Wavelet Methods and Adaptive Grids in One-Dimensional Movable Boundary Problems

Zhiyu Xu,1,2 Yonghua Tan,2,3 and Xiaoming Li1

1Xi’an Aerospace Propulsion Institute, CASC, Xi’an 710100, China
2Science and Technology on Liquid Rocket Engine Laboratory, CASC, Xi’an 710100, China
3Academy of Aerospace Propulsion Technology, CASC, Xi’an 710100, China

Correspondence should be addressed to Zhiyu Xu; xuzhiyu611@163.com

Received 19 November 2019; Accepted 30 January 2020; Published 9 March 2020

Copyright © 2020 Zhiyu Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Adaptive wavelet collocation methods use wavelet transform and filtering to generate adaptive grids. However, if the boundary moves, the grid becomes aberrant. It baffles wavelet transform and makes the adaptive wavelet methods lose advantages on computational efficiency. This paper develops a series of methods or skills to effectively perform wavelet transform and to generate adaptive grids for one-dimensional movable boundary problems. The methods remain the original inner grid points and keep the grid in the original-nested structure, in order to remain scanty during the whole computing process. For boundary extending, the adaptive wavelet program begins to run on the very new grid beyond the original boundary once it reaches a nested structure, which is called the Intermittent Adaptive Method as a consequence. If the boundary extends tremendously, the new nested grids can be combined to a greater nested grid for further efficiency, which is named the Grid Combine Method. While for boundary contracting, a fictitious boundary is addressed to replace the original boundary that will recede, so wavelet transform can be successfully performed on the original nested grid. Finally, two numerical tests, local features moving and gas gun, were resolved and discussed to show the evolution process of the adaptive grids with the boundaries moving. For boundary contracting, the valid points decrease because the computation domain recedes; while for boundary extending, the valid point numbers vary between a range that almost remains unchanged.

1. Introduction

The adaptive wavelet methods [1] have been researched and employed in Computational Fluid Dynamics (CFD) for about thirty years to pursue high-computational efficiency and resolution. Early in 1986, Farge et al. introduced wavelets and multiresolution analysis into fluid mechanics research [2, 3], while it was essentially to process and analyze the data of high-resolution simulation. And it is still excellent on complex turbulence research today. Farge et al. suggested to resolve partial differential equations (PDEs) using wavelets [3], and then a lot of studies were conducted to solve the basic problems, including wavelet decomposition and adaptive grids, boundary conditions and boundary effect problems, and differentiation matrix. Vasilyev and Bowman [1] introduced second-generation wavelets [4] into the Adaptive Wavelet Collocation Method (AWCM). The second-generation wavelets make wavelet decomposition faster and dealing with boundary conditions easier. In recent years, there are many studies on adaptive wavelet methods developed for computing shock waves and turbulent flows, and the latter look more prolific.

About from year 1992 to 2000, wavelets were used as an excellent compression algorithm to reduce computational freedoms, so as to directly economize computational costs. Harten [5] demonstrated that the magnitude of a wavelet coefficient is of the same order as the jump discontinuity, so it is intuitionistic to recognize and trace shock waves through wavelet coefficients. Then, many studies were naturally conducted to compress grids by wavelets and to perform iterating by traditional schemes, e.g., HLLC, ENO, WENO, MUSCL, and some other new
schemes proposed in recent years [6]. But it induces collisions when the grid points used by traditional schemes are out of the irregular grids. Regele et al. [6, 7] proposed a simple and robust method for computation of shock waves based on AWCM, which is absolutely different from all the former studies because it uses not only wavelets to trace shock waves or generate adaptive grids but also to control artificial viscosity to restrain numerical oscillations. In recent years, the method had been applied to simulate combustions, initiations, and detonations successfully [7–9].

On the other hand, the wavelet methods on turbulence numerical simulation may be more potential and valuable, for that fine turbulent flow structures need extremely fine grids, while the computer techniques are quite unacceptable. In order to reduce computational costs, wavelet threshold filtering and traditional turbulence models are integrated to generate many useful methods, e.g., Wavelet Direct Numerical Simulation (W-DNS), Coherent Vortex Simulation (CVS), and Stochastic Coherent Adaptive Large Eddy Simulation (SCALES). However, it has been recognized that these methods would be applied in other fields, e.g., astrophysics and oceanography [10–16].

All the studies reviewed above focus on fixed boundary problems. But what happens when boundary moves, e.g., piston and bullet? The nest property is destroyed when any point is added or deleted to the original boundary. So, it turns complex and costly. As we searched, there are few studies about wavelet methods on movable boundary problems (MBPs). Of course, it is very possible that there were some excellent studies but neglected by us. Anyway, this work is to give a glimpse on and attempt to explore some practical methods to deal with the difficulties in MBPs in our own ways.

This paper is organized as follows. In Section 2, the adaptive wavelet methods and adaptive grid techniques are reviewed, and then several problems induced by movable boundary are discussed in detail. In Section 3, extending boundary and contracting boundary are described and resolved by our methods on one-dimensional MBPs. And two examples are computed and discussed to demonstrate the variational adaptive grids in Section 4. Finally, a short summary is given.

2. Difficulties of Wavelet Methods for MBPs

2.1. Review of Adaptive Wavelet Collocation Methods. The adaptive wavelet methods for partial differential equations are based on wavelet decomposing, wavelet threshold filtering and adaptive grids, spatial differentiation computing, and resolving ordinary differential equations.

Any one-dimensional function can be approximately expressed as

\[ f(x) \approx \sum_{k \in \mathbb{Z}} s_{j_k,k} \phi_{j_k,k}(x) + \sum_{j=j_{\min}}^{j_{\max}} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x), \]

where \( \phi_{j_k,k}(x) \) are scaling functions on the lowest level of resolution, \( \psi_{j,k}(x) \) are wavelet functions on level \( j \), \( J \) is the maximum scale factor, and \( s_{j_k,k} \) and \( d_{j,k} \) are the coefficients of scaling functions and wavelets.

The coefficients are calculated through wavelet decomposition in discrete schemes by Mallat algorithm, while second-generation wavelets make it faster and easier. All the discussions in this paper are based on second-generation wavelets and their ramifications. The second-generation wavelet used herein is a kind of lifted interpolating wavelet [1] defined by the prediction and update operators:

\[
P = \frac{1}{16} [-1, 9, 9, -1],
\]

\[
U = \frac{1}{32} [-1, 9, 9, -1].
\]

It has been demonstrated that the wavelet coefficients are great in rough regions, while small in the smooth. So, equation (1) can be decomposed to two terms whose wavelet coefficients are above and below a threshold parameter \( \varepsilon (\varepsilon > 0) \) [1]:

\[
f(x) = f_\geq (x) + f_\leq (x),
\]

where

\[
f_\geq (x) = \sum_{k \in \mathbb{Z}} s_{j_k,k} \phi_{j_k,k}(x) + \sum_{j=j_{\min}}^{j_{\max}} \sum_{|d_{j,k}| > \varepsilon} d_{j,k} \psi_{j,k}(x),
\]

\[
f_\leq (x) = \sum_{j=j_{\min}}^{j_{\max}} \sum_{|d_{j,k}| < \varepsilon} d_{j,k} \psi_{j,k}(x).
\]

For a regular function, the error is bounded as [1]

\[
\|f(x) - f_\leq (x)\| \leq C_1 \varepsilon \|f(x)\|.
\]

Because any wavelet and scale function maps a collocation, so the irregular grid is generated by deleting the small-coefficient wavelets. There are two skills that should be noted: (1) for evolution equations, some collocations near the rough regions should be remained to capture the evolution of local features of the next time step, although their coefficients are truly smaller than the threshold parameter \( \varepsilon \); (2) wavelet decomposition should be performed on the irregular grid to decrease computational costs which may become equal to iteration. More details can be found in Vasilyev and Bowman’s article [1].

On summary, the adaptive grid generated by wavelet threshold filtering includes four kinds of points that are named notable (red), neighbor (blue), assistant (green), and invalid (hollow) in this paper, as shown in Figure 1. The notable points map to the scaling functions and the wavelets whose coefficients are greater than the threshold parameter \( \varepsilon \). And the other points including predicted, assistant, and invalid are all corresponding to the wavelet functions whose coefficients are smaller. The neighbor points in the neighbor domain of the notable are “deleted” firstly after wavelet filtering, but “added” after checking the valid, which are used to predict the local structures and transit the adaptive grid from the current to the next time step. Both the notable and the neighbor are included in
computation, so called “valid.” The assistant points are used to perform wavelet decomposition. And the invalid points are “deleted” finally. So, it is important to keep in mind that all the grid points corresponding to wavelet collocations are in the nested structure.

Here, we introduce two parameters, compression ratio $\beta$ (6) and compression efficiency $\eta$ (7), as tokens of the irregular grid that will be used in the following discussions:

$$\beta = \frac{N_{\text{valid}}}{N_{\text{total}}} \times 100\%,$$

(6)

$$\eta = 1 - \beta,$$

(7)

where $N_{\text{total}}$ denotes the regular grid point number and $N_{\text{valid}}$ is the number of the point used in computation.

2.2. Huge Costs and Ill-Conditioned Grid. It is nature to generate a new grid by the new boundaries at first, as shown in Figure 2(a). But, it will be very costly. The entire new regular grid points (red squares) have new position information that must be updated every time step. And each new point should be evaluated by the old grid points, whose costs are comparative to the iterating process. What is worse, the recursion program should be performed on the new regular grid including all the collocations. In consequence, the huge costs make the advantage of efficient computation lost. So, this method should be refused considering the computational costs.

How about fixing the points on their original positions? As we know, the grid generated by wavelet threshold filtering is nested, which means that each collocation has mother-collocations till the top level. But when the collocations on the boundary added or deleted destroy the nest, some daughter-collocations will lose mother-collocations, which results in that the recursion program cannot run to the top level, as shown in Figure 2(b). For case 1, the right boundary contracts in between points 7 and 8. The right boundary point is not corresponding to any wavelet collocations, so it should be managed separately. Consequently, point 7 becomes the boundary of the nest grid, and there are 8 points left on the domain. It is easy to validate that wavelet decomposition cannot be performed on points 6 and 4 by order. For case 2, the left boundary extends out and generates a regular integer grid cell and a fraction. The new grid possesses 10 points including the point of index -1. If wavelet decomposition is performed from left to right, it is interrupted at the beginning by the ill point of index 1 that was “deleted” before. In a word, the number of new grid points does not remain with $N_0 \cdot 2^n + 1 (N_0, n \in \mathbb{Z}^+)$ anymore.

The discussions above do not attempt to prove it is impossible to perform adaptive wavelet methods for MBPs. Both that are not problems if computational cost is ignored. However, it should be borne in mind that a main purpose of developing adaptive wavelet methods is to save computational costs.

3. Adaptive Wavelet Methods for Boundary Moving

As discussed in Section 2.2, great costs should be avoided for MBPs. So, it is adverse to generate quite a new nested grid on the whole domain or perform adaptive wavelet program on it. The inner points independent of the movable boundary are remained on their original positions. Then, the problem turns to how to keep the nest property of the grid while the boundary moves.

3.1. Boundary Extending. When the boundary extends out of the original boundary, the original grid does not contain the points produced by the extending boundary. If the points beyond the old boundary are brought into the old grid, the nest has a break between the old and new. So, the old boundary should be fixed to keep the nest property of the original grid, as shown in Figure 3. The points out of the old boundary can construct a new nested grid intermittently, on which the adaptive wavelet program is performed.

But it does not need to perform adaptive wavelet program once a nested grid is constructed. Because, the new nested grid out of the old boundary grows from very small to large, and it is not compression-effective for a small size of points. However, if the nested grid grows great, it needs iterating for all the points till using the adaptive method. So, the problem turns to discuss when and by what size to perform the adaptive wavelet program on the new nested grid.

Now a ratio of the number of the cells out of the original boundary to the original regular grid is introduced:

$$\alpha = \frac{N - \bar{N}}{N},$$

(8)

where $N$ is the number of the cells of the original regular grid and is unchanged in the whole process and $\bar{N}$ is the number of the cells beyond the boundary. Generally, we set

$$\alpha = 2^{-n}, \quad n \in \mathbb{Z}^+,$$

(9)

so that both the new and the original grids can use the same adaptive wavelet program without any collisions. And the adaptive wavelet program is performed for the newly constructed grid intermittently, so it is named intermittent adaptive method in this paper.

As the boundary extends continuously, the new grid boundary should be fixed and the same process for the
points recycled beyond the new boundary, which is noted as "turn to step 1" in Figure 3. In this way, the new (blocked in rectangle) turn sold and a new one appears. Here, we have two methods to deal with the new and the old: (1) let the new grids free and perform wavelet decomposition separately on them as the boundary extends continuously (Figure 4(a)), which is named the grid separate method, and (2) the grids of the same size and scale factor (expressed by the same color) combine together to be a greater size grid with the scale factor added one: when the front new grid grows equal to the back, combine them and loop the process on the whole domain, then perform adaptive wavelet program on it (Figure 4(b)), which is defined as the grid combine method. Furthermore, the points of the separated grids are redundant for the combined to perform adaptive wavelet program, so there is no need to do any interpolating more.

3.2. Boundary Contracting. All the points except the part out of the movable boundary have to be kept in count because it needs great computational cost, if a new nested grid is constructed, as mentioned in Section 2. Boundary contracting destroys the nest property of the original grid as the same as boundary extending.

In order to use the same adaptive wavelet program, the part that has been cut off from the boundary should be mapped to the original grid. So, a fictitious boundary and some necessary points are added to the real physical boundary, as shown in Figure 5. The inner grid points are corresponding to the original points. But it is not enough for perform wavelet decomposition because some points needed for wavelet decomposition may stay in the domain that has been cut off. These points (the green on the right side in Figure 5) should be evaluated by the other points that belong to the real physical boundary and valid inner points that have real values. Generally, the nearest one or two points are enough to evaluate the assistant points in the cut-off domain because generating adaptive grid by wavelet threshold filtering is less strict than numerical computation. And "the nearest one" means that the assistant points in the cut-off domain can be evaluated just by the real physical boundary point, which is the simplest approach.

4. Examples and Discussions

4.1. Problem Statement

4.1.1. Local Feature Moving. In order to demonstrate the evolution process of the methods in a clear way, we take a simple example that does not need any integral process but is evaluated by a test function:

\[ f(x) = -\tanh\left(50\left(\frac{x-a}{L} - \frac{1}{3}\right)\right) + \exp\left(-64\left(\frac{x-a}{L} - \frac{2}{3}\right)^2\right), \]

(10)

where \(a\) decides the left boundary, \(L\) defines the domain \([a, a+L]\), and 1/3 and 2/3 position two different local features, as shown in Figure 6. The black curve is the basis, with \(a = 0\) and \(L = 1\), while the red is extended from the black, and the blue is moved from the red.

The initial condition is given by the basis function \((x \in [0,1])\), and the features are changed by the moving boundaries. The left boundary moves with velocity \(v_L\), and the right boundary moves with \(v_R\). So, the domain is decided by \(v_L\), \(v_R\), and time, and the field is evaluated by the moving function, which is very simple and can be used to check the algorithm for MBPs perspicuously. For this test, we set \(v_L = 0.2\) and \(v_R = 0.1\), which means that both the boundaries move from left to right and the domain is compressed.
4.1.2. Gas Gun. In this part, a gas gun model is introduced to demonstrate the applications of the proposed methods for MBPs. The gas gun shown in Figure 7 has a long bore with a hard, small projectile that weighs 40 kg/m² and whose thickness is unconsidered. The projectile is driven by the high-pressure gas initially held in the space between 0 and 0.01 m. In the driving process, the pneumatic resistance by the air is considered.

Both the high-pressure and low-pressure gases are governed by one-dimensional Euler equations:

\[
\begin{align*}
\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} &= 0, \\
U &= \begin{bmatrix} \rho & \rho u & E \end{bmatrix}^T, \\
F &= \begin{bmatrix} \rho u & \rho u^2 + p & (E + p)u \end{bmatrix}^T,
\end{align*}
\]

and the ideal-gas state equation:

\[
p = (\gamma - 1) \left( E - \frac{1}{2} \rho u^2 \right),
\]

where \( \rho \) is density, \( u \) is velocity, \( p \) is static pressure, and \( E \) is total energy of unit volume. \( U \) is the conservation variable vector and \( F \) is the flux vector of the conservation variables. Heat ratio \( \gamma = 1.4 \). The state parameters and their initial values of the gases are listed in Table 1.

The projectile is actuated by the force on the base and baffled on the head:

\[
m_p \frac{d v_p}{dt} = F_{\text{base}} - F_{\text{head}},
\]

where \( m_p \) and \( v_p \) are the mass and velocity of the projectile and \( F_{\text{base}} \) and \( F_{\text{head}} \) are the forces on the base and the head. The boundary on the left of the bore and two faces of the projectile are set as wall, and the vent of the bore is set to free flow. Particularly, the moving boundaries of the gas and the projectile have the same velocity:

\[
v_g = v_p.
\]

The Euler equations (11) are resolved by 1st order upwind scheme [17] based on the Steger–Warming flux vector split method on adaptive grids. The ordinary differential equation (13) is resolved by the explicit Euler method.
4.2. Numerical Results

4.2.1. Problem I. This problem is solved to demonstrate how the method works. According to the given conditions of the moving boundaries, the left boundary contracts in and the right one extends out. Set $J = 7$, so the level index counts from 1 up to 7. More information is presented in Table 2.

For $v_1 = 0.2$ and $v_2 = 0.1$, the function transforms, as shown in Figure 8. The left column shows the functions at different time, while the right does the grid structures. It can be found that the two local features transform while the boundaries move step by step. The left boundary contracts in, so we use the original grid constructed on $[0,1]$ and perform wavelet decomposition on the irregular grid. The right boundary extends out, but the added points cannot be included in the original grid on $[0,1]$. For $t < 2.5$, the added points are less that $N/4$, so it does not perform wavelet decomposition or generate adaptive grid. All the added points are collocated on Level 0 from $t = 1$ to $t = 2.5$. While $t = 2.5$, $N = 64$ ($\alpha = 1/4$), so it is triggered to perform an adaptive wavelet program for the new nested grid. There are two nested grids: the old is on $[0,1]$ and the new is on $[1,1.25]$. For $[1,1.25]$, the finest level is of index 5, and there are 64 cells in the new regular grid and 15 points remained for the new adaptive grid.

If the boundary extends more, the added points cannot be included in the grid on $[1,1.25]$. And it can be anticipated that when the right boundary extends above 1.5, another new nest grid is constructed on $[1.25,1.5]$. Of course, we can perform the adaptive wavelet program for $[1.25,1.5]$ as same as $[1,1.25]$. This approach is natural and simple in logic, and it is fully adaptive to many real problems. However, if the right boundary extends continuously, we must loop them on $[1.5,1.75]$, $[1.75,2]$, etc. Consequently, it is inefficient to perform computer programs and compression efficient turns lower. When $t = 5$, the function transforms in domain $[1,1.5]$. There are two regular nest grids ($J = 5$) on domain $[1,1.25]$ and $[1.25,1.5]$, and the adaptive grids are shown in Figure 9(a). If we combine the two regular grids into one nest grid on $[1,1.5]$, the combined regular grid has 128 pieces, and $J = 6$. But the space step is still 1/256. Then, perform wavelet decomposition and generate adaptive grid that is shown in Figure 9(b). The separated grids have 69 points in total that are 4 points more than the combined adaptive grid. If the extending domain turns very large, there would be more and more “4 points,” which would become tremendous. This talk will be further discussed and demonstrated at the end of Section 4.2.2.

4.2.2. Problem II. The gas gun is computed on four grids of four different scales that are listed in Table 3, including some other necessary information as well. It should be explained particularly that the threshold parameter is set to be $1 \times 10^{-5}$ for the high-pressure area and $0.01 \times 10^{-5}$ for the low-pressure area, for the initial pressure and density are 0.01 times of the high-pressure area. The computation terminates when the projectile is driven to the vent. Figure 10 gives the histories of velocity and displacement of the projectile. It can be found that there is little difference from the results of different scale factors. By the way, the projectile can be driven to ~82.5 m/s within 8 centimeter stroke and 1.26 milliseconds.

The pressure and velocity fields of the scale factor $J = 7$ are selected to be shown in Figure 11. The right boundary of the high-pressure domain extends out with the finest grid area near the movable boundary, for we do not perform the adaptive wavelet program in this domain. The grid combine method is used to make the grid compression very efficient, so as to the computation. The computational domain varies from small to very large, and we set $\alpha = 1/2$ to deal with the new grid points, which is a little different from $\alpha = 1/4$ in Problem I. Consequently, the valid point numbers vary between 20 and 200, and when the displacement increases the numbers climb up gradually and jump down abruptly intermittently every half centimeter, which loops in this way, as shown in Figure 12(a). On the whole, it is really scanty, except for the period from 0.15 ms to 0.3 ms, as shown in Figure 12(b). For the low-pressure domain, the left boundary contracts and the valid point numbers slightly increase at the beginning for the gradient generated beside the boundary then continuously decrease for the computation domain retracts. When the left boundary moves near the right boundary, the domain is compressed nearly to zero and the compression ratio increases faster and faster to 100% that means only a few cells are remaining in the domain and all are valid. Finally the ratio dives to zero, for there are no points any more.

At last, the combine method is discussed comparing to the separate method to show the difference of adaptive grids by wavelet threshold filtering, as shown in Figure 13. As mentioned before, the combine method makes the valid point numbers varying between about 20 and 200, which is similar to a periodic function of uniform amplitude. While for the separate method, the amplitude increases gradually as the domain extends. And the difference turns more and more remarkable when the domain extends to be very large. So, it is suggested to use the combine method in the situation that the computational domain varies from very small to very large.

5. Conclusions

The adaptive wavelet methods are based on nested grids, so as to MBPs. However, it is very costly if a new nested grid is constructed by the new boundaries for every time step. If the inner points are fixed and any points beside the boundary
are added or deleted, the nest property of the grid is destroyed, so the recursive adaptive wavelet program does not adapt to any grid. This paper developed a series of methods or skills to deal with the two difficulties of adaptive wavelet methods for movable boundary problems.

The methods remain the original points in the computational domain and keep the original grid in the nested structure, in order to perform the same recursive adaptive wavelet program till the top level. When boundary extends, it begins to use the adaptive wavelet collocation methods on
Table 3: Parameters used in computation and their values.

| J | Cells (for 1 cm) | Δt (nondimensional) | ε (×10⁻³) | α   |
|---|-----------------|---------------------|----------|-----|
| 4 | 2 × 2⁴          | 5 × 10⁻³            |          | 1/2 |
| 5 | 2 × 2⁵          | 2 × 10⁻³            |          | 1/2 |
| 6 | 2 × 2⁶          | 1 × 10⁻³            | 1 & 0.01 | 1/2 |
| 7 | 2 × 2⁷          | 5 × 10⁻⁴            |          | 1/2 |

Figure 10: Histories of velocity and displacement of the projectile.

Figure 11: Distributions of pressure, velocity, and points of scale factor 7 at different time. (a) Boundary extending. (b) Boundary contracting.
the new grid beyond the original boundary once it reaches a nested structure. So, the adaptive wavelet program is intermittently performed on the new nested grids. If the boundary extends tremendously, the number of the new nested grids grows bigger. It is better to combine them to a greater nested grid for further efficiency. While for boundary contracting, the original boundary remains as a fictitious boundary, and some collocations between the new and original boundaries must be evaluated and used in the wavelet transform, so the adaptive wavelet program can be successfully performed on the original nested grid. Finally, two numerical tests, local features moving and gas gun, were resolved and discussed to show how the methods work. And gas gun numerical test validates that it is really scanty to combine the new nested grids when the computation domain varies from small to very large.

The methods will be developed for two- and three-dimensional movable problems in the near future. It can be predicted that, for boundary extending, the adaptive grid grows like laying bricks, and for boundary contracting, the adaptive grid changes within the framework of the original boundaries.

Data Availability

The data and some codes used to support the findings in the article are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] O. V. Vasilyev and C. Bowman, "Second-generation wavelet collocation method for the solution of partial differential equations," Journal of Computational Physics, vol. 165, no. 2, pp. 660–693, 2000.
[2] M. Farge, “Wavelet transforms and their applications to turbulence,” *Annual Review of Fluid Mechanics*, vol. 24, no. 1, pp. 395–458, 1992.

[3] M. Farge, N. Kevlahan, V. Perrier, and E. Goirand, “Wavelets and turbulence,” *Proceedings of the IEEE*, vol. 84, no. 4, pp. 639–669, 1996.

[4] W. Sweldens, “The lifting scheme: a construction of second generation wavelets,” *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[5] A. Harten, “Adaptive multiresolution schemes for shock computations,” *Journal of Computational Physics*, vol. 115, no. 2, pp. 319–338, 1994.

[6] J. D. Regele and O. V. Vasilyev, “An adaptive wavelet collocation method for shock computations,” *International Journal of Computational Fluid Dynamics*, vol. 23, no. 7, pp. 503–518, 2009.

[7] J. D. Regele, D. R. Kassoy, and O. V. Vasilyev, “Numerical modeling of acoustic timescale detonation initiation,” in *Proceedings of the 46th Aerospace Sciences Meeting and Exhibition AIAA 2008-1037*, Reno, NV, USA, January 2008.

[8] J. D. Regele, D. R. Kassoy, A. Vezolainen, and O. V. Vasilyev, “Purely gasdynamic multidimensional indirect detonation initiation using localized acoustic timescale power deposition,” in *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition AIAA 2013-1172*, Dallas, TX, USA, January 2013.

[9] J. D. Regele, D. R. Kassoy, M. Aslani, and O. V. Vasilyev, “Evolution of detonation formation initiated by a spatially distributed transient energy source,” *Journal of Fluid Mechanics*, vol. 802, pp. 305–332, 2016.

[10] E. Brown-Dymkoski, N. Kasimov, and O. V. Vasilyev, “A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows,” *Journal of Computational Physics*, vol. 262, pp. 344–357, 2014.

[11] S. J. Reckinger, D. Livescu, and O. V. Vasilyev, “Comprehensive numerical methodology for direct numerical simulations of compressible Rayleigh-Taylor instability,” *Journal of Computational Physics*, vol. 313, pp. 181–208, 2016.

[12] G. De Stefano, A. Nejadmalayeri, and O. V. Vasilyev, “Wall-resolved wavelet-based adaptive large-eddy simulation of bluff-body flows with variable thresholding,” *Journal of Fluid Mechanics*, vol. 788, pp. 303–336, 2016.

[13] E. Brown-Dymkoski and O. V. Vasilyev, “Adaptive-anisotropic wavelet collocation method on general curvilinear coordinate systems,” *Journal of Computational Physics*, vol. 333, pp. 414–426, 2017.

[14] G. De Stefano, O. V. Vasilyev, and E. Brown-Dymkoski, “Wavelet-based adaptive unsteady Reynolds-averaged turbulence modeling of external flows,” *Journal of Fluid Mechanics*, vol. 837, pp. 765–787, 2018.

[15] X. Ge, O. V. Vasilyev, G. De Stefano, and M. Y. Hussaini, “Wavelet-based adaptive unsteady Reynolds-averaged Navier-Stokes computation of wall-bounded internal and external compressible turbulent flows,” in *Proceedings of the 2018 AIAA Aerospace Sciences Meeting*, Kissimmee, FL, USA, January 2018.

[16] X. Ge, O. V. Vasilyev, and M. Y. Hussaini, “Wavelet-based delayed detached eddy simulation method for compressible wall bounded turbulent flow modeling,” in *Proceedings of the 2018 AIAA Aerospace Sciences Meeting*, Kissimmee, FL, USA, January 2018.

[17] C. B. Laney, *Computational Gasdynamics*, Cambridge University Press, New York, NY, USA, 1998.