AIAA JOURNAL
Vol. 46, No. 5, May 2008

Improved Computational Efficiency of Unsteady Flow Problems via the Modified Wavelet Method

Hyung Min Kang,* Dong Ho Lee,† and Kyu Hong Kim‡
Seoul National University, Seoul 151-742, Republic of Korea
and
Do Hyung Lee§
Hanyang University, Ansan 425-791, Republic of Korea

DOI: 10.2514/1.34294

A newly modified adaptive wavelet method is proposed to enhance the computational efficiency and to maintain a level of accuracy in unsteady flow calculations that is similar to the conventional schemes. The slim and compressed data set is well adapted to the local features of a solution, and fluxes are only calculated according to this adaptive data set, which results in shorter total computation time. While retaining this advantage, the following methods are applied to the wavelet method. First, residual interpolation is applied and entire residual values are calculated before time integration, and it is therefore not necessary to add several cells to a data set, which were definitely necessary in previous wavelet methods. Second, the threshold value is modified to consider the temporal accuracy as well as the spatial accuracy of the conventional schemes. This modification ensures that the errors caused by the thresholding method do not damage the numerical accuracy of the conventional schemes. The modified wavelet method is successfully applied to the unsteady flow problems and compared with the results of the conventional solvers. In consequence, computing efficiency is enhanced without the deterioration of the solution accuracy.

Nomenclature

a	=	speed of sound
$d_{l,k}$	=	wavelet coefficient at $x_{l,k}$
$d^n_{i,j}$	=	wavelet coefficient at the (i, j) cell of the n time step
$f_{l,k}$, $f(x_{l,k})$	=	exact functional value at $x_{l,k}$
$\tilde{f}_{l,k}$, $\tilde{f}(x_{l,k})$	=	interpolated value given by the interpolating subdivision scheme
$I(\varepsilon)$	=	adaptive data set by thresholding with the ε value
$I(\varepsilon')$	=	adaptive data set by modified thresholding with the ε' value
$L^n_{i,j}$	=	(ρ, p, u, v) at the (i, j) cell of the n time step
$\tilde{L}^n_{i,j}$	=	interpolated value of $L^n_{i,j}$ given by the interpolating subdivision scheme
p	=	pressure
$R^n_{i,j}$	=	residual values at the (i, j) cell of n time steps
$\tilde{R}^n_{i,j}$	=	interpolated value of $R^n_{i,j}$ given by the interpolating subdivision scheme
u, v	=	components of the velocity vector
V_l	=	dyadic grid set
$x_{l,k}$	=	kth cell on V_l
γ	=	ratio of specific heat (1.4 for air)
ε	=	threshold value
ε'	=	modified threshold value
ρ	=	density

Subscripts

i, j	=	position of a cell
K	=	position of the cell
L	=	wavelet resolution level

Superscript

n	=	current time step

Received 28 August 2007; revision received 9 January 2008; accepted for publication 14 January 2008.

*Ph.D. Candidate, School of Mechanical and Aerospace Engineering, Building 301-1213, San 56-1, Shinlim-dong, Gwanak-gu; kangm@snu.ac.kr.

†Professor, School of Mechanical and Aerospace Engineering, Institute of Advanced Aerospace Technology, Building 301-1302, San 56-1, Shinlim-dong, Gwanak-gu; donghlee@snu.ac.kr. Member AIAA.

‡Assistant Professor, School of Mechanical and Aerospace Engineering, Institute of Advanced Aerospace Technology, Building 302-628, San 56-1, Shinlim-dong, Gwanak-gu; aerocfd1@snu.ac.kr. Member AIAA (Corresponding Author).

§Professor, Department of Mechanical Engineering, 1271 Sa 1, KyungKi-Do, Dong; dohyung@hanyang.ac.kr. Member AIAA.

I. Introduction

RECENTLY, there have been numerous improvements in the computational accuracy and efficiency by the developments of the numerical schemes and the analysis systems in computational fluid dynamics (CFD). However, to calculate the flow problems more accurately, a large number of grid points are still needed in the spatial domain. Moreover, the size of the time step reduces according to the increment of the grid points (i.e., the decrement of the size of the cells). In spite of the necessity of a dense-grid system, it is a waste of the computational resources to use the fine grids in the whole domain, because the majority of the CFD data sets are in the smooth region in which the accurate solutions can be obtained with the relatively coarse grids. The dense grids are needed only in the rapidly changing regions such as shock waves, boundary layers, etc.

For that reason, several types of adaptive methods have been implemented to improve the overall efficiency of CFD. The representative examples are the adaptive mesh refinement (AMR) method [1] and the adaptive wavelet method. The main focus of the AMR method is to add some mesh points to physically important region such as shocks and to improve the solution's accuracy without the loss of efficiency. On the other hand, the main feature of the adaptive wavelet method is to omit some mesh points from the given data set with the threshold value and improve the computational efficiency while retaining the solution accuracy. Currently, many numerical techniques based on the wavelet method have emerged for resolving the problems originated by the dense-grid system.

The adaptive wavelet method has been applied to CFD in various ways. Harten [2] presented an adaptive multiresolution scheme for computing the discontinuous solutions of hyperbolic PDEs. Holmström [3] proposed the algorithm that uses the interpolating wavelet transformation to organize an adaptive data set. Sjögreen [4] also used a multiresolution scheme based on the interpolating wavelet transformation to solve the compressible Euler equations. Vasilyev and Paolucci [5] extended the adaptive wavelet schemes to multidimensional partial differential equations.

The wavelet method is based on the following concept. In wavelet decomposition and thresholding about a flow solution, some cells remain in the adaptive data set if the wavelet coefficients at these cells are larger than ε. These procedures allow the remaining cells to follow the local flow feature and enable the data set to be adapted to this local feature. Complicated flux evaluations and time integration are performed only at the cells included in the adaptive data set, and so the total computation time is reduced. After time integration, the flow variables are simply interpolated at the cells excluded from the adaptive data set.

Generally, the thresholding process results in the insertion of an ε order of error to the solution. This $\mathcal{O}(\varepsilon)$ error is transferred to the adjacent computational domain during the time-integration process and therefore obstructs the stable construction of an adaptive data set at the next time step ($n + 1$ time step). Also, if the flow features change largely at the $n + 1$ time step, the important cells (i.e., the cells included in the adaptive data set) should change so as to follow the change of the local flow features. However, due to the thresholding at the current time step (n time step), the adaptive data set includes only the information of the n time step. The solution's accuracy can be deteriorated because the flow variables are interpolated at the $n + 1$ time step based on the adaptive data set at the n time step. To solve this problem, the previous wavelet methods added several adjacent cells to the adaptive data set at the n time step considering the change of the flow features.

However, it is difficult to determine the adequate number of additional cells because each problem has its own flow features. No rigorous mathematical approach has been presented to make this determination, but rather it has been determined arbitrarily by the user's experience. Moreover, the increase of the number of the cells in an adaptive data set lowers the compression ratio of the wavelet methods, and it decreases the overall computational efficiency. Also, if the additional $\mathcal{O}(\varepsilon)$ errors are larger than the truncation errors of the numerical schemes, the errors can damage the spatial or the temporal accuracy of the conventional scheme. Especially, in the case of the flow problem with nonuniform grids or local time method, the order of the truncation error may vary according to the size of the cells. Then to retain the numerical accuracy of the conventional schemes, the threshold value in the previous wavelet method should be selected to have the value lower than the minimum truncation error at the minimum-size cell, and such a selection will reduce the overall computational efficiency significantly.

The objective of this study is the improvement of the computational efficiency of the unsteady flow problems while the numerical accuracy of a solution is automatically retained without the user's arbitrary adjustment of the threshold value. For achieving these purposes, the general adaptive wavelet procedure is changed by adopting residual interpolation at the n time step, not at the $n + 1$ time step, a concept that was introduced by Chiavassa and Donat [6] for the point-value setting. In this paper, we modify this method for the cell-value setting and perform time integration on the entire domain. Because the interpolation is performed at the n time step, there is no need to add several adjacent cells to the adaptive data set.

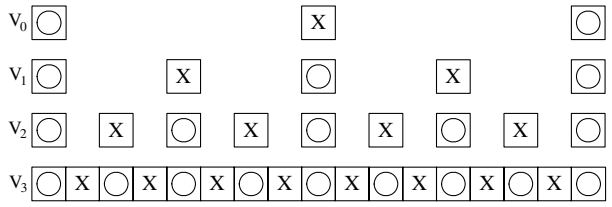


Fig. 1 Dyadic structure of data set.

Also, the threshold value is modified to maintain the spatial and temporal accuracies of the conventional schemes. In the previous research, we modified the threshold value to maintain the spatial accuracy of a conventional CFD solver in the steady-state flow problems [7]. Here, the method is extended to cover the unsteady problems by considering the temporal accuracy as well as the spatial accuracy of the solver. Throughout these processes, the accuracy of the conventional solver is conserved and the computation time is substantially reduced. Several numerical examples are presented to verify the enhancement of the efficiency and the maintenance of the numerical accuracy of the developed method.

This paper is organized as follows. After the Introduction, we briefly explain the interpolating wavelet decomposition and thresholding process as the preliminary backgrounds. Then we describe the overall implementation of the adaptive wavelet method in a conventional unsteady CFD solver, including the modified threshold method and residual interpolation. To demonstrate the efficiency and the accuracy of the developed method, several unsteady simulations are performed. Finally, the conclusions and future works are presented.

II. Interpolating Wavelet Transformation

In this research, we used the interpolating wavelet transformation presented by Donoho [8]. This wavelet transformation extends the interpolating subdivision scheme presented by Deslauriers and Dubuc [9]. A brief summary of the wavelet transformation processes is as follows: First, assume that the dyadic grid set is presented as Eq. (1).

$$V_l = \{x_{l,k} \in R\colon x_{l,k} = 2^{-l}k, k \in Z\}, \qquad l \in Z \tag{1}$$

Then the dyadic grid V_{l+1} contains all the cells in V_l and the additional cells located in the middle of the adjacent two cells in V_l, as shown in Fig. 1.

In the decomposition step, the values at the even-numbered cells on V_{l+1} ($\bigcirc$ positions) are saved at the cells on V_l. The values at the odd-numbered cells on V_{l+1} (X positions) are estimated by the interpolation polynomials P_{l+1} based on the values on V_l. Then the interpolating wavelet transformation procedure is expressed as in Eq. (2).

$$\begin{cases} \tilde{f}_{l+1,2k} = f_{l,k} & \text{at even-numbered cells} \\ \tilde{f}_{l+1,2k+1} = P_{l+1,2k+1}(f_{l,k-p/2+1}, \ldots, f_{l,k}, f_{l,k+1}, \ldots, f_{l,k+p/2}) & \text{at odd-numbered cells} \end{cases}, \quad \forall\ k \in Z \tag{2}$$

where

$$P_{l+1,2k+1}(f_{l,k-p/2+1}, \ldots, f_{l,k}, f_{l,k+1}, \ldots, f_{l,k+p/2}) = \sum_{m=-p/2+1}^{p/2} h_m f_{l,k+m}$$

and p is an even number.

The coefficients h_m are summarized in Table 1 about the fourth order and sixth order of the interpolating polynomials. Then $d_{l,k}$ can be defined as Eq. (3).

Table 1 Interpolation coefficients according to each order

m	-2	-1	0	1	2	3
Fourth order ($p = 4$)		$-1/16$	9/16	9/16	$-1/16$	
Sixth order ($p = 6$)	3/256	$-25/256$	75/128	75/128	$-25/256$	3/256

$$d_{l,k} = f_{l+1,2k+1} - \tilde{f}_{l+1,2k+1} \quad \forall\ k \in Z \tag{3}$$

By Eqs. (2) and (3), one level of the wavelet decomposition is performed and it can be iterated with a coarser resolution level of data set V_l until the coarsest data set V_0 is obtained.

After calculating all $d_{l,k}$ values according to l and k, the thresholding process is applied. If $d_{l,k}$ is smaller than the threshold value ε, the real function values at the (l, k) cell can be ignored in the computation domain; if not, the cell is preserved and, consequently, the $I(\varepsilon)$ data set is constructed as Eq. (4).

$$I(\varepsilon) = \{(l,k):\ |d_{l,k}| > \varepsilon,\ l_{\text{coarsest level}} \le l < L_{\text{finest level}}\} \tag{4}$$

This $I(\varepsilon)$ data set has the following features: In the smooth region, $f_{l,k}$ can be accurately approximated by the interpolating polynomials; that is, $d_{l,k}$ is small enough. The remaining number of cells in the $I(\varepsilon)$ data set is small in this region. Near the region that the function values change rapidly, $f_{l,k}$ are not accurately estimated by interpolation; that is, $d_{l,k}$ is larger than ε and many cells are included in the $I(\varepsilon)$ data set. Consequently, the $I(\varepsilon)$ data set follows the local features of the function and becomes an adaptive data set to the solution. Also, the $I(\varepsilon)$ data set contains the interpolation errors at the excluded cells in the data set. Donoho [8] proved that the maximum $|d_{l,k}|_\infty$ is bounded within the order of ε, as shown in Eq. (5).

$$|d_{l,k}|_\infty = |f_{l,k} - \tilde{f}_{l,k}|_\infty < \mathcal{O}(\varepsilon) \tag{5}$$

III. Implementation of the Modified Wavelet Method

In this research, the two-dimensional Euler equations are used as the governing equations of the unsteady flow problems. The two-dimensional Euler equations are written as Eq. (6).

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \quad \text{with} \quad Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{bmatrix}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e_t + p)u \end{bmatrix}, \qquad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e_t + p)v \end{bmatrix} \tag{6}$$

$$e_t = \frac{a^2}{\gamma(\gamma - 1)} + \frac{1}{2}(u^2 + v^2)$$

where all the properties and governing equations are non-dimensionalized.

By the generalized coordinate transformation, Eq. (6) is rewritten as Eq. (7).

$$\frac{\partial \bar{Q}}{\partial \tau} = -\left[\frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta}\right] = -\bar{R}^n_{i,j} \quad \text{with} \quad \bar{Q} = \frac{Q}{J}$$

$$\bar{E} = \frac{1}{J}[\xi_t + \xi_x E, +\xi_y F], \qquad \bar{F} = \frac{1}{J}[\eta_t + \eta_x E, +\eta_y F] \tag{7}$$

Here, we aimed to improve the computational efficiency of the unsteady flow problems, while maintaining the numerical accuracy of the conventional solvers without adjustment of the threshold value by the user. For these purposes, the general adaptive wavelet procedure is changed by adopting residual interpolation at the n time step and time integration is performed on the entire domain. The procedure of selecting and adding additional cells to an adaptive data set according to the wavelet resolution levels and positions is excluded. And the modified threshold method is applied to maintain the spatial and the temporal accuracies of the conventional schemes. Also, a stabilization method is applied to construct an adaptive data set stably while the compression ratio of the wavelet method is enhanced. The flowchart in Fig. 2 shows the overall implementation of the modified adaptive wavelet method in the conventional solver. This implementation is composed of the following sequential steps.

A. Wavelet Decomposition

Assume an initial two-dimensional dyadic grid set, as shown in Fig. 3. In this figure, ○ cells are even-numbered cells and the others

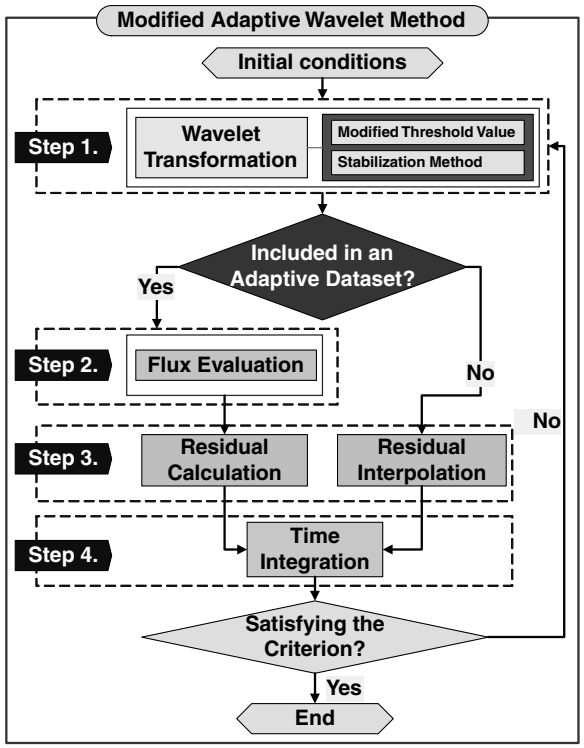


Fig. 2 Overall procedure of flow simulation with the modified wavelet method.

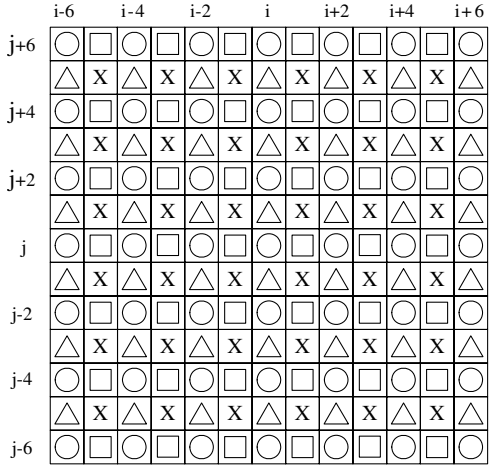


Fig. 3 Example of a two-dimensional dyadic grid set.

are odd-numbered cells. Then one-dimensional interpolating wavelet transformation defined by Eq. (2) can be rewritten for the two-dimensional formulations. In this research, we concentrated on the unsteady flow problems. In the unsteady problems, a tiny change of the flow may grow into a large change with the passage of time. Thus, even a tiny variation has to be considered. In this research, we used the sixth order of interpolating polynomial and present the two-dimensional formulations in Eqs. (8).

For □:

$$\tilde{L}^n_{i+1,j} = \frac{3}{256}L^n_{i-4,j} - \frac{25}{256}L^n_{i-2,j} + \frac{75}{128}L^n_{i,j} + \frac{75}{128}L^n_{i+2,j} - \frac{25}{256}L^n_{i+4,j} + \frac{3}{256}L^n_{i+6,j} \tag{8a}$$

For △:

$$\tilde{L}^n_{i,j+1} = \frac{3}{256}L^n_{i,j-4} - \frac{25}{256}L^n_{i,j-2} + \frac{75}{128}L^n_{i,j} + \frac{75}{128}L^n_{i,j+2} - \frac{25}{256}L^n_{i,j+4} + \frac{3}{256}L^n_{i,j+6} \tag{8b}$$

For X:

$$\begin{aligned}\tilde{L}^n_{i+1,j+1} = \frac{1}{512}\times\Big(&3L^n_{i,j-4} + 3L^n_{i+2,j-4} + 2L^n_{i-2,j-2} \\ &- 27L^n_{i,j-2} - 27L^n_{i+2,j-2} + 2L^n_{i+4,j-2} + 3L^n_{i-4,j} - 27L^n_{i-2,j} \\ &+ 174L^n_{i,j} + 174L^n_{i+2,j} - 27L^n_{i+4,j} + 3L^n_{i+6,j} + 3L^n_{i-4,j+2} \\ &- 27L^n_{i-2,j+2} + 174L^n_{i,j+2} + 174L^n_{i+2,j+2} - 27L^n_{i+4,j+2} \\ &+ 3L^n_{i+6,j+2} + 2L^n_{i-2,j+4} - 27L^n_{i,j+4} - 27L^n_{i+2,j+4} \\ &+ 2L^n_{i+4,j+4} + 3L^n_{i,j+6} + 3L^n_{i+2,j+6}\Big)\end{aligned} \tag{8c}$$

Here, the formulation at the X position in Eq. (8) uses 24 cells for the sixth-order interpolating polynomial and is very complicated. Instead of this equation, we formulate Eq. (9) using 12 cells as follows.

$$\begin{aligned}\tilde{L}^n_{i+1,j+1} = 0.5\times&\left(\frac{3}{256}L^n_{i-4,j-4} - \frac{25}{256}L^n_{i-2,j-2} + \frac{75}{128}L^n_{i,j} + \frac{75}{128}L^n_{i+2,j+2} - \frac{25}{256}L^n_{i+4,j+4} + \frac{3}{256}L^n_{i+6,j+6}\right) + 0.5 \\ &\times\left(\frac{3}{256}L^n_{i-4,j+6} - \frac{25}{256}L^n_{i-2,j+4} + \frac{75}{128}L^n_{i,j+2} + \frac{75}{128}L^n_{i+2,j} - \frac{25}{256}L^n_{i+4,j-2} + \frac{3}{256}L^n_{i+6,j-4}\right)\end{aligned} \tag{9}$$

Based on the many numerical simulations, Eq. (9) has much better efficiency than the accurate formulation with little loss of accuracy. Therefore, it is used for the calculation of $\tilde{L}_{i,j}$ at X cells in the present paper.

Note that some indexes of $\tilde{L}_{i,j}$ in Eq. (8) are located outside of the computation domain near its boundaries. To avoid this situation, cells $i = 2^{(l-1)}$ ($l_{\text{coarsest level}} \le l < L_{\text{finest level}}$) are always included in the adaptive data set. At the next odd-numbered cells $i = 2^l + 2^{(l-1)}$, the fourth-order interpolating polynomial is applied. The overall boundary treatment is presented in Fig. 4 and the same procedure can be applied to the other side of the boundary or the boundaries along the j direction.

Then $d^n_{i,j}$ of each □, △, and X position can be derived as in Eqs. (10).

For □:

$$d^n_{i+1,j} = L^n_{i+1,j} - \tilde{L}^n_{i+1,j} \tag{10a}$$

For △:

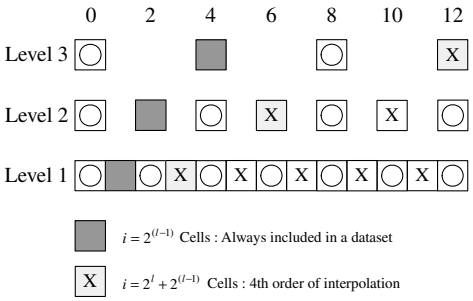


Fig. 4 Boundary treatment in a wavelet decomposition.

$$d^n_{i,j+1} = L^n_{i,j+1} - \tilde{L}^n_{i,j+1} \tag{10b}$$

For X:

$$d^n_{i+1,j+1} = L^n_{i+1,j+1} - \tilde{L}^n_{i+1,j+1} \tag{10c}$$

These routines are reiterated in the next-coarser level of the grid set and, consequently, all wavelet coefficients are computed.

B. Modified Thresholding Method

The modified thresholding method modifies the threshold value itself and applies the stabilization method. At first, the threshold value is reformed to maintain the spatial and the temporal accuracies of the conventional CFD solvers. Then the stabilization method is applied to improve the compression ratio of the wavelet method. The details are presented in the following subsections.

1. *Modified Threshold Value*

The fluxes are discretized with the lth order of spatial accuracy and each primitive variable or flux has the lth order of truncation error in Eq. (11).

$$\begin{aligned}\rho^n_{i,j} &= \rho^n_{i,j,\text{real}} + \mathcal{O}(\Delta x^{l+1}), & u^n_{i,j} &= u^n_{i,j,\text{real}} + \mathcal{O}(\Delta x^{l+1}) \\ v^n_{i,j} &= v^n_{i,j,\text{real}} + \mathcal{O}(\Delta x^{l+1}), & p^n_{i,j} &= p^n_{i,j,\text{real}} + \mathcal{O}(\Delta x^{l+1}) \\ E^n_{1/2} &= E^n_{1/2,\text{real}} + \mathcal{O}(\Delta x^{l+1}), & F^n_{1/2} &= F^n_{1/2,\text{real}} + \mathcal{O}(\Delta x^{l+1})\end{aligned} \tag{11}$$

Also, when we perform thresholding for the wavelet coefficients below the threshold value ε, each primitive variable or flux value has the $\mathcal{O}(\varepsilon)$ error as given by Eq. (12).

$$\begin{aligned}\tilde{\rho}^n_{i,j} &= \rho^n_{i,j} + \mathcal{O}(\varepsilon), & \tilde{u}^n_{i,j} &= u^n_{i,j} + \mathcal{O}(\varepsilon) \\ \tilde{v}^n_{i,j} &= v^n_{i,j} + \mathcal{O}(\varepsilon), & \tilde{p}^n_{i,j} &= p^n_{i,j} + \mathcal{O}(\varepsilon) \\ \tilde{E}^n_{1/2} &= E^n_{1/2} + \mathcal{O}(\varepsilon), & \tilde{F}^n_{1/2} &= F^n_{1/2} + \mathcal{O}(\varepsilon)\end{aligned} \tag{12}$$

From the combination of Eqs. (11) and (12), the fluxes come to have the following order of errors due to spatial discretization and thresholding, as shown in Eq. (13).

$$\begin{aligned}\tilde{E}^n_{1/2} &= E^n_{1/2,\text{real}} + \mathcal{O}(\Delta x^{l+1}) + \mathcal{O}(\varepsilon) \\ \tilde{F}^n_{1/2} &= F^n_{1/2,\text{real}} + \mathcal{O}(\Delta x^{l+1}) + \mathcal{O}(\varepsilon)\end{aligned} \tag{13}$$

By time integration with the mth order of temporal accuracy, all terms of errors are derived as Eq. (14).

$$\tilde{Q}^{n+1}_{i,j} = \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right) + \mathcal{O}(\Delta x^l\cdot\Delta t) + \mathcal{O}(\Delta t^{m+1}) + \mathcal{O}(\varepsilon\cdot\Delta t) \tag{14}$$

Here, the $\mathcal{O}(\varepsilon)$ term is important for determining the overall accuracy. If $\mathcal{O}(\varepsilon)$ is greater than $\mathcal{O}(\Delta x^l)$ or $\mathcal{O}(\Delta t^m)$, the spatial accuracy or the temporal accuracy may be deteriorated. Thus, the

threshold value ε must be determined such that the magnitude of $\mathcal{O}(\varepsilon)$ is less than the value of $\mathcal{O}(\Delta x^l)$ or $\mathcal{O}(\Delta t^m)$. However, there is no obvious standard to compare the temporal error term against the spatial error term because $\mathcal{O}(\varepsilon)$ is essentially generated from interpolation in space. Therefore, the temporal truncation error term is changed in terms of the spatial derivatives to determine the dominance between the two error terms.

More detailed expressions of the spatial and the temporal truncation error terms are written in Eq. (15).

$$\begin{aligned}\mathcal{O}(\Delta x^l \cdot \Delta t) &\approx \left|\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\right|(\Delta x)^l\mathcal{O}(\Delta t)\\ \mathcal{O}(\Delta t^{m+1}) &\approx \left|\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial t^{m+1}}\right|(\Delta t)^m\mathcal{O}(\Delta t)\end{aligned} \tag{15}$$

If we assume a linear advection equation, we obtain the relation between the temporal derivative and the spatial derivative, $\partial f/\partial t = -u(\partial f/\partial x)$. Here, we use the maximum contravariant velocity, U or V, as the advection speed u for application to two dimensions. Then the following Eq. (16a) can be derived.

$$\frac{\partial \tilde{Q}^n_{i,j}}{\partial t} = -\max(|U|,|V|)\frac{\partial \tilde{Q}^n_{i,j}}{\partial x} \tag{16a}$$

Based on Eq. (16a), the second-order temporal derivative term can be rewritten as Eq. (16b).

$$\begin{aligned}\frac{\partial^2 \tilde{Q}^n_{i,j}}{\partial t^2} &= \frac{\partial}{\partial t}\left(\frac{\partial \tilde{Q}^n_{i,j}}{\partial t}\right) = \frac{\partial}{\partial t}\left(-\max(|U|,|V|)\frac{\partial \tilde{Q}^n_{i,j}}{\partial x}\right)\\ &= -\max(|U|,|V|)\frac{\partial}{\partial x}\left(\frac{\partial \tilde{Q}^n_{i,j}}{\partial t}\right) = (-\max(|U|,|V|)^2\frac{\partial^2 \tilde{Q}^n_{i,j}}{\partial x^2}\end{aligned} \tag{16b}$$

Then the $(m+1)$th order temporal derivative can be transformed to the $(m+1)$th order spatial derivative in Eq. (16c).

$$\begin{aligned}\frac{\partial^{m+1} \tilde{Q}^n_{i,j}}{\partial t^{m+1}} &= \frac{\partial}{\partial t}\left(\frac{\partial^m \tilde{Q}^n_{i,j}}{\partial t^m}\right) = \frac{\partial}{\partial t}\left((-\max(|U|,|V|))^m\frac{\partial^m \tilde{Q}^n_{i,j}}{\partial x^m}\right)\\ &= (-\max(|U|,|V|))^m\frac{\partial^m}{\partial x^m}\left(\frac{\partial \tilde{Q}^n_{i,j}}{\partial t}\right)\\ &= (-\max(|U|,|V|))^{m+1}\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial x^{m+1}}\end{aligned} \tag{16c}$$

By introducing Eqs. (15) and (16c) into Eq. (14), each error term is written as shown in Eq. (17).

$$\begin{aligned}\tilde{Q}^{n+1}_{i,j} &= \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right)\\ &+ \left|\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\right|(\Delta x)^l\mathcal{O}(\Delta t)\\ &+ \left|(-\max(|U|,|V|))^{m+1}\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial x^{m+1}}\right|(\Delta t)^m\mathcal{O}(\Delta t) + \mathcal{O}(\varepsilon\cdot\Delta t)\\ &= \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right)\\ &+ \left[\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\mathcal{O}(\Delta x^l)\right.\\ &+ \left.\left|\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial x^{m+1}}\right|\mathcal{O}(\max(|U|,|V|)^{m+1}\cdot\Delta t^m) + \mathcal{O}(\varepsilon)\right]\Delta t\end{aligned} \tag{17}$$

By introducing the Courant–Friedrichs–Lewy (CFL) condition, where

$$\Delta t = \frac{\text{CFL}\cdot\Delta x}{\max(|U|,|V|)}$$

$$\begin{aligned}\tilde{Q}^{n+1}_{i,j} &= \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right)\\ &+ \left[\left|\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\right|\mathcal{O}(\Delta x^l) + \left|\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial x^{m+1}}\right|\mathcal{O}\left(\max(|U|,|V|)^{m+1}\right.\right.\\ &\left.\left.\cdot\left(\frac{\text{CFL}\cdot\Delta x}{\max(|U|,|V|)}\right)^m\right) + \mathcal{O}(\varepsilon)\right]\Delta t = \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}}\right.\\ &\left. - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right) + \left[\left|\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\right|\mathcal{O}(\Delta x^l)\right.\\ &+ \left.\left|\max(|U|,|V|)\frac{\partial^{m+1}\tilde{Q}^n_{i,j}}{\partial x^{m+1}}\right|O(\text{CFL}^m\cdot\Delta x^m) + \mathcal{O}(\varepsilon)\right]\Delta t\\ &= \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right)\\ &+ \left[\left|\frac{\partial^{l+1}\tilde{E}^n_{i,j}}{\partial x^{l+1}}\right|\mathcal{O}(\Delta x^l) + \left|\frac{\partial^{m+1}\tilde{E}^n_{i,j}}{\partial x^{m+1}}\right|\mathcal{O}(\text{CFL}^m\cdot\Delta x^m) + \mathcal{O}(\varepsilon)\right]\Delta t\end{aligned} \tag{18}$$

Consequently, Eq. (19) can be derived from Eq. (18).

$$\begin{aligned}\tilde{Q}^{n+1}_{i,j} &= \tilde{Q}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(E^n_{1/2,\text{real}} - E^n_{-1/2,\text{real}} + F^n_{1/2,\text{real}} - F^n_{-1/2,\text{real}}\right)\\ &+ [\mathcal{O}(\Delta x^l) + \mathcal{O}(\text{CFL}^m\cdot\Delta x^m) + \mathcal{O}(\varepsilon)]\Delta t\end{aligned} \tag{19}$$

In Eq. (19), if $\Delta x^l \geq \text{CFL}^m\cdot\Delta x^m$, the spatial error term is dominant and the modified threshold value is set to $\varepsilon' = \min\{\varepsilon, \Delta x^l\}$. If $\Delta x^l < \text{CFL}^m\cdot\Delta x^m$, then $\varepsilon' = \min\{\varepsilon, \text{CFL}^m\cdot\Delta x^m\}$. Therefore, to maintain the lth order of spatial accuracy and the mth order of temporal accuracy of the numerical schemes simultaneously, the modified threshold value ε' is defined, as given by Eq. (20).

$$\varepsilon' = \min[\varepsilon, \max(\Delta x^l, \text{CFL}^m\cdot\Delta x^m)] \tag{20}$$

Based on this modified threshold value ε', we can control the flag values of the grid points and threshold a data set. If $d^n_{i,j}$ is larger than ε', the flag value of the point (i,j) is determined as 1 and the point is included in $I(\varepsilon')^n$ data set; if not, the flag value is set as 0 and the grid point is excluded from the $I(\varepsilon')^n$ data set. Then the data set is adapted to the flow features while maintaining the numerical accuracy of the conventional schemes.

2. *Stabilization of Constructing an Adaptive Data Set*

As previously mentioned, the thresholding of a data set inevitably leads to the additional $\mathcal{O}(\varepsilon')$ errors that are imposed on the solution. These errors are transferred to the surroundings. The wavelet coefficients at the surrounding region also have the $\mathcal{O}(\varepsilon')$ errors, which make the $I(\varepsilon')^n$ data set change independently of the flow features. For resolving this problem, we developed a stabilization method and applied it to the steady-state problems in the previous research [7].

Algorithm 1. The stabilization method for constructing an adaptive data set is as follows:

$I\text{tag}_{i,j}$: Included positions in an adaptive data set at n time step

$I\text{tag}1_{i,j}$: Included positions in an adaptive data set at $n-1$ time step

Ctemp: $|d_{i,j}|$

do $j := 1$ to $j$max

do $i := 1$ to $i$max

If $C\text{temp} > \varepsilon'$, then

if $I\text{tag}1_{i,j} \neq 1$, then

if $C\text{temp} > \beta\varepsilon'$, then $I\text{tag}_{i,j} = 1$

else $I\text{tag}_{i,j} = 0$

```
      endif
    else Itag_{i,j} = 1
    endif
  else
    if Itag1_{i,j} = 1, then
      if Ctemp < αε', then Itag_{i,j} = 0
      else Itag_{i,j} = 1
      endif
    else Itag_{i,j} = 0
    endif
  endif
  enddo
  enddo
```

We applied this method again to the unsteady flow problems to construct an adaptive data set stably and retain the compression ratio of the wavelet method.

C. Flux Evaluation

After constructing the $I(\varepsilon')^n$ data set, the flag values of the data set determine whether the flux values are to be evaluated or not. If the flag value is 1, the flux value is calculated by the conventional schemes such as AUSMPW+ or Roe's flux-difference splitting method. In this research, we use the finite volume method (FVM); that is, the numerical fluxes $\bar{E}$ and $\bar{F}$ are defined at these interfaces. Thus, if the flag value of the cell (i, j) is 1, flux evaluations are performed on all the sides of the cell and residual values at the cell (i, j) are calculated. If the flag value is 0, flux evaluation is not performed, and so the total computation time is decreased.

D. Residual Interpolation and Time Integration

Generally, the flow properties are interpolated after time integration in the most previous wavelet methods. That is, the flow properties are reconstructed at the $n + 1$ time step based on $I(\varepsilon')^n$. This discrepancy of the time step causes the following numerical problem: If the flow changes rapidly (e.g., shock or vortex propagation, etc.), the neighboring cells of the cell that is included in $I(\varepsilon')^n$ become important as well as the cell in $I(\varepsilon')^n$ itself. If they are not included in $I(\varepsilon')^n$, the interpolation at these neighboring cells cannot guarantee the solution accuracy and may deteriorate it at the $n + 1$ time step.

Concerning these problems, the previous wavelet methods add some adjacent cells to $I(\varepsilon')^n$ so that the data set includes the information of both the n and the $n + 1$ time steps. However, the adequate number of additional cells is selected arbitrarily depending on the user's experience, because each flow has its own flow features. Thus, the user should adjust the criteria of the thresholding so that the number of additional cells can be suitable for the each problem. Moreover, the increase of the number of the cells in $I(\varepsilon')^n$ results in more calculation of flux values, which requires additional computational cost. To solve this problem, residual interpolation is adopted at the n time step before time integration in the present research. Because interpolation is performed based on the solution at the n time step, there is no need to consider the situation at the $n + 1$ time step and to add some cells to $I(\varepsilon')^n$.

Consider that the spatial discretization method with the lth order spatial accuracy is used. Then the residual values have the $(l + 1)$th order of the spatial error term as given by Eq. (21).

$$\begin{aligned}\tilde{R}^n_{i,j} &= \left(\Delta y_{i+1/2,j}\tilde{E}^n_{i+1/2} - \Delta y_{i-1/2,j}\tilde{E}^n_{i-1/2}\right) \\ &+ \left(\Delta x_{i,j+1/2}\tilde{F}^n_{j+1/2} - \Delta x_{i,j-1/2}\tilde{F}^n_{j-1/2}\right) \\ &= \Delta y\left(E^n_{i+1/2} - E^n_{i-1/2} + \mathcal{O}(\Delta x^l)\right. \\ &\left. + \mathcal{O}(\varepsilon)\right) + \Delta x\left(F^n_{j+1/2} - F^n_{j-1/2} + \mathcal{O}(\Delta y^l) + \mathcal{O}(\varepsilon)\right) \\ &= R^n_{i,j} + \mathcal{O}(\Delta x^l \Delta y) + \mathcal{O}(\Delta y^l \Delta x) \approx R^n_{i,j} + \mathcal{O}(\Delta x^{l+1})\end{aligned} \tag{21}$$

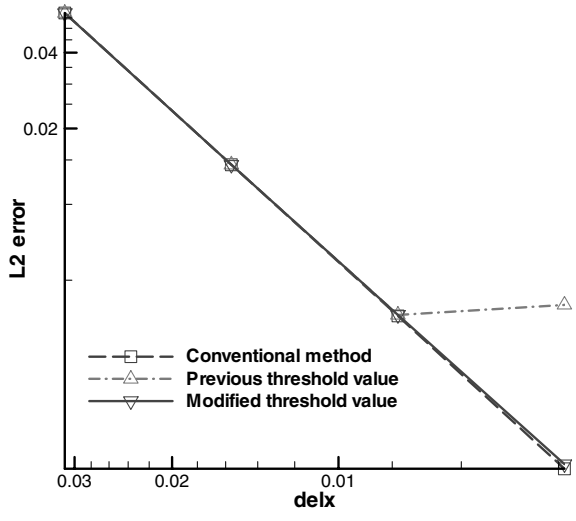


Fig. 5 L2 error according to Δx with $a = 1$.

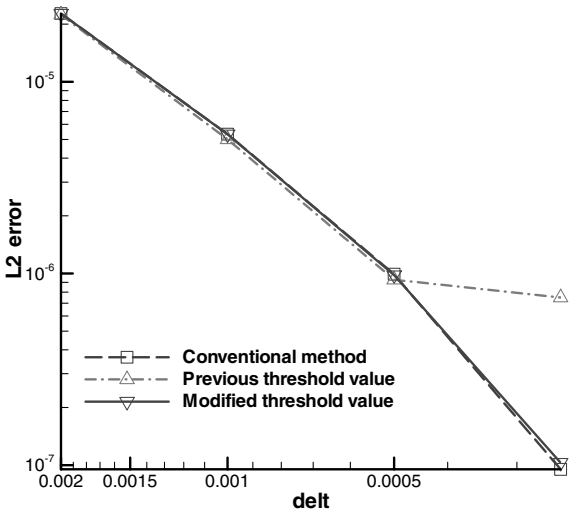


Fig. 6 L2 error according to Δt with $a = 0.5$.

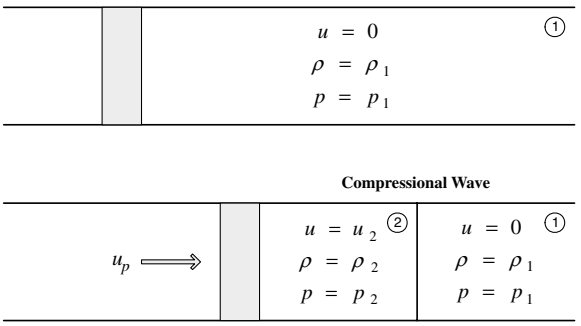


Fig. 7 Schematic diagram of the one-dimensional piston problem.

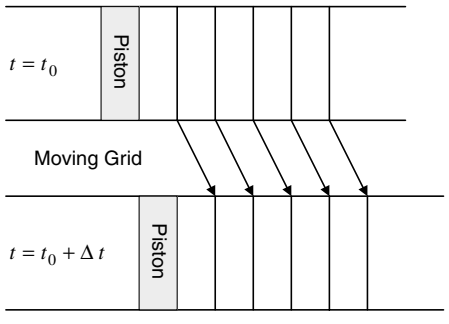


Fig. 8 Moving-grid system for the one-dimensional piston problem.

Table 2 L2 error for the piston problem

	Conventional	Level 1	Level 2	Level 3	Level 4
L2 error	8.47E − 05	8.52E − 05	8.52E − 05	8.52E − 05	8.52E − 05

From Eq. (21), a more than $(l+2)$th order interpolating polynomial is recommended for the accurate reconstruction of residuals. In a steady-state calculation, flow is converged into a fixed pattern and the $(l+2)$th order of polynomial may be enough [7]. However, we concentrated on the unsteady flow calculations. In the unsteady problems, tiny physical variations may grow and finally make a meaningful flow structure with time. In some cases, the $(l+2)$th order of accuracy is insufficient for the accurate residual reconstruction (see Sec. IV.B.) Therefore, at least the $(l+3)$th order interpolating polynomial is indispensable for the more precise calculation of the local flow features in an unsteady problem. In the present paper, the second-order spatial accuracy is used (i.e., $l=2$). Considering the symmetric formulation, the sixth-order interpolating polynomial is used at the end for residual interpolation, as shown in Eqs. (22).

For □:

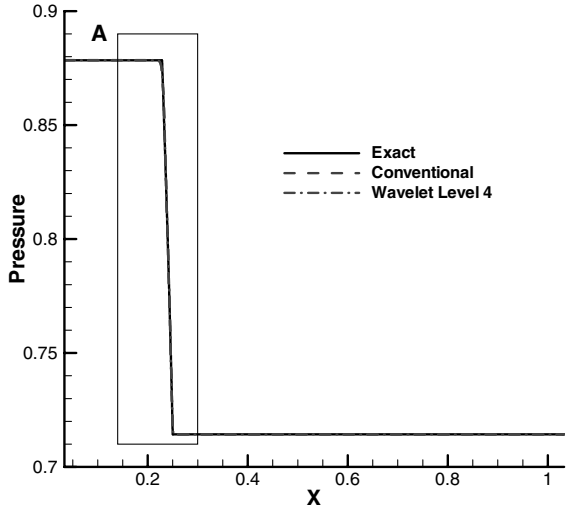
$$\tilde{R}^n_{i+1,j} = \frac{3}{256}R^n_{i-4,j} - \frac{25}{256}R^n_{i-2,j} + \frac{75}{128}R^n_{i,j} + \frac{75}{128}R^n_{i+2,j} - \frac{25}{256}R^n_{i+4,j} + \frac{3}{256}R^n_{i+6,j} \tag{22a}$$

For △:

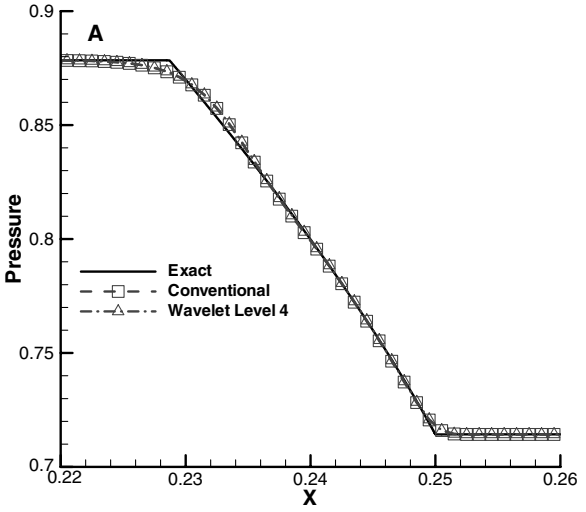
$$\tilde{R}^n_{i,j+1} = \frac{3}{256}R^n_{i,j-4} - \frac{25}{256}R^n_{i,j-2} + \frac{75}{128}R^n_{i,j} + \frac{75}{128}R^n_{i,j+2} - \frac{25}{256}R^n_{i,j+4} + \frac{3}{256}R^n_{i,j+6} \tag{22b}$$

For X:

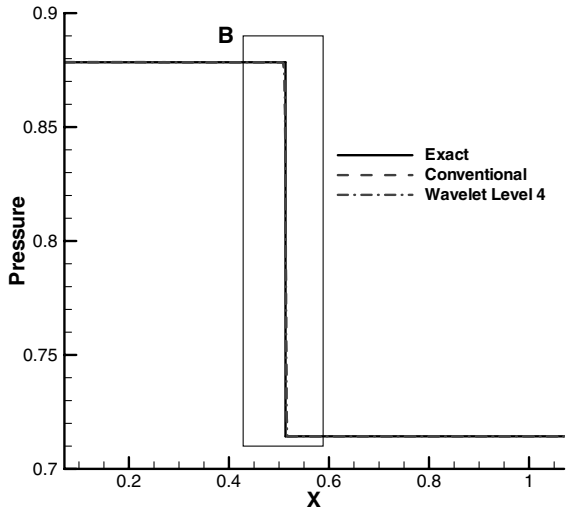
$$\tilde{R}^n_{i+1,j+1} = 0.5\times\left(\frac{3}{256}R^n_{i-4,j-2} - \frac{25}{256}R^n_{i-2,j-2} + \frac{75}{128}R^n_{i,j} + \frac{75}{128}R^n_{i+2,j+2} - \frac{25}{256}R^n_{i+4,j+4} + \frac{3}{256}R^n_{i+6,j+6}\right) + 0.5\times\left(\frac{3}{256}R^n_{i-4,j+6} - \frac{25}{256}R^n_{i-2,j+4} + \frac{75}{128}R^n_{i,j+2} + \frac{75}{128}R^n_{i+2,j} - \frac{25}{256}R^n_{i+4,j-2} + \frac{3}{256}R^n_{i+6,j-4}\right) \tag{22c}$$



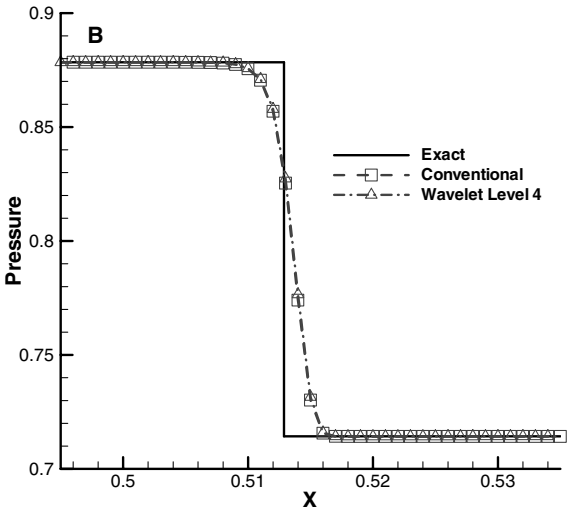
a) Pressure at 0.25 s



b) Magnified pressure plots in the region of A



c) Pressure at 0.5 s



d) Magnified pressure plots in the region of B

Fig. 9 Comparisons of pressure distributions.

After organizing the residual distribution in the whole computational domain, time integration is performed.

IV. Numerical Tests and Discussion

In the previous sections, we explained the overall procedure of the modified adaptive wavelet method. To demonstrate the performance of the method, several numerical simulations were executed. At first, the modified threshold value was assessed by the one-dimensional wave problems to show that the modified adaptive wavelet method maintains the numerical accuracy of the conventional solver. Then a complicated unsteady problem was solved according to the decomposition level of the wavelet method to confirm that the wavelet method improves the computational efficiency and maintains the solution's accuracy.

A. Assessment of the Modified Threshold Value

We tested the modified threshold value through the one-dimensional problems. The governing equation was the one-dimensional wave equation and the initial condition was a sine wave, as shown in Eq. (23).

$$\frac{\partial u}{\partial t} = a\frac{\partial u}{\partial x} \quad \text{with} \quad u(x,0) = \sin(2\pi x) \tag{23}$$

where $0 \le x \le 1$.

For boundary conditions, we used the periodic conditions in Eq. (24).

$$u_1 = u_{i\max,} \qquad u_{i\max+1} = u_2 \tag{24}$$

The number of grid points was set as $m \times 2^l + 1$ (where l is the number of the resolution level) for dyadic coarsening in the wavelet transformation.

First, convergence tests were performed with the decrease of Δx to verify the spatial accuracy according to threshold values. For this, we applied the second-order spatial discretization and the second-order Runge–Kutta time integration. We changed the number of grid points from 33 to 257; that is, Δx changed from 1/32 to 1/256. To make the spatial truncation error term dominant, a was defined as 1 and Δt and CFL were controlled to satisfy the condition of $\Delta x^2 \ge \mathrm{CFL}^2 \cdot \Delta x^2$ in all cases. And ε was set to 5×10^{-5}, and therefore $\varepsilon' = \min(5 \times 10^{-5}, \Delta x^2)$.

For comparison, we calculated L2 norm errors between the exact solution and the simulation results. Figure 5 shows the results of the conventional method and the wavelet methods with the previous and the modified threshold value according to Δx. Here, when the number of grid points was 33, 65, and 129, Δx^2 was larger than ε. On the other hand, in the case of 257 grid points, Δx^2 was smaller than ε. It is shown from Fig. 5 that both the previous and the modified threshold values provided the second-order spatial accuracy when Δx^2 was larger than ε. However, when Δx^2 became smaller than ε, only the modified threshold value maintained the numerical accuracy by switching the threshold value from ε to Δx^2.

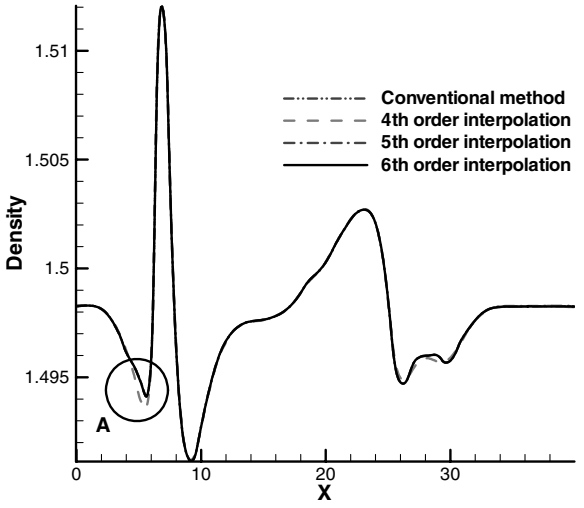
Second, to guarantee the temporal accuracy, convergence tests were performed with variation of Δt. Here, we set the number of grid points as 101 with $a = 0.5$ and changed Δt from 1/250 to 1/2000. Then the CFL changed from 1/5 to 1/40, accordingly. The fourth-order spatial discretization and the second-order Runge–Kutta time integration were selected to satisfy the condition of $\Delta x^4 \le \mathrm{CFL}^2 \cdot \Delta x^2$. Also, the value of ε was set as 2.5×10^{-6} to show the comparison of the accuracy clearly, then $\varepsilon' = \min(2.5 \times 10^{-6}, \mathrm{CFL}^2 \cdot \Delta x^2)$. Figure 6 shows the test results of the conventional method and the wavelet methods with the previous and the modified threshold values according to Δt. The previous threshold value did not provide the second-order temporal accuracy. On the other hand, the modified threshold value maintained the temporal accuracy of the conventional solver by switching the threshold value from ε to $\mathrm{CFL}^2 \cdot \Delta x^2$.

B. Assessment of the Computational Accuracy of the Modified Wavelet Method

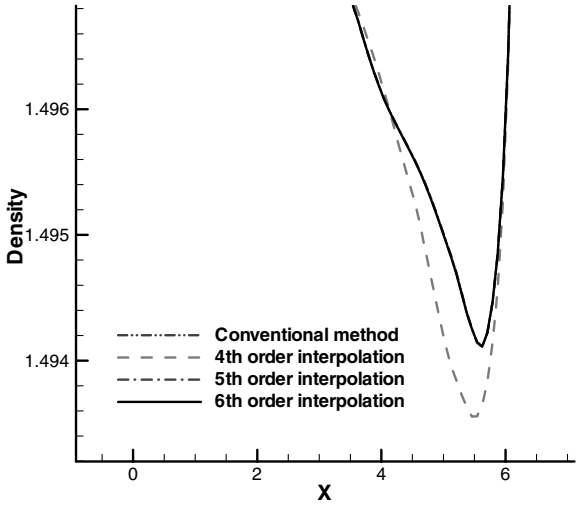
To assess the computational accuracy of the modified adaptive wavelet method, we computed a one-dimensional simple piston problem. The schematic diagram of this problem is shown in Fig. 7. The number of grid points is 1001 and the nondimensional time step is 0.00005. The initial condition is $(\rho_1, u_1, p_1) = (1, 0.7143, 0)$ and the threshold value ε is set as 10^{-5}. At first, the piston moves to the right with the constant acceleration of 2.5. After 0.06 s, the piston speed is constant: $u_{\text{piston}} = 0.15$. To express the movement of the piston, we adopted the moving-grid system without any deformation, as in Fig. 8. The exact solution was obtained by the method of characteristic line. For the computational results, we used the AUSMPW+ method [10], the second-order monotone upstream-centered schemes for conservation laws (MUSCL) scheme [11] with the van Leer limiter, and the fourth-order Runge–Kutta explicit method [12]. The boundary conditions are presented in Eq. (25).

$$\rho_1 = \rho_2, \qquad \rho_{i\max+1} = \rho_{i\max} \qquad u_1 = -u_2 + 2 \times u_{\text{piston}}$$
$$u_{i\max+1} = u_{i\max} \qquad p_1 = p_2, \qquad p_{i\max+1} = p_{i\max} \tag{25}$$

The L2 errors in Table 2 are the differences between the exact solution and the computational results at 0.5 s. The L2 error of the modified wavelet method is almost the same as that of the conventional solver. The pressure distributions of the exact solution,

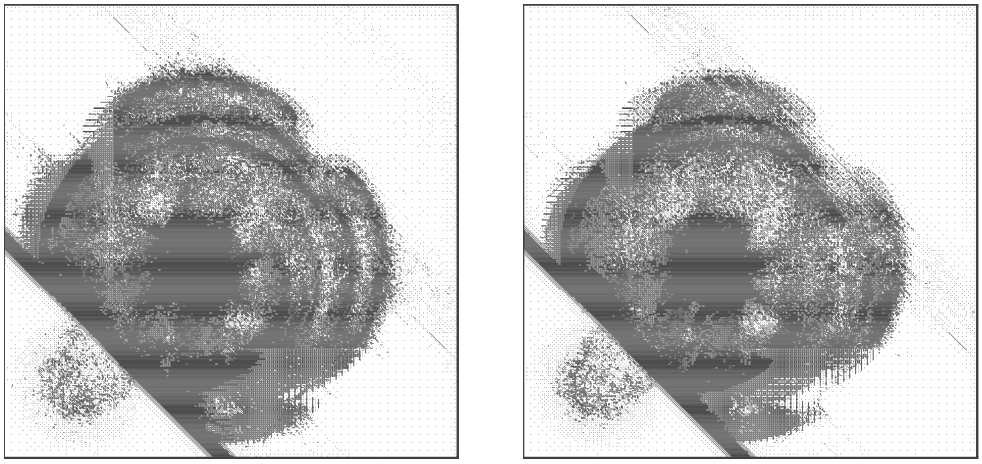


a) Comparison of density



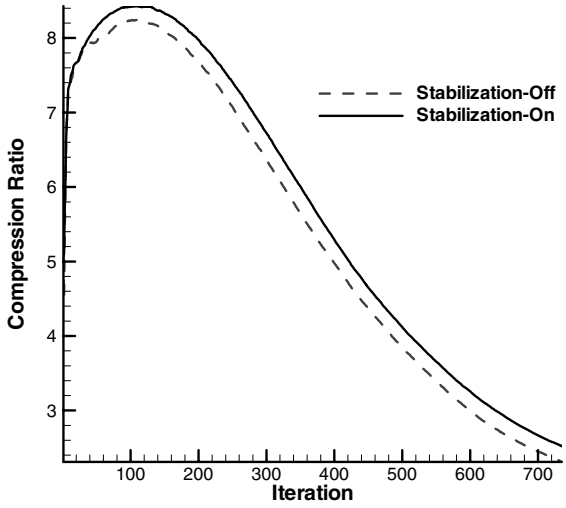
b) Comparison of density at A region

Fig. 10 Comparison of density of shock–vortex interaction problem according to the order of residual interpolation.



a) Adaptive dataset without stabilization method

b) Adaptive dataset with stabilization method



c) Comparison of compression ratio

Fig. 11 Comparison of adaptive data sets with/without the stabilization method.

the conventional solver, and the modified wavelet method with level 4 are presented in Fig. 9 at 0.25 and 0.5 s. Those of the conventional solver and the modified wavelet method agree well with the exact solutions, and the computational accuracy of the modified wavelet method is the same as that of the conventional solver.

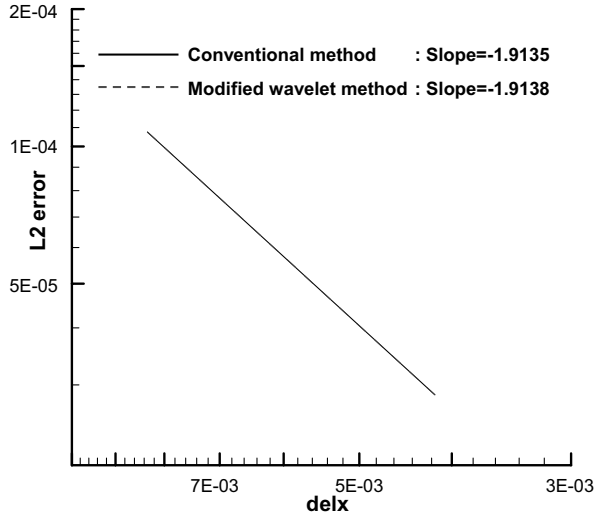


Fig. 12 Result of the grid convergence test.

C. Application of the Modified Wavelet Method to the Shock–Vortex Interaction Problem

To demonstrate that the modified wavelet method enhanced the computational efficiency, it was applied to a shock–vortex interaction problem. The computational domain was set as $0 \leq x$ and $y \leq 40$. The initial velocity, density, and pressure distributions of a vortex flow are presented in Eqs. (26) [13,14].

Tangential velocity:

$$u_\theta = M_v r \exp[(1 - r^2)/2] \tag{26a}$$

Radial velocity:

$$u_r = 0 \tag{}$$

Vorticity distribution:

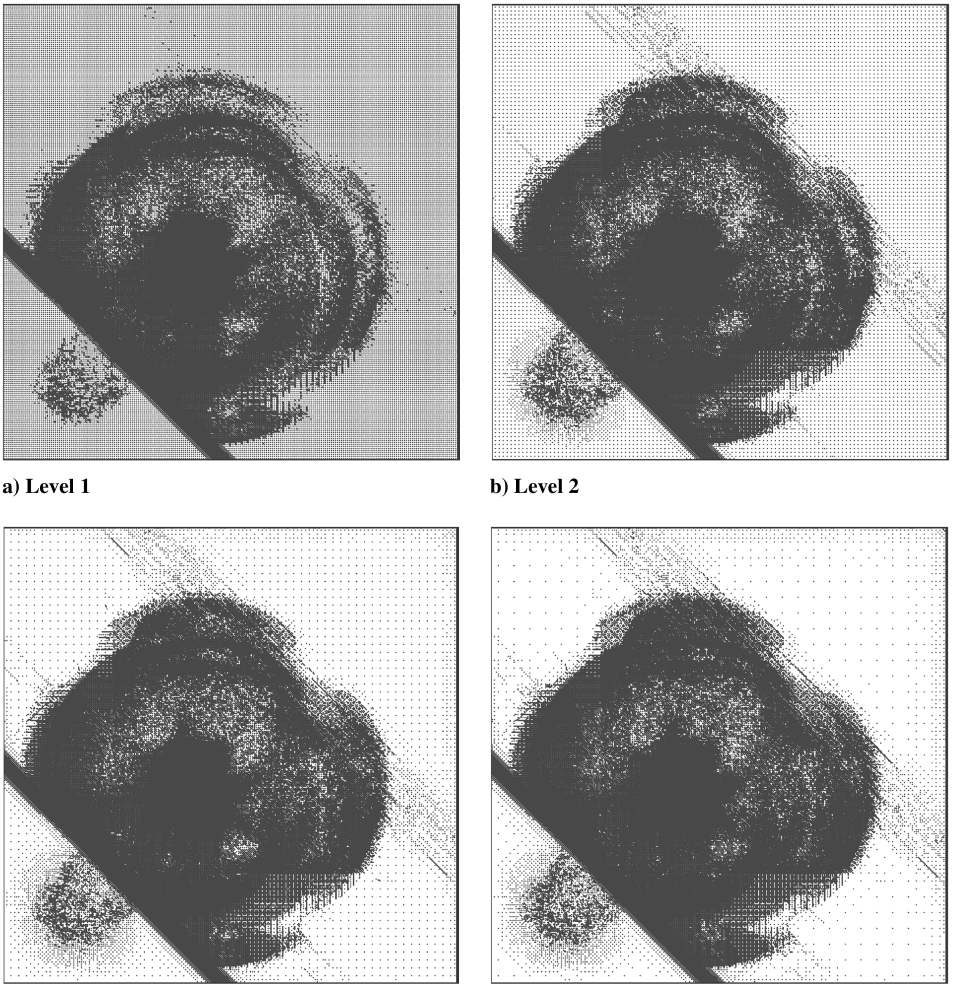
$$\omega(r) = M_v(2 - r^2) \exp[(1 - r^2)/2] \tag{26c}$$

Pressure distribution:

$$p(r) = \frac{1}{\gamma}\left[1 - \frac{\gamma - 1}{2} M_v^2 \exp(1 - r^2)\right]^{\gamma/(\gamma-1)} \tag{26d}$$

Density distribution:

$$\rho(r) = \frac{1}{\gamma}\left[1 - \frac{\gamma - 1}{2} M_v^2 \exp(1 - r^2)\right]^{1/(\gamma-1)} \tag{26e}$$



a) Level 1 b) Level 2

c) Level 3 d) Level 4

Fig. 13 Adaptive data sets of shock–vortex interaction problem according to a wavelet decomposition level at $t = 15$ s.

The Mach number of vortex M_v was 0.39 and the initial vortex core was located at the point (7, 7). This vortex propagated to a stationary normal shock of the Mach number 1.29. And the normal shock inclined by 45-deg angle at the point (10, 10). At the boundaries of the domain, boundary conditions are set as in Eqs. (27).

At $x = 0$,

$$Q_{i,j} = Q_{i+1,j-1} \tag{27a}$$

At $y = 0$,

$$Q_{i,j} = Q_{i-1,j+1} \tag{27b}$$

At $x = 40$,

$$Q_{i,j} = Q_{i-1,j+1} \tag{27c}$$

At $y = 40$,

$$Q_{i,j} = Q_{i+1,j-1} \tag{27d}$$

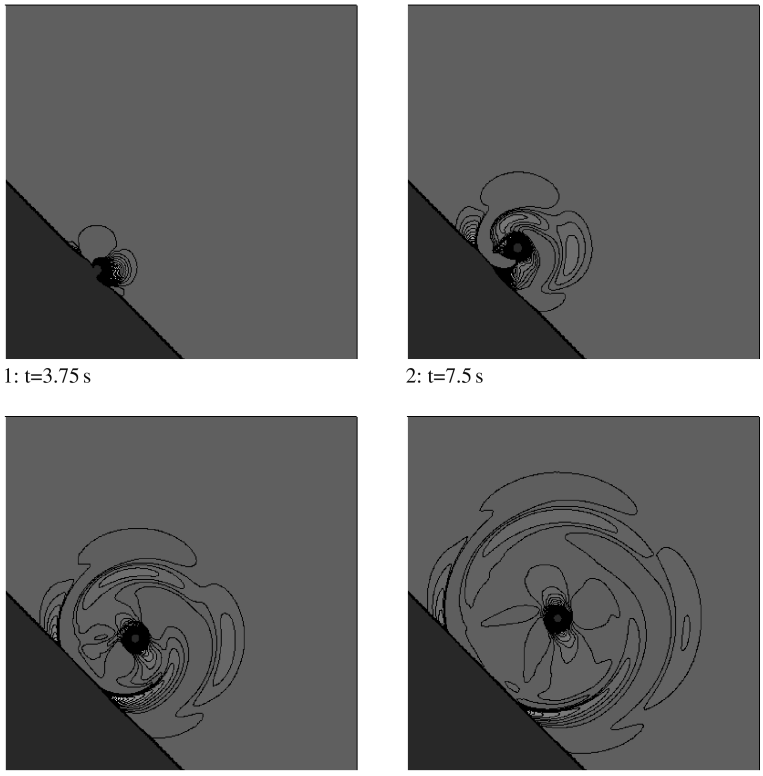
The governing equation was the two-dimensional Euler equations For spatial discretization, the AUSMPW+ method with the second-order MUSCL scheme was used. For time integration, the fourth-order Runge–Kutta method with CFL = 1 was used. The value of ε was set as 10^{-5}. The modified threshold value is presented as Eq. (28).

$$\varepsilon' = \min[10^{-5}, \max(\Delta x^2, \mathrm{CFL}^4 \cdot \Delta x^4)] \tag{28}$$

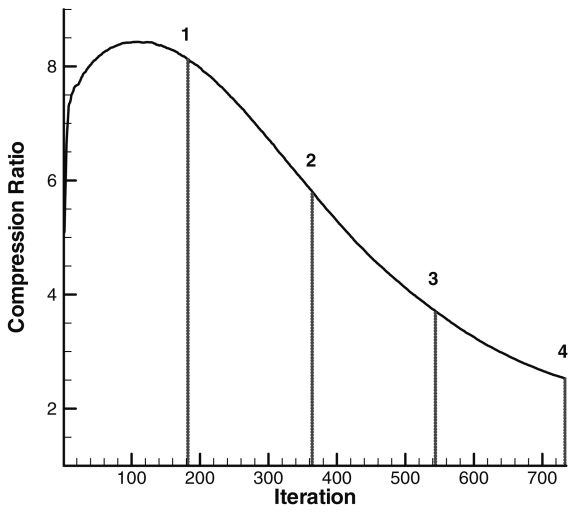
First, we checked the accuracy of the residual reconstruction numerically with the variation of the order of interpolating polynomial. The order of residual interpolation had to be two orders higher than the error order of residual values for the accurate reconstruction of the flow features. Because we performed the second-order spatial discretization, the residual values had the third order of accuracy. Figure 10 shows that the fourth-order residual interpolation was insufficient to reconstruct the inflection point on the density profile and to reconstruct the peak value, because time integration brought about integrated errors that decrease the one order of the spatial accuracy. On the other hand, an interpolating polynomial of more than fourth order represented them accurately.

Second, we tested the effect of the stabilization method on the construction of an adaptive data set. The adaptive data set with the stabilization method was compared with the data set without the method in Figs. 11a and 11b. Also, the compression ratio of the each case is shown in Fig. 11c. In these figures, the number of the remaining points in the data set with the stabilization method is less than that of the points without the stabilization method, and the compression ratio increases. It is because the $O(\varepsilon')$ error is transferred to the surrounding region and the $I(\varepsilon')$ data set changes regardless of the flow features. This defect can be limited with the stabilization method, and the data set is constructed stably and the compression ratio of the wavelet method increases.

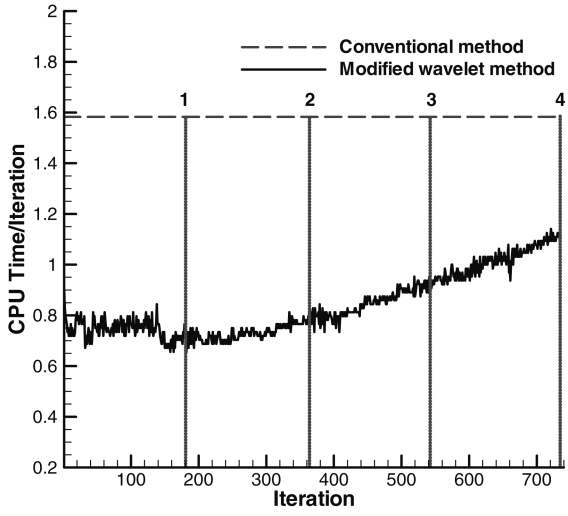
With the sixth-order residual interpolation and the stabilization method, the numerical simulations were performed to investigate the maintenance of the accuracy and the increase of the computational efficiency. At first, the convergence tests were performed with three grid systems: 121×121, 241×241, and 481×481. The errors between the finest and the middle and between the finest and the



a) Change of density according to time



b) Variation of compression ratio



c) Variation of CPU time/iteration

Fig. 14 Variation of flow features of shock–vortex interaction problem according to time.

coarse were computed and presented in Fig. 12. In this figure, the slope of the convergence line of the conventional solver is −1.9135 and that of the modified wavelet method is −1.9138. It is verified that the modified wavelet method has the second-order spatial accuracy as well as the conventional solver.

Figure 13 shows the adaptive data sets according to the wavelet resolution level at $t = 15$ s. Here, the data set follows the change of the flow features and many cells remain in $I(\varepsilon')$ near the vortex and shock regions. In the other regions, the changes of the flow properties are negligible and the remaining cells are sparsely distributed. The $I(\varepsilon')$ data set is effectively adapted to the solution's features. Also, in these figures, the adaptive data sets between levels 3 and 4 are very similar. There is a proper level of wavelet resolution according to the solution complexity. In this case, the appropriate level is 4.

Figure 14 shows the change of density, variation of the compression ratio, and CPU time/iteration during time integration, respectively. The CPU time/iteration of the conventional method was a constant 1.58 s. At the beginning of the calculation, that of the modified wavelet method was much smaller. However, as time went on, the moving vortex penetrated into and interacted with the stationary normal shock, which complicated the flow features. Then the CPU time/iteration of the modified wavelet method increased gradually because more grid points were included in the data set to adapt the data set to the flow features. Therefore, the compression ratio decreased and accompanied the increase of the CPU time/iteration of the modified wavelet method.

Overall, the computation time became about 2.0 times faster at maximum when the resolution level of wavelet was 4. The L2 norm

Table 3 Results of efficiency improvements and L2 error for the shock–vortex interaction problem with a 481 × 481 grid

Wavelet decomposition	L2 error	CPU time	Time ratio
Conventional		1162.27	
Level 1	3.96E − 08	921.75	1.26
Level 2	7.08E − 08	629.52	1.85
Level 3	1.26E − 07	581.58	2.00
Level 4	2.56E − 07	578.70	2.01

Table 4 Results of efficiency improvements and L2 error for the shock–vortex interaction problem with a 241 × 241 grid

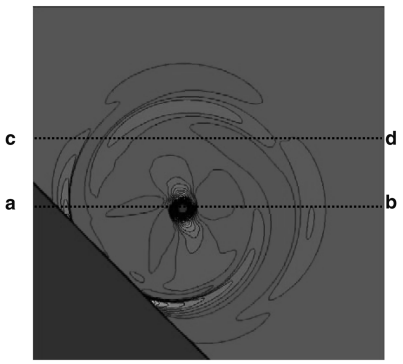
Wavelet decomposition	L2 error	CPU time	Time ratio
Conventional		140.08	
Level 1	4.77E − 08	112.47	1.25
Level 2	1.07E − 07	83.92	1.67
Level 3	2.91E − 07	80.48	1.74
Level 4	3.85E − 07	81.05	1.73

between the solutions of the conventional solver and the modified wavelet method was 2.56×10^{-7}. The overall efficiency improvement and the L2 error according to wavelet decomposition level are summarized in Table 3. Generally, if the number of the cells becomes large, the higher computational efficiency can be obtained because the portion of the region in which the flow properties change smoothly increases and the compression ratio of the wavelet method is enhanced. To confirm this result, we performed the same simulations with 241×241 grids and summarize the results in Table 4. Although the order of L2 errors was very similar for all the cases, the maximum improvement of the computational efficiency was reduced from 2.0 to 1.74 times. The present method can become more effective in the flow problems that require a huge size of grid points.

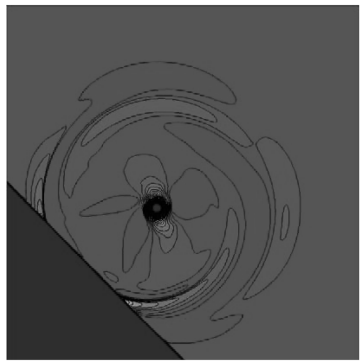
Figures 15a and 15b show the density distributions of the conventional and the modified wavelet methods, respectively, at $t = 15$ s. The comparisons of the density distributions at $y = 17$ (a–b line) and $y = 25$ (c–d line) are presented in Figs. 15c and 15d, respectively. The density distributions are very similar to each other. Also in Figs. 15c and 15d, the original features around the vortex core or the shock discontinuity region are exactly represented by the modified wavelet method.

V. Conclusions

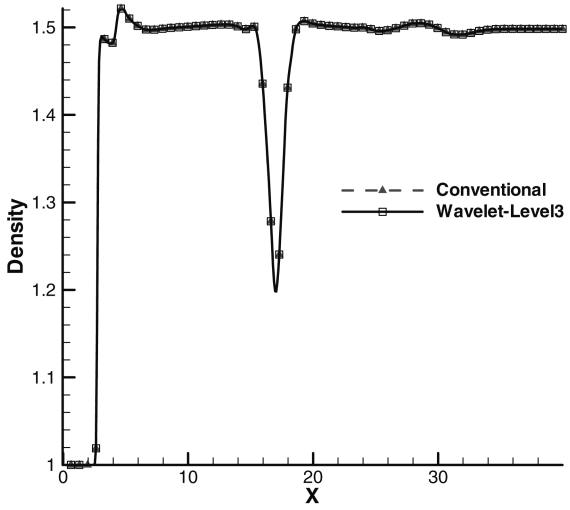
The modified adaptive wavelet method was proposed for the unsteady flow problems to increase the overall computational efficiency of a conventional CFD solver while guaranteeing the numerical accuracy of the solutions. First, residual interpolation was performed at the n time step, not at the $n + 1$ time step. It enables the process of arbitrary addition of adjacent cells to a $I(\varepsilon')^n$ data set to be excluded from the wavelet process. Therefore, the $I(\varepsilon')^n$ data set can be automatically constructed without the user's arbitrary adjustment of the threshold value. Second, the threshold value was modified to maintain the spatial and the temporal accuracies of a conventional scheme, simultaneously. For this aim, we changed the temporal truncation error term to the spatial truncation error term based on the analysis of the order-of-error terms. Then ε was compared with the dominant error term between the spatial truncation error and the reformed temporal truncation error. Between ε and the dominant



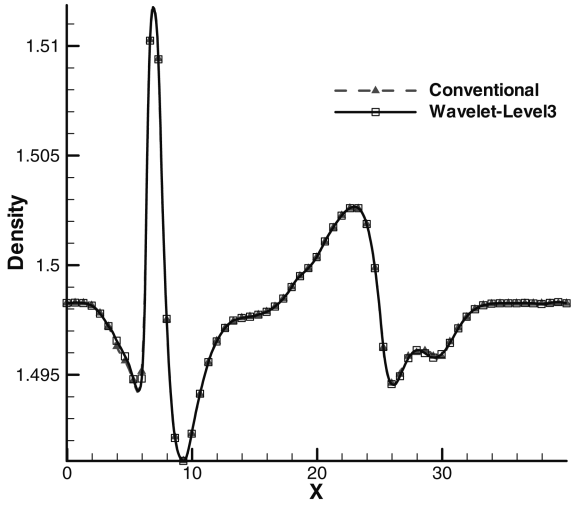
a) Conventional method



b) Modified wavelet method



c) Comparison of density at y=17



d) Comparison of density at y=25

Fig. 15 Density plots of shock–vortex interaction problem at $t = 15$ s; the wavelet decomposition level is 3.

error term, the smaller value was selected as ε' to guarantee the accuracy of the conventional solver.

Acknowledgments

The authors gratefully acknowledge that this research was supported by the Brain Korea 21 program for the Mechanical and Aerospace Engineering Research at Seoul National University, by the Center of Innovative Design Optimization Technology, Korea Science and Engineering Foundation and by grant R01-2006-000-10034-0 from the Basic Research Program of the Korea Science and Engineering Foundation. Also the authors gratefully acknowledge the financial support by the Agency for Defense Development and by the Flight Vehicle Research Center (FVRC), Seoul National University.

References

[1] Berger, M. J., and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, No. 1, 1989, pp. 64–84.
doi:10.1016/0021-9991(89)90035-1

[2] Harten, A., "Adaptive Multiresolution Schemes for Shock Computation," *Journal of Computational Physics*, Vol. 115, No. 2, 1994, pp. 319–338.
doi:10.1006/jcph.1994.1199

[3] Holmström, M., "Solving Hyperbolic PDEs Using Interpolation Wavelets," *SIAM Journal on Scientific Computing*, Vol. 21, No. 2, 1999, pp. 405–420.
doi:10.1137/S1064827597316278

[4] Sjögreen, B., "Numerical Experiments with the Multiresolution Scheme for the Compressible Euler Equations," *Journal of Computational Physics*, Vol. 117, No. 2, 1995, pp. 251–261.
doi:10.1006/jcph.1995.1063

[5] Vasilyev, O. V., and Paolucci, S., "A Fast Adaptive Wavelet Collocation Algorithm for Multidimensional PDEs," *Journal of Computational Physics*, Vol. 138, No. 1, 1997, pp. 16–56.
doi:10.1006/jcph.1997.5814

[6] Chiavassa, G., and Donat, R., "Point Value Multiscale Algorithms for 2D Compressible Flows," *SIAM Journal on Scientific Computing*, Vol. 23, No. 3, 2001, pp. 805–823.
doi:10.1137/S1064827599363988

[7] Kang, H., Kim, K., Lee, D., and Lee, D., "Improvement in Computational Efficiency of Euler Equations via a Modified Sparse Point Representation Method," *Computers and Fluids* (submitted for publication).

[8] Donoho, D. L., "Interpolating Wavelet Transforms," Stanford Univ., Dept. of Statistics, TR 408, Stanford, CA, 1992.

[9] Deslauriers, G., and Dubuc, S., "Symmetric Iterative Interpolation Processes," *Constructive Approximation*, Vol. 5, No. 1, 1989, pp. 49–68.
doi:10.1007/BF01889598

[10] Kim K. H., Kim, C., and Rho, O., "Methods for the Accurate Computations of Hyperbolic Flows, 1: AUSMPW+ Scheme," *Journal of Computational Physics*, Vol. 174, No. 1, 2001, pp. 38–80.
doi:10.1006/jcph.2001.6873

[11] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V: A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 135, No. 2, 1997, pp. 229–248.
doi:10.1006/jcph.1997.5704

[12] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA, AIAA Paper 1981-1259, June 1981.

[13] Inoue, O., and Hattori, Y., "Sound Generation by Shock–Vortex Interactions," *Journal of Fluid Mechanics*, Vol. 380, Feb. 1999, pp. 81–116.
doi:10.1017/S0022112098003565

[14] Kim, K. H., and Kim, C., "Accurate, Efficient and Monotonic Numerical Methods for Multidimensional Compressible Flows, Part 2: Multi-Dimensional Limiting Process," *Journal of Computational Physics*, Vol. 208, No. 2, 2005, pp. 570–615.
doi:10.1016/j.jcp.2005.02.022

T. Jackson
Associate Editor