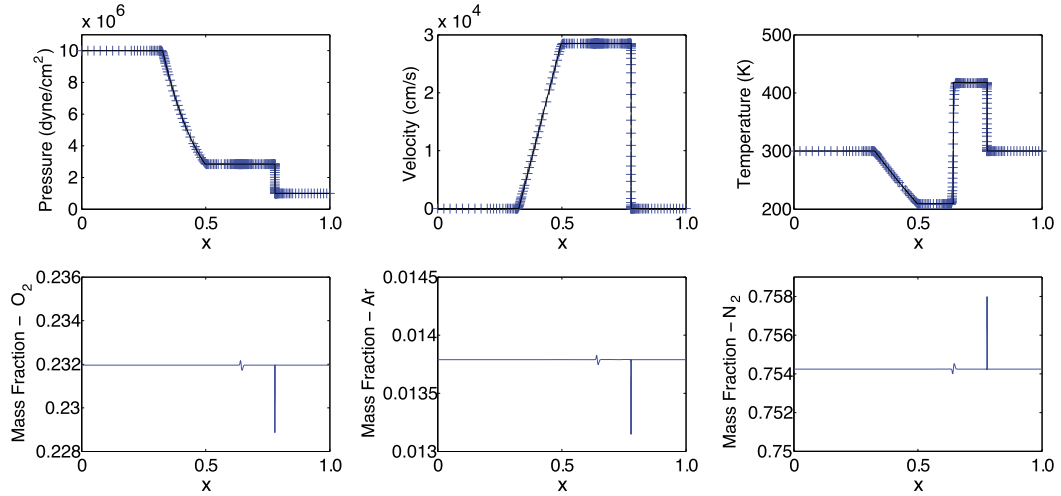
**Fig. 9.**  $L_{V_{A,\infty}}$ -error as a functions of the number of grid points,  $N_A$ , resulting from varying  $\varepsilon$  (left), and  $\varepsilon$  versus  $N_A$  (right) at  $t = 1/2$  and  $t = 1$ .



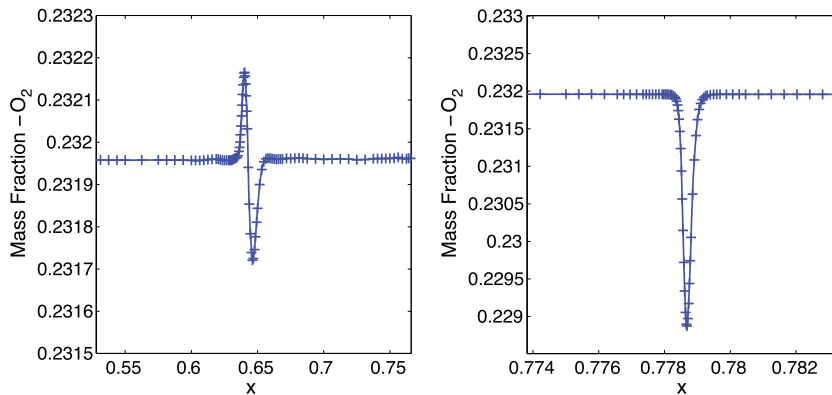
**Fig. 10.** Numerical Schlieren image in the  $x$ - $t$  plane.



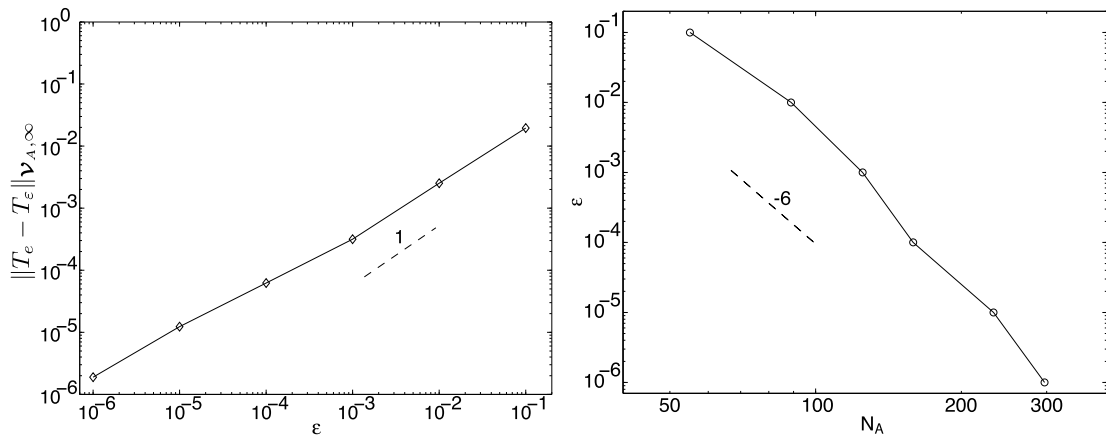
**Fig. 11.** Density fields of the numerical (viscous) and exact (inviscid) solutions of the 1-D Sod problem at  $t = 0.5 \mu\text{s}$  with magnifications of contact and shock regions.



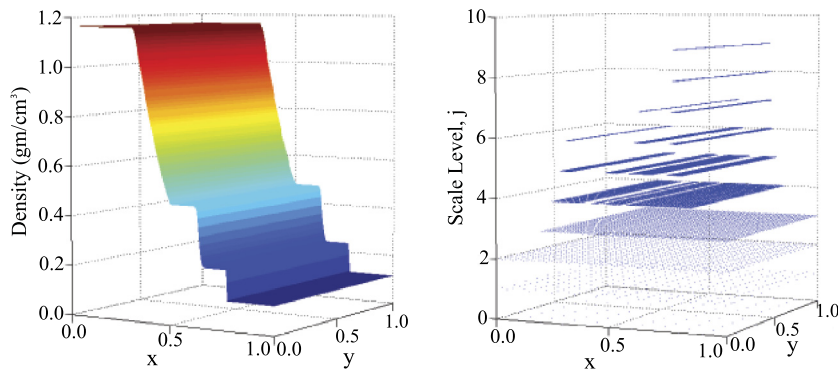
**Fig. 12.** Pressure, velocity, temperature, and species mass fractions of numerical (viscous) and exact (inviscid) solutions at  $t = 0.5 \mu\text{s}$ .



**Fig. 13.** Close up view of the  $\text{O}_2$  mass fraction near the contact (left) and shock (right) layers.



**Fig. 14.**  $L_{V_{A,\infty}}$ -error of temperature as a functions of  $\varepsilon$  (left), and  $\varepsilon$  as a function of the number of active grid points,  $N_A$  (right), at time  $t = 0.05 \mu\text{s}$  for  $p = 6$  and  $n = 4$ .



**Fig. 15.** Density field (left) and adaptive grid (right) of the 2-D viscous Sod problem.

but with a uniform spatial grid much smaller (corresponding to  $\varepsilon = 10^{-14}$ ) than the smallest grid used by the adaptive algorithm. We treated such result as a very accurate representation of the solution and used it to computed errors produced by the adaptive algorithm at  $t = 0.05 \mu\text{s}$ . In addition, as done in the previous case, small relative and absolute errors in the time integration are enforced so as to minimize the error in the time discretization. As can be seen from Fig. 14, we obtain the expected convergence rates based on estimates provided by (13), (14), and (21) for wavelet parameters  $p = 6$  and  $n = 4$ . The convergence of the error as a function of the threshold parameter is slightly less than unity, while the convergence of threshold parameter as a function of the number of active points has a slightly steeper slope than  $-6$ .

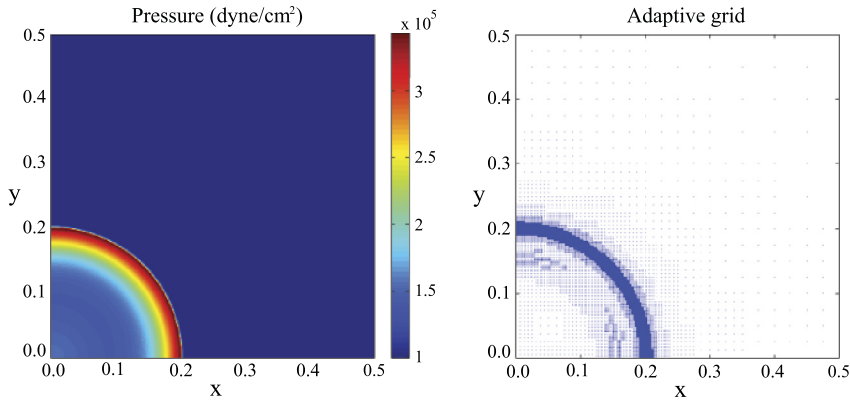
A two-dimensional version of the Sod problem is also considered for verification purposes. The dimensionless domain is square,  $\mathbf{x} \in [0, 1]^2$ , with symmetry boundary conditions on the top and bottom sides. The initial conditions are of the same form as (25). The result is planar propagating waves in the  $x$ -direction; Fig. 15 shows the density field for this case and the adaptive grid in location-scale space. The waves remain perfectly planar as expected (the same problem with the shock propagating in the  $y$ -direction was performed with identical results). As can be noted, grid points are concentrated in areas with the finest structures, most notably in the shock and contact layers.

### 5.3. Taylor/Sedov blast wave

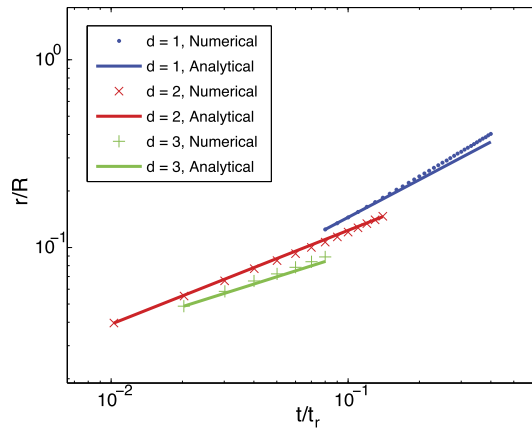
The Taylor/Sedov blast wave problem gives rise to a similarity solution originally used to estimate the energy released from a nuclear explosion [63,64]. The blast wave problem considers a large but finite amount of energy released instantaneously at a point. The energy release generates a radially symmetric shock wave expanding from the point. The similarity solution assumes a non-dissipative fluid with an initially infinite pressure ratio. The radius of the expanding shock wave,  $r$ , is given as a function of time as

$$r(t) = \left( \frac{E_0}{\rho_0} \right)^{1/a} t^{2/a}, \quad a = 2 + d, \quad (26)$$

where  $E_0$  is the energy released,  $\rho_0$  is the density of the atmosphere, and  $d$  is the dimensionality of the problem.



**Fig. 16.** Pressure field (left) and adaptive grid (right) for two-dimensional viscous Taylor/Sedov blast wave at  $t = 30$  ns.



**Fig. 17.** Comparison of analytic (inviscid) and numerical (viscous) results of Taylor/Sedov blast-wave radius, the radius is scaled by domain size,  $R = 0.1$  cm, and time by  $t_r = 1$   $\mu$ s.

Our solution of the Taylor/Sedov problem considers the more realistic dissipative problem in a  $d$ -dimensional domain with physical dimensions in each coordinate direction being  $L_1$ ,  $L_2$ , and  $L_3$ , respectively. The domain is filled with the same inert three-component  $78\text{N}_2:21\text{O}_2:\text{Ar}$  air mixture used for the Sod problem, and is initially at rest with temperature  $T_0$  and pressure  $p_0$ . An initial pressure spike with a Gaussian profile is located at a point  $\mathbf{x}_0$  in the domain,

$$p(\mathbf{x}, 0) = p_0 \left[ 1 + p_R \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{\delta}\right) \right], \quad (27)$$

where  $p_R$  is the peak pressure ratio,  $\|\mathbf{y}\|$  indicates the standard Euclidean norm, and  $\delta$  is a parameter measuring the width of the Gaussian. The density is initially uniform and the temperature is obtained from the equation of state (A.1) using the pressure given in (27). The peak pressure ratio is limited by the use of thermodynamic properties with dependencies on curve fits of species specific heats. The curve fits are valid for the range of  $300\text{--}5000$  K  $\pm 20\%$ , providing a maximum temperature of approximately  $6000$  K. Generally, using a reference temperature of  $T_0 = 300$  K restricts the peak pressure ratio to  $0 \leq p_R \leq 20$ . For the present problem, the dimensionless domain is given by  $L_i = 1$  for  $i = 1, 2$  and  $3$ . The parameters for (27) are  $p_0 = 10^5$  dyne/cm<sup>2</sup>,  $T_0 = 300$  K,  $p_R = 20$ , and  $\delta = 0.02$ . The symmetry of the blast wave is used with the initial pressure spike located in the corner of the computational domain,  $\mathbf{x}_0 = \mathbf{0}$ , and symmetry boundary conditions are used.

Fig. 16 shows the pressure field for the 2-D problem and the associated adaptive grid. The results behave as expected, retaining radial symmetry and the adaptive grid automatically refining near the shock front. Fig. 17 plots the blast-wave radius as a function of time with comparisons to the inviscid similarity solution. The shock wave in the numerical solution is taken to be located at the radial distance of the maximum slope of the pressure field. In each case, there is very good agreement between the numerical and analytical solutions.

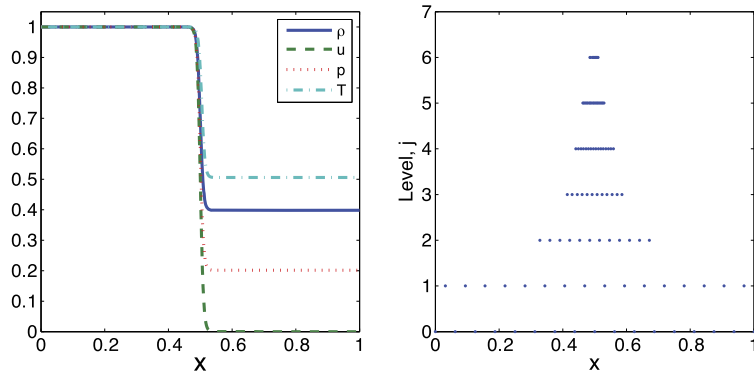


Fig. 18. Initial conditions and corresponding collocation grid for 1-D viscous detonation.

#### 5.4. 1-D detonation

Following the problem considered in [30,65], a one-dimensional detonation is simulated. The fluid is a stoichiometric mixture of hydrogen and oxygen, diluted by 70% argon,  $2\text{H}_2:\text{O}_2:7\text{Ar}$ . The chemical mechanism is included in Appendix B and contains 37 elementary reactions involving nine species: H, O, Ar,  $\text{H}_2$ ,  $\text{O}_2$ , OH,  $\text{H}_2\text{O}$ ,  $\text{HO}_2$ , and  $\text{H}_2\text{O}_2$ . The domain is 12 cm long, the left boundary is an inlet and the right a solid, adiabatic wall. The problem is initialized with a shock traveling to the right with the following jump conditions, with subscripts 1 and 2 designating the left and right sides of the jump:

$$\begin{aligned} x < 6 \text{ cm}, & & 6 \leq x \leq 12 \text{ cm}, \\ \rho_1 = 1.8075 \times 10^{-4} \text{ g/cm}^3, & & \rho_2 = 7.2 \times 10^{-5} \text{ g/cm}^3, \\ p_1 = 3.5594 \times 10^6 \text{ dyne/cm}^2, & & p_2 = 7.173 \times 10^5 \text{ dyne/cm}^2, \\ u_1 = 4.8734 \times 10^4 \text{ cm/s}, & & u_2 = 0.0 \text{ cm/s}. \end{aligned}$$

The initial shock profile is given by a hyperbolic tangent:

$$\Gamma(x) = \frac{1}{2} \left[ (\Gamma_1 + \Gamma_2) - (\Gamma_1 - \Gamma_2) \tanh\left(\frac{x - x_0}{\delta}\right) \right], \quad (28)$$

where  $\Gamma = (\rho, p, u)$ ,  $x_0 = 6 \text{ cm}$ , and  $\delta = 1.2 \times 10^{-2} \text{ cm}$ .

Fig. 18 shows the scaled initial conditions in terms of density, velocity, pressure, and temperature, and the associated distribution of collocation points by resolution level. The shock travels down the tube and reflects off the right end-wall; the temperature behind the reflected shock rises and initiates a reaction wave traveling to the left. The reaction wave eventually catches and overtakes the leading shock to form a detonation wave. Fig. 19 shows scaled plots of the density, velocity, pressure, and temperature at times of 180 and 230  $\mu\text{s}$ . Fig. 20 shows the pressure and temperature near the detonation front. The von Neumann spike appears to be a sharp peak in the previous views, but upon zooming in, one sees the fully resolved leading shock and the plateau of the induction zone which extends less than 0.5 mm behind the shock. Fig. 21 shows the distribution of grid points by resolution level for these two times. As can be seen, the WAMR method uses high resolution only in regions of rapid variation, such as near the shock and detonation waves. For this problem a finest grid size of approximately 0.4  $\mu\text{m}$  was required to satisfy the accuracy requirement, which is just above mean-free path estimates and consistent with findings in [2].

Fig. 22 shows the mass fraction of each of the nine species at 180  $\mu\text{s}$  and compares favorably with previously published results. However, in the current results, additional structures attributed to the use of multi-component and thermal diffusion can be seen, as highlighted previously in the Sod problem. When these effects are included, spikes appear in the mass fractions of Ar,  $\text{H}_2$ , and  $\text{O}_2$ , with a close-up view of the last shown in Fig. 23. The magnified view of the mass fraction of  $\text{O}_2$  shows a small spike at the location of the leading shock of the detonation front. These spikes do not appear if baro- and thermo-diffusive effects are excluded (i.e., by setting  $D_{jk} = D_k^T = 0$ ). Note that spikes are only seen in these three species as all others are only present in trace amounts at the shock. Through the induction and reaction zones, large changes in species concentrations occur, but changes in pressure and temperature due to chemical reactions are quite gradual (especially when compared to the their variation within the shock). As such, the species production due to reactions and Fickian mass diffusion dominates pressure and thermal diffusion, and no spikes are seen.

#### 6. Algorithmic performance

The algorithmic speedup and compression of WAMR is examined by comparing the present method to the non-adaptive method providing equivalent resolution for a two-dimensional test problem. The test problem is a variation of the Taylor/Se-

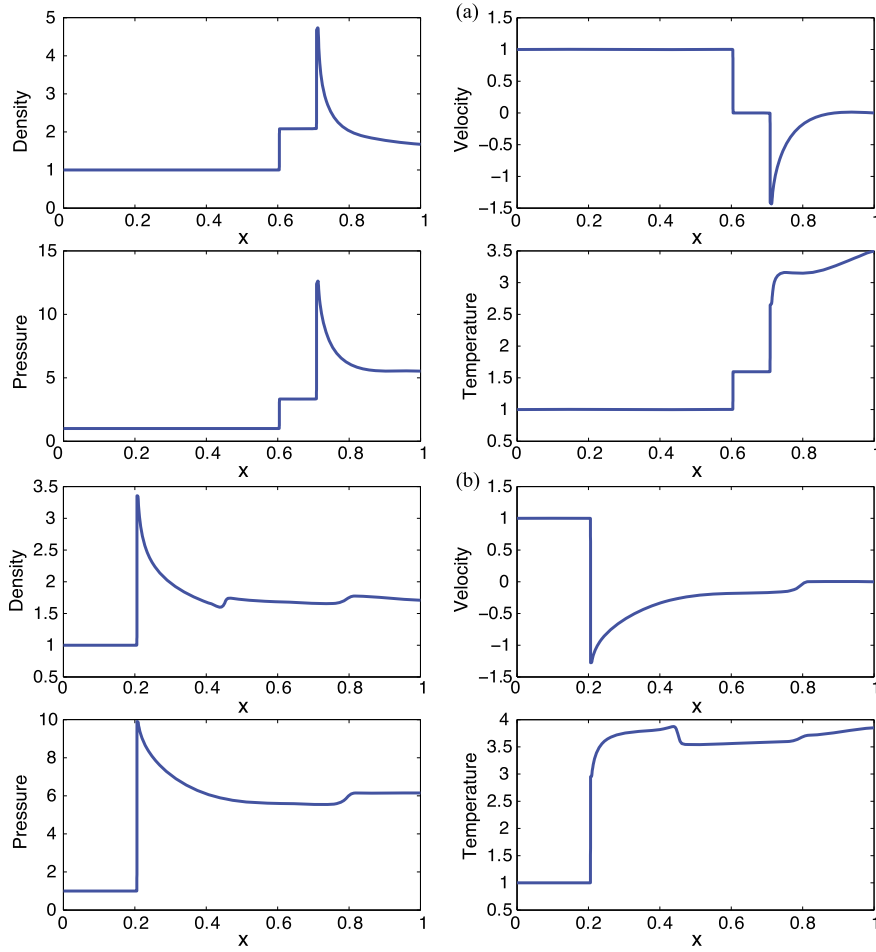


Fig. 19. Solution of viscous detonation at (a)  $t = 180 \mu\text{s}$  and (b)  $t = 230 \mu\text{s}$ .

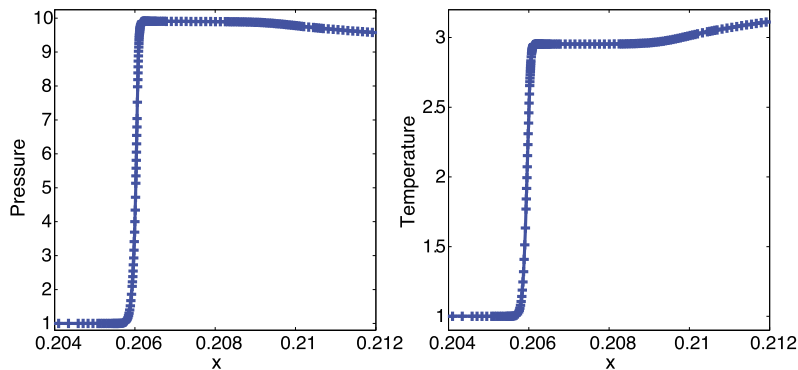


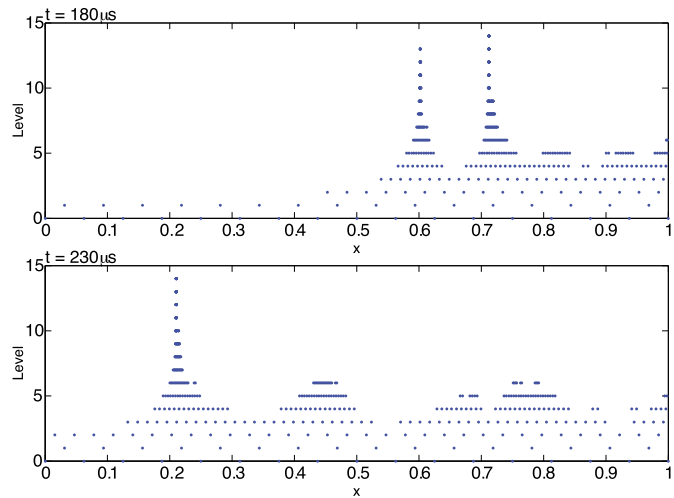
Fig. 20. Close-up view of detonation front pressure and temperature fields at  $t = 230 \mu\text{s}$ .

dov blast wave problem described previously in Section 5.3. The domain is square with  $L_{1,2} = 1.0$ , and a circular region of high pressure with radius  $r_0$  is centered at  $\mathbf{x}_0$ . The initial pressure field is defined by

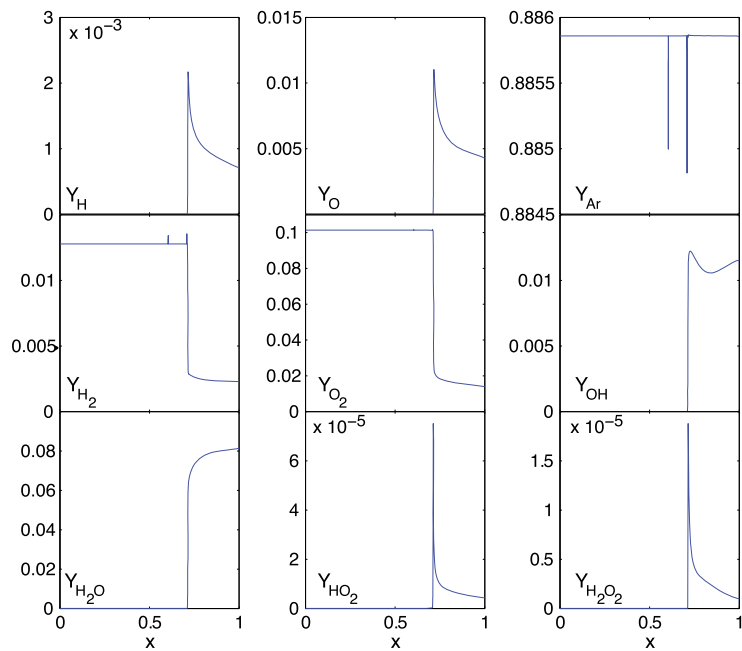
$$p(\mathbf{x}, 0) = p_0 \left\{ 1 + \frac{1}{2}(p_R - 1) \left[ 1 - \tanh\left(\frac{\|\mathbf{x} - \mathbf{x}_0\| - r_0}{\delta}\right) \right] \right\}. \quad (29)$$

The reference pressure and temperature are  $p_0 = 10^3 \text{ dyne/cm}^2$  and  $T_0 = 300 \text{ K}$ , with  $p_R = 20$ . The high pressure region is centered at  $\mathbf{x}_0 = (0.5, 0.5)$  with radius  $r_0 = 0.25$  and  $\delta = 5 \times 10^{-2}$ . The gas mixture is an inert, three-component air mixture of  $78\text{N}_2:21\text{O}_2:\text{Ar}$ .





**Fig. 21.** Adaptive collocation grid for the solution at  $t = 180 \mu\text{s}$  (top) and  $t = 230 \mu\text{s}$  (bottom).



**Fig. 22.** Mass fraction of each species at  $t = 180 \mu\text{s}$ .

**Table 1**  
Comparison of performance between WAMR and uniform grid.

Measures	WAMR/uniform grid
Runtime	0.08
Average No. of grid points	0.13
No. of time steps	0.50
CPU-time/grid-point/time-step	1.19

For the adaptive case, a coarse grid of  $[10 \times 10]$  points and threshold of  $\varepsilon = 10^{-3}$  are used. Fig. 24 shows the number of active collocation points,  $N_A$ , and the maximum refinement level used,  $J$ , by the WAMR method for this test problem. Table 1 summarizes the results of the test problem using the WAMR method and the equivalent non-adaptive method. The average number of points in the adaptive method is  $\langle N_A \rangle = 13,836$  with a maximum refinement level of  $J = 5$ . An equivalent uniform grid requires 321 points in each direction for a total of  $N = 103,041$  points. The WAMR method uses on average 13% of the number of grid points used in the non-adaptive method and completes the simulation over 12 times

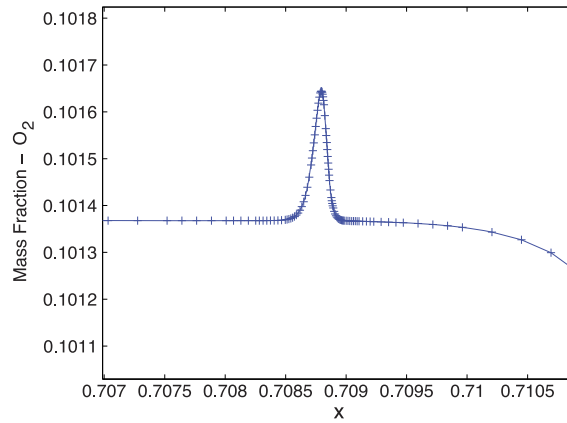


Fig. 23. Close-up of the mass fraction of  $O_2$  at  $t = 180 \mu s$ .

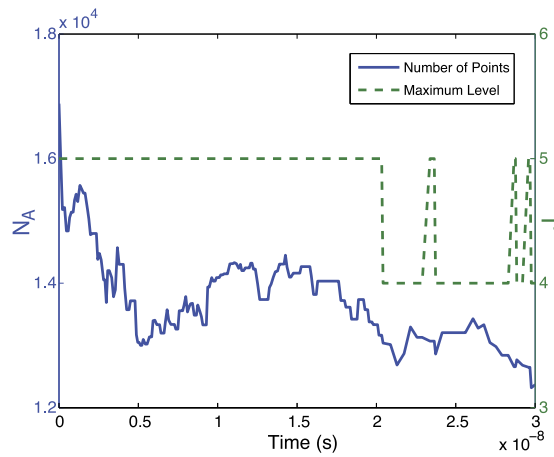


Fig. 24. Number of active collocation points,  $N_A$ , and maximum refinement level,  $J$ , as functions of time used by the WAMR method for the blast wave test problem.

Table 2

Compression ratios for test problems.

Problem	$d$	$J$	$\langle N_A \rangle$	$N$	$C$
Sod shock tube	1	9	132	5120	38
Sod shock tube	2	9	18,473	26,214,400	1419
Taylor/Sedov blast wave	1	11	129	20,480	158
Taylor/Sedov blast wave	2	10	47,767	104,857,600	2195
Taylor/Sedov blast wave	3	9	4,751,411	$1.342 \times 10^{11}$	28,247
$H_2/O_2/Ar$ detonation	1	14	186	262,144	1409

faster. The improvement in overall runtime comes from both the obvious difference in the total number of collocation points used at any time and the time-step dependence on the finest grid resolution. In this problem, the maximum resolution required changes by one level, during which time the WAMR method is able to take time steps twice the size of that of the non-adaptive method. The reduction in the number of time steps is related to the variation in the maximum refinement that is needed; thus for problems whose maximum resolution varies by more than one level, even greater savings are seen. The ratio of CPU-time per grid-point per time-step between the adaptive and non-adaptive methods provides a normalized comparison between the two methods. For each point, the computational cost is larger for the WAMR method by approximately 19% due to the additional expense of performing the adaption itself, including performing the wavelet transforms, rearranging the grid, and interpolations. However, this additional cost is more than compensated by the reduction of the necessary number of points and number of time steps.

Table 2 shows a comparison of the average number of grid points used by the WAMR method  $\langle N_A \rangle$  and the equivalently refined uniform grid of size  $N$  for each of the test problems in the previous section. As can be seen, the compression ratio

$C = N/\langle N_A \rangle$  increases as both the maximum refinement level and problem dimensionality increase. The compression ratios range from  $O(10)$  for 1-D problems up to  $O(10^4)$  for 3-D problems, but note that the ratio also depends on the choice of  $\varepsilon$ .

## 7. Conclusions

Wavelet amplitudes of the solution provide a direct measure of local approximation error at collocation points, and by thresholding grid points according to their amplitudes, the WAMR method provides a robust means for driving spatial grid adaptivity. Furthermore, this method of adaption does not rely on *ad hoc* error estimators, and can be seen to provide automatically verified solutions. The compressible reactive flow equations are discretized on a sparse, irregular grid generated by the WAMR method, and solved using finite difference approximations for derivatives at those points, and any appropriate ODE integrator.

An adaptive algorithm for solving the reactive flow equations in  $d$ -dimension is developed. The connection between the order of wavelets, the order of corresponding Lagrange interpolants used to approximate derivatives, the threshold values, and the accuracy of the numerical solution is numerically investigated. It is found that the relationships between these quantities behave similarly to those associated with the sparse wavelet approximation of a function and with derivative approximations on an irregular grid.

The focus of this work is the application of the WAMR algorithm on the simulation of compressible reactive flows. The compressible, reactive balance equations are implemented in a general  $d$ -dimensional conservative form. Detailed models for diffusive transport are employed, including baro- and thermo-diffusion of species, Sorét and Dufour effects. Detailed mass-action kinetics are employed with thermodynamic and transport properties evaluated using the Chemkin and Transport libraries. Common boundary conditions for compressible, reactive flows are also implemented and include walls, symmetry, periodic, and locally one-dimensional non-reflecting conditions.

The WAMR method is applied to several test problems. Non-reactive problems in 1-, 2-, and 3-D include the classic Sod shock tube problem and the Taylor–Sedov blast wave. In both cases, dissipative solutions from the WAMR method agree well with the non-dissipative analytic solutions; however, additional structures are seen in the dissipative numerical solutions, where the effect of detailed multicomponent diffusion produces small concentration variations among species as they preferentially diffuse along pressure and temperature gradients. A 1-D, reactive viscous detonation in an  $\text{H}_2\text{:O}_2\text{:Ar}$  mixture is also computed. Like the inert test problems, small inhomogeneities in specie mass fractions due to multicomponent diffusion are also seen in this case. It is emphasized that, for a given error tolerance, WAMR resolves all physical/chemical characteristics present in the model being solved, thus providing a verified numerical solution for which numerical dissipation is much smaller than the physical dissipation.

Performance of the algorithms is quantified using a 2-D Taylor–Sedov blast wave problem. The algorithm is found to be over an order of magnitude faster than the non-adaptive method. In addition, it uses a fraction of the total number of grid points. This point is particularly salient as large parallel computational systems move towards architectures with less memory available per core. The second part of this work [40] addresses algorithmic aspects necessary for implementation of WAMR in parallel architectures.

It should be noted that the current work only deals with the problem of spatial adaptivity. A complementary method of temporal adaptivity would provide the best reduction in computational demand for this type of multiscale problem. There are various approaches in this direction such as the use of adaptive time-stepping, as seen in adaptive mesh refinement algorithms [66,67], or the *G-scheme* [68], but this aspect is left as possible future work.

## Acknowledgements

Support of this work was provided by National Aeronautics and Space Administration (NASA) under Grant No. NNX07AD10A; this support is gratefully acknowledged. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Lastly, the authors would like to thank Mr. Temistocle Grenga for verifying some of the numerical results.

## Appendix A. Equations of state and constitutive equations

### A.1. Equations of state

The thermal equation of state for a perfect gas mixture obeying Dalton's law is

$$\rho = \frac{p\bar{M}}{\Re T}, \quad (\text{A.1})$$

where  $\Re$  is the universal gas constant,  $T$  is the temperature, and  $\bar{M}$  is the mixture molar mass that is defined by

$$\bar{M} \equiv \sum_{k=1}^K X_k M_k, \quad (\text{A.2})$$

where  $X_k$  and  $M_k$  are the mole fraction and molecular mass of species  $k$ , respectively, and  $Y_k = (M_k/\bar{M})X_k$ .

The mixture specific internal energy is found from the caloric equation of state

$$e = \sum_{k=1}^K \frac{Y_k}{M_k} (h_k^0 - \Re T), \quad (\text{A.3})$$

with

$$h_k^0 = h_{T_{\text{ref}},k}^0 + \int_{T_{\text{ref}}}^T c_{p,k}(\hat{T}) d\hat{T}, \quad (\text{A.4})$$

where  $h_k^0$  is the enthalpy of the  $k$ th specie with the superscript 0 denoting evaluation at a reference pressure  $p_{\text{ref}}$  with reference temperature  $T_{\text{ref}}$ . The specific heat at constant pressure of the  $k$ th specie,  $c_{p,k}$ , is determined from a polynomial fit of thermodynamic data whose coefficients are provided in the Chemkin thermodynamics database [61] or NASA Glenn thermodynamic database [69]. The mixture temperature in (A.3) is determined iteratively from a given value of  $e$  using a Newton root-finding procedure adapted from [53].

## A.2. Diffusive transport

The form of diffusive transport models used follow those of Bird, et al. [41] and Kee, et al. [70]. Models used to determine transport properties are summarized, and correspond to those used in [71], where further details and additional references may be found.

The viscous stress tensor  $\tau_{ij}$  for a Newtonian fluid is given as

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \left( \kappa - \frac{2}{3}\mu \right) \frac{\partial u_k}{\partial x_k} \delta_{ij}, \quad (\text{A.5})$$

where the shear viscosity  $\mu$  and the bulk viscosity  $\kappa$  of the mixture, are generally functions of temperature and composition. The mixture shear viscosity is determined from the Wilke formula

$$\mu = \sum_{k=1}^K \frac{X_k \mu_k}{\sum_{j=1}^K X_j \Phi_{jk}}, \quad \Phi_{jk} = \frac{1}{\sqrt{8}} \left( 1 + \frac{M_j}{M_k} \right)^{-\frac{1}{2}} \left[ 1 + \left( \frac{\mu_j}{\mu_k} \right)^{\frac{1}{2}} \left( \frac{M_k}{M_j} \right)^{\frac{1}{4}} \right]^2, \quad (\text{A.6})$$

where  $\mu_k$  is the viscosity of specie  $k$ . The pure-species viscosities are, in turn, obtained from an  $N - 1$  order polynomial fit of the logarithm of temperature

$$\ln \mu_k = \sum_{n=1}^N a_{n,k} (\ln T)^{n-1}, \quad (\text{A.7})$$

where  $a_{n,k}$  are the polynomial coefficients for specie  $k$ . The bulk viscosity  $\kappa$  is important in mixtures containing polyatomic species, and represents the relaxation of energy from translational to internal molecular modes. Monatomic gases have no internal degrees of freedom, and thus the bulk viscosity is exactly zero for them [72]. Kinetic theory models for bulk viscosity, as well as other properties, can be found in [73] and [62], and solution techniques using iterative linear solvers, implemented in a Fortran library [74] are available. In the present work, since no information regarding its value is available in the Transport library used [71], it is taken to be zero.

The heat flux vector  $q_i$  contains, first, a standard Fourier's law term and, second, a Sorét effect term accounting for energy transport due to mass diffusion with thermal diffusion coefficients  $D_k^T$ :

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \sum_{k=1}^K \left( j_{i,k} h_k - \frac{\Re T}{M_k X_k} D_k^T d_{i,k} \right). \quad (\text{A.8})$$

The species mass flux vector  $j_{i,k}$  contains ordinary multicomponent mass diffusion with diffusion coefficients  $D_{jk}$  and mass transport due to thermal gradients, known as the Dufour effect:

$$j_{i,k} = \frac{\rho Y_k}{X_k \bar{M}} \sum_{\substack{j=1 \\ j \neq k}}^K M_j D_{kj} d_{i,j} - D_k^T \frac{1}{T} \frac{\partial T}{\partial x_i}, \quad (\text{A.9})$$

where

$$d_{i,k} = \frac{\partial X_k}{\partial x_i} + (X_k - Y_k) \frac{1}{p} \frac{\partial p}{\partial x_i}. \quad (\text{A.10})$$

Multicomponent diffusion coefficients  $D_{jk}$ , thermal diffusion coefficients  $D_k^T$ , and mixture thermal conductivity  $\lambda$ , are determined from the solution of a linear system of equations defined by a  $3K \times 3K$  matrix, and is described in [71]. Note that the use of full detailed transport properties requires the solution of the system at each collocation point in the discretized problem.

### A.3. Detailed chemical kinetics

The present work uses a detailed chemical kinetics mechanism with specific data evaluated using Chemkin [61]. A general mechanism consists of  $I$  elementary reversible reactions

$$\sum_{k=1}^K \nu'_{ki} S_k \rightleftharpoons \sum_{k=1}^K \nu''_{ki} S_k, \quad i = 1, \dots, I, \quad (\text{A.11})$$

where  $\nu'_{ki}$  and  $\nu''_{ki}$  are reactant and product stoichiometric coefficients for the  $k$ th specie  $S_k$  in reaction  $i$ , and the net stoichiometric coefficient is defined as  $\nu_{ki} = \nu''_{ki} - \nu'_{ki}$ . Each reaction has a forward rate constant  $k_i^f$  described by the Arrhenius form

$$k_i^f = A_i T^{\beta_i} \exp\left(-\frac{E_i}{\mathfrak{R}T}\right), \quad (\text{A.12})$$

where the pre-exponential or frequency factor  $A_i$ , temperature exponent  $\beta_i$ , and activation energy  $E_i$  are provided by the chemical mechanism. The reverse reaction rate constants  $k_i^r$  are obtained from the ratio of forward rate and equilibrium constants  $K_i^e$ :

$$k_i^r = \frac{k_i^f}{K_i^e}. \quad (\text{A.13})$$

The equilibrium constants are given by

$$K_i^e = \left(\frac{p_{\text{ref}}}{\mathfrak{R}T}\right)^{\sum_{k=1}^K \nu_{ki}} \exp\left(\frac{\Delta s_i^0}{\mathfrak{R}} - \frac{\Delta h_i^0}{\mathfrak{R}T}\right), \quad (\text{A.14})$$

where  $\Delta s_i^0$  and  $\Delta h_i^0$  are the change in entropy and enthalpy for a complete change of reactants to products in reaction  $i$ ,

$$\frac{\Delta s_i^0}{\mathfrak{R}} = \sum_{k=1}^K \nu_{ki} \frac{s_k^0}{\mathfrak{R}} \quad \text{and} \quad \frac{\Delta h_i^0}{\mathfrak{R}T} = \sum_{k=1}^K \nu_{ki} \frac{h_k^0}{\mathfrak{R}T}, \quad (\text{A.15})$$

with the enthalpy given by (A.4) and the entropy defined as

$$s_k^0 = s_{T_{\text{ref}},k}^0 + \int_{T_{\text{ref}}}^T \frac{c_{p,k}(\hat{T})}{\hat{T}} d\hat{T}. \quad (\text{A.16})$$

The rate of progress of each reaction,  $r_i$ , is then found from

$$r_i = k_i^f \prod_{k=1}^K [X_k]^{\nu'_{ki}} - k_i^r \prod_{k=1}^K [X_k]^{\nu''_{ki}}, \quad (\text{A.17})$$

where  $[X_k]$  denotes the molar concentration of specie  $k$ . The molar production rate of specie  $k$  can then be obtained from

$$\dot{\omega}_k = \sum_{i=1}^I \nu_{ki} r_i. \quad (\text{A.18})$$

## Appendix B. Detailed hydrogen–oxygen reaction mechanism [65]

<i>i</i>	Reaction	$A_i \left( \frac{\text{mol}}{\text{cm}^3 \text{ s K}^{\beta_i}} \right)$	$\beta_i$	$E_i \left( \frac{\text{kJ}}{\text{mol}} \right)$
1	$\text{O}_2 + \text{H} \rightarrow \text{OH} + \text{O}$	$2.00 \times 10^{14}$	0.00	70.30
2	$\text{OH} + \text{O} \rightarrow \text{O}_2 + \text{H}$	$1.46 \times 10^{13}$	0.00	2.08
3	$\text{H}_2 + \text{O} \rightarrow \text{OH} + \text{H}$	$5.06 \times 10^4$	2.67	26.30
4	$\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{O}$	$2.24 \times 10^4$	2.67	18.40
5	$\text{H}_2 + \text{OH} \rightarrow \text{H}_2\text{O} + \text{H}$	$1.00 \times 10^8$	1.60	13.80
6	$\text{H}_2\text{O} + \text{H} \rightarrow \text{H}_2 + \text{OH}$	$4.45 \times 10^8$	1.60	77.13
7	$\text{OH} + \text{OH} \rightarrow \text{H}_2\text{O} + \text{O}$	$1.50 \times 10^9$	1.14	0.42
8	$\text{H}_2\text{O} + \text{O} \rightarrow \text{OH} + \text{OH}$	$1.51 \times 10^{10}$	1.14	71.64
9	$\text{H} + \text{H} + \text{M} \rightarrow \text{H}_2 + \text{M}$	$1.80 \times 10^{18}$	−1.00	0.00
10	$\text{H}_2 + \text{M} \rightarrow \text{H} + \text{H} + \text{M}$	$6.99 \times 10^{18}$	−1.00	436.08
11	$\text{H} + \text{OH} + \text{M} \rightarrow \text{H}_2\text{O} + \text{M}$	$2.20 \times 10^{22}$	−2.00	0.00
12	$\text{H}_2\text{O} + \text{M} \rightarrow \text{H} + \text{OH} + \text{M}$	$3.80 \times 10^{23}$	−2.00	499.41
13	$\text{O} + \text{O} + \text{M} \rightarrow \text{O}_2 + \text{M}$	$2.90 \times 10^{17}$	−1.00	0.00
14	$\text{O}_2 + \text{M} \rightarrow \text{O} + \text{O} + \text{M}$	$6.81 \times 10^{18}$	−1.00	496.41
15	$\text{H} + \text{O}_2 + \text{M} \rightarrow \text{HO}_2 + \text{M}$	$2.30 \times 10^{18}$	−0.80	0.00
16	$\text{HO}_2 + \text{M} \rightarrow \text{H} + \text{O}_2 + \text{M}$	$3.26 \times 10^{18}$	−0.80	195.88
17	$\text{HO}_2 + \text{H} \rightarrow \text{OH} + \text{OH}$	$1.50 \times 10^{14}$	0.00	4.20
18	$\text{OH} + \text{OH} \rightarrow \text{HO}_2 + \text{H}$	$1.33 \times 10^{13}$	0.00	168.30
19	$\text{HO}_2 + \text{H} \rightarrow \text{H}_2 + \text{O}_2$	$2.50 \times 10^{13}$	0.00	2.90
20	$\text{H}_2 + \text{O}_2 \rightarrow \text{HO}_2 + \text{H}$	$6.84 \times 10^{13}$	0.00	243.10
21	$\text{HO}_2 + \text{H} \rightarrow \text{H}_2\text{O} + \text{O}$	$3.00 \times 10^{13}$	0.00	7.20
22	$\text{H}_2\text{O} + \text{O} \rightarrow \text{HO}_2 + \text{H}$	$2.67 \times 10^{13}$	0.00	242.52
23	$\text{HO}_2 + \text{O} \rightarrow \text{OH} + \text{O}_2$	$1.80 \times 10^{13}$	0.00	−1.70
24	$\text{OH} + \text{O}_2 \rightarrow \text{HO}_2 + \text{O}$	$2.18 \times 10^{13}$	0.00	230.61
25	$\text{HO}_2 + \text{OH} \rightarrow \text{H}_2\text{O} + \text{O}_2$	$6.00 \times 10^{13}$	0.00	0.00
26	$\text{H}_2\text{O} + \text{O}_2 \rightarrow \text{H}_2\text{O} + \text{OH}$	$7.31 \times 10^{14}$	0.00	303.53
27	$\text{HO}_2 + \text{HO}_2 \rightarrow \text{H}_2\text{O}_2 + \text{O}_2$	$2.50 \times 10^{11}$	0.00	−5.20
28	$\text{OH} + \text{OH} + \text{M} \rightarrow \text{H}_2\text{O}_2 + \text{M}$	$3.25 \times 10^{22}$	−2.00	0.00
29	$\text{H}_2\text{O}_2 + \text{M} \rightarrow \text{OH} + \text{OH} + \text{M}$	$2.10 \times 10^{24}$	−2.00	206.80
30	$\text{H}_2\text{O}_2 + \text{H} \rightarrow \text{H}_2 + \text{HO}_2$	$1.70 \times 10^{12}$	0.00	15.70
31	$\text{H}_2 + \text{HO}_2 \rightarrow \text{H}_2\text{O}_2 + \text{H}$	$1.15 \times 10^{12}$	0.00	80.88
32	$\text{H}_2\text{O}_2 + \text{H} \rightarrow \text{H}_2\text{O} + \text{OH}$	$1.00 \times 10^{13}$	0.00	15.00
33	$\text{H}_2\text{O} + \text{OH} \rightarrow \text{H}_2\text{O}_2 + \text{H}$	$2.67 \times 10^{12}$	0.00	307.51
34	$\text{H}_2\text{O}_2 + \text{O} \rightarrow \text{OH} + \text{HO}_2$	$2.80 \times 10^{13}$	0.00	26.80
35	$\text{OH} + \text{HO}_2 \rightarrow \text{H}_2\text{O}_2 + \text{O}$	$8.40 \times 10^{12}$	0.00	84.09
36	$\text{H}_2\text{O}_2 + \text{OH} \rightarrow \text{H}_2\text{O} + \text{HO}_2$	$5.40 \times 10^{12}$	0.00	4.20
37	$\text{H}_2\text{O} + \text{HO}_2 \rightarrow \text{H}_2\text{O}_2 + \text{OH}$	$1.63 \times 10^{13}$	0.00	132.71

Enhanced third-body efficiencies with *M*:  $f_{\text{H}_2} = 1.00$ ,  $f_{\text{O}_2} = 0.35$ ,  $f_{\text{H}_2\text{O}} = 6.5$ .

## References

- [1] Department of Energy Office of Science, Basic research needs for clean and efficient combustion of 21st century transportation fuels, [http://www.sc.doe.gov/bes/reports/files/CTF\\_rpt.pdf](http://www.sc.doe.gov/bes/reports/files/CTF_rpt.pdf), 2006.
- [2] J.M. Powers, S. Paolucci, Accurate spatial resolution estimates for reactive supersonic flow with detailed chemistry, *AIAA J.* 43 (2005) 1088–1099.
- [3] M.J. Berger, J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512.
- [4] P. Fernandes, P. Girdino, P. Molfino, M. Repetto, A comparison of adaptive strategies for mesh refinement based on “a posteriori” local error estimation procedures, *IEEE Trans. Magn.* 26 (1990) 795–798.
- [5] O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, *The Finite Element Method for Fluid Dynamics*, 6th ed., Elsevier, 2005.
- [6] D.P. Young, R.G. Melvin, M.B. Bieterman, A locally refined rectangular grid finite element method: Application to computational dynamics and computational physics, *J. Comput. Phys.* 92 (1991) 1–66.
- [7] W. Gui, I. Babuska, The h, p, and h–p versions of the finite element method in 1 dimension. Part I. The error analysis of the p-version, *Numer. Math.* 49 (1986) 577–612.
- [8] W. Gui, I. Babuska, The h, p, and h–p versions of the finite element method in 1 dimension. Part II. The error analysis of the h- and h–p versions, *Numer. Math.* 49 (1986) 613–657.
- [9] L. Demkowicz, J.T. Oden, W. Rachowicz, O. Hardy, Toward a universal h–p adaptive finite element strategy, Part 1. Constrained approximation and data structure, *Comput. Methods Appl. Mech. Eng.* 77 (1989) 79–112.
- [10] A. Rajagopal, S.M. Sivakumar, A combined r–h adaptive strategy based on material forces and error assessment for plane problems and bimaterial interfaces, *Comput. Mech.* 41 (2007) 49–72.
- [11] I. Babuska, W.C. Rheinboldt, Error estimates for adaptive finite element computations, *SIAM J. Numer. Anal.* 15 (1978) 736–754.
- [12] J.T. Oden, L. Demkowicz, W. Rachowicz, T.A. Westermann, Toward a universal h–p adaptive finite element strategy, Part 2. A posteriori error estimation, *Comput. Methods Appl. Mech. Eng.* 77 (1989) 113–180.
- [13] C.A. Duarte, J.T. Oden, An h–p adaptive method using clouds, *Comput. Methods Appl. Mech. Eng.* 139 (1996) 237–262.
- [14] K. Eriksson, C. Johnson, Adaptive finite element methods for parabolic problems I: A linear model problem, *SIAM J. Numer. Anal.* 28 (1991) 43–77.
- [15] N.G. Pantelelis, A.E. Kanarachos, The parallel block adaptive multigrid method for the implicit solution of the Euler equations, *Int. J. Numer. Methods Fluids* 22 (1996) 411–428.
- [16] G.W. Zumbusch, A sparse grid PDE solver, in: H.P. Langtangen, A.M. Bruaset, E. Quak (Eds.), *Advances in Software Tools for Scientific Computing*, Springer, 2000, pp. 133–177.



- [17] M. Griebel, G.W. Zumbusch, Adaptive sparse grids for hyperbolic conservation laws, in: M. Fey, R. Jeltsch (Eds.), *Hyperbolic Problems: Theory, Numerics, Applications*, 7th International Conference in Zuerich, February 1998, Birkhäuser, 1999, pp. 411–422.
- [18] K. Schneider, O.V. Vasilyev, Wavelet methods in computational fluid dynamics, *Annu. Rev. Fluid Mech.* 42 (2010) 473–503.
- [19] Y. Maday, V. Perrier, J.-C. Ravel, Dynamically adaptive using wavelets basis for the approximation of partial differential equations, *C. R. Acad. Sci. Paris, Ser. I* 312 (1991) 405–410.
- [20] J. Liandrat, V. Perrier, P. Tchamitchian, Numerical resolution of the regularized Burgers equations using the wavelet transform, in: *Wavelets and Applications*, vol. 20, 1991, pp. 420–433.
- [21] K. Amaratunga, J.R. Williams, S. Qian, J. Weiss, Wavelet-Galerkin solutions for one-dimensional partial differential equations, *Int. J. Numer. Methods Eng.* 37 (1994) 2703–2716.
- [22] J.C. Xu, W.C. Shann, Galerkin-wavelet methods for 2-point boundary-value problems, *Numer. Math.* 63 (1992) 123–144.
- [23] O.V. Vasilyev, S. Paolucci, A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, *J. Comput. Phys.* 125 (1996) 498–512.
- [24] O.V. Vasilyev, S. Paolucci, A fast adaptive wavelet collocation algorithm for multidimensional PDEs, *J. Comput. Phys.* 138 (1997) 16–56.
- [25] S. Bertoluzza, Adaptive wavelet collocation method for the solution of Burgers equation, *Transp. Theory Stat. Phys.* 25 (1996) 339–359.
- [26] O.V. Vasilyev, S. Paolucci, Dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, *J. Comput. Phys.* 125 (1996) 498–512.
- [27] M. Domingues, S. Gomes, L. Diaz, Adaptive wavelet representation and differentiation on block-structured grids, in: 4th Pan-American Workshop on Applied and Computational Mathematics, Cordoba, Argentina, Jul 01–05 2002, in: *Appl. Numer. Math.*, vol. 47, Univ. Nacl. Cordoba, 2003, pp. 421–437.
- [28] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, *J. Comput. Phys.* 188 (2003) 493–523.
- [29] S. Mueller, Y. Stiriba, A multilevel finite volume method with multiscale-based grid adaptation for steady compressible flows, *J. Comput. Appl. Math.* 227 (2009) 223–233.
- [30] Y.A. Rastigejev, S. Paolucci, Wavelet adaptive multiresolution representation: Applications to viscous multiscale flow simulations, *Int. J. Wavelets Multiresolut. Inf. Process.* 4 (2006) 333–343.
- [31] Y.A. Rastigejev, S. Paolucci, Wavelet-based adaptive multiresolution computation of viscous reactive flows, *Int. J. Numer. Methods Fluids* 52 (2006) 749–784.
- [32] D. Wirasaet, S. Paolucci, Adaptive wavelet method for incompressible flows in complex domains, *J. Fluids Eng.* 127 (2005) 656–665.
- [33] M. Griebel, F. Koster, Adaptive wavelet solvers for the unsteady incompressible Navier–Stokes equations, in: *Advances in Mathematical Fluid Mechanics*, 2000, pp. 67–118.
- [34] N.K.-R. Kevlahan, O.V. Vasilyev, An adaptive wavelet collocation method for fluid–structure interaction at high Reynolds numbers, *SIAM J. Sci. Comput.* 26 (2005) 1894–1915.
- [35] O.V. Vasilyev, N.K.-R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems, *J. Comput. Phys.* 206 (2005) 412–431.
- [36] S.J. Reckinger, D. Livescu, O.V. Vasilyev, Adaptive wavelet collocation method simulations of Rayleigh–Taylor instability, *Phys. Scr. T* 142 (2010) 0146064.
- [37] M. Farge, Wavelet transforms and their applications to turbulence, *Annu. Rev. Fluid Mech.* 24 (1992) 395–457.
- [38] M. Farge, K. Schneider, N. Kevlahan, Non-gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis, *Phys. Fluids* 11 (1999) 2187–2201.
- [39] M. Farge, K. Schneider, Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets, *Flow Turbul. Combust.* 66 (2001) 393–426.
- [40] S. Paolucci, Z. Zikoski, T. Grenga, WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. Part II. The parallel algorithm, *J. Comput. Phys.* (2012), submitted for publication.
- [41] R.B. Bird, W.E. Stewart, E.N. Lightfoot, *Transport Phenomena*, John Wiley & Sons, Inc., 1960.
- [42] R. Prosser, R.S. Cant, Wavelet methods in computational combustion, in: T. Eschekki, E. Mastorakos (Eds.), *Turbulent Combustion Modeling*, in: *Fluid Mech. Appl.*, vol. 95, Springer, Netherlands, 2011, pp. 331–351.
- [43] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.
- [44] S. Dubuc, Interpolation through an iterative scheme, *J. Math. Anal. Appl.* 114 (1986) 185–204.
- [45] G. Deslauriers, S. Dubuc, Symmetric iterative interpolation processes, *Constr. Approx.* 5 (1989) 49–68.
- [46] N. Saito, G. Beylkin, Multiresolution representations using the auto-correlation functions of compactly supported wavelets, *IEEE Trans. Signal Process.* 41 (1993) 3584–3590.
- [47] D. Donoho, Interpolating wavelet transform, Report, Department of Statistics, Stanford University, 1992.
- [48] S. Bertoluzza, G. Naldi, J.C. Ravel, Wavelet methods for the numerical solution of boundary value problems on the interval, in: C.K. Chui, L. Montefusco, L. Puccio (Eds.), *Wavelets: Theory, Algorithms, and Applications*, vol. 5, Academic Press Inc., 1994, pp. 425–448.
- [49] F. Koster, A proof of the consistency of the finite difference technique on sparse grids, *Computing* 65 (2000) 247–261.
- [50] D. Wirasaet, Numerical solutions of multidimensional partial differential equations using an adaptive wavelet method, Ph.D. in Mechanical Engineering, University of Notre Dame, Notre Dame, IN, 2008.
- [51] M. Holmström, Solving hyperbolic PDEs using interpolating wavelets, *SIAM J. Sci. Comput.* 21 (1999) 405–420.
- [52] W.E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press Ltd., San Diego, 1991.
- [53] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing*, second edition, Cambridge University Press, 1992.
- [54] E. Tadmor, Numerical viscosity and the entropy condition for conservative difference schemes, *Math. Comput.* 43 (1984) 369–381.
- [55] E.S. Oran, V.N. Gamezo, Origins of the deflagration-to-detonation transition in gas-phase combustion, *Combust. Flame* 148 (2007) 4–47.
- [56] K. Mazaheri, Y. Mahmoudi, M.I. Radulescu, Diffusion and hydrodynamic instabilities in gaseous detonations, *Combust. Flame* 159 (2012) 2138–2154.
- [57] C.M. Romick, T.D. Aslam, J.M. Powers, The effect of diffusion on the dynamics of unsteady detonations, *J. Fluid Mech.* 699 (2012) 453–464.
- [58] P.J. Roache, *Fundamentals of Verification and Validation*, Hermosa Publishers, Socorro, NM, 2009.
- [59] S.A. Sod, Survey of several finite-difference methods for systems of non-linear hyperbolic conservation laws, *J. Comput. Phys.* 27 (1977) 1–30.
- [60] R.J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, 1992.
- [61] R.J. Kee, F.M. Rupley, J.A. Miller, Chemkin-II: A Fortran chemical kinetics package for the analysis of gas phase chemical kinetics, Technical Report SAND89-8009B, Sandia National Laboratories, Albuquerque, New Mexico, 1992.
- [62] A. Ern, V. Giovangigli, Multicomponent Transport Algorithms, *Lect. Notes Phys.*, M Monogr., vol. m24, Springer-Verlag, 1994.
- [63] G.I. Taylor, The formation of a blast wave by a very intense explosion. I. Theoretical discussion, *Proc. R. Soc. Lond. A* 201 (1950) 159–174.
- [64] L.I. Sedov, Propagation of strong shock waves, *J. Appl. Math. Mech.* 10 (1946) 241–250.
- [65] S. Singh, Y. Rastigejev, S. Paolucci, J.M. Powers, Viscous detonation in  $H_2$ – $O_2$ –Ar using intrinsic low-dimensional manifolds and wavelet adaptive multilevel representation, *Combust. Theory Model.* 5 (2001) 163–184.
- [66] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [67] A.M. Khokhlov, Fully thread tree algorithms for adaptive refinement fluid dynamics simulations, *J. Comput. Phys.* 143 (1998) 519–543.

- [68] M. Valorani, S. Paolucci, The G-scheme: A framework for multi-scale adaptive model reduction, *J. Comput. Phys.* 228 (2009) 4665–4701.
- [69] B.J. McBride, M.J. Zehe, S. Gordon, NASA Glenn coefficients for calculating thermodynamic properties of individual species, Technical Report E-13336, NASA Glenn Research Center, 2002.
- [70] R.J. Kee, M.E. Coltrin, P. Glarborg, *Chemically Reactive Flow*, 2nd edition, John Wiley & Sons Publication, New York, 1986.
- [71] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, J.A. Miller, A Fortran computer code package for the evaluation of gas-phase multicomponent transport properties, Technical Report SAND86-8246, Sandia National Laboratories, Albuquerque, New Mexico, 1991.
- [72] G. Billet, V. Giovangigli, G. de Gassowski, Impact of volume viscosity on a shock–hydrogen-bubble interaction, *Combust. Theory Model.* 12 (2008) 221–248.
- [73] L. Monchick, K.S. Yun, E.A. Mason, Formal kinetic theory of transport phenomena in polyatomic gas mixtures, *J. Chem. Phys.* 39 (1963) 654–669.
- [74] A. Ern, V. Giovangigli, EGLib: A general-purpose Fortran library for multicomponent transport property evaluation, <http://www.cmap.polytechnique.fr/www.eglib/home.html>, 2004.