



A high-order flux reconstruction method with adaptive mesh refinement and artificial diffusivity on unstructured moving/deforming mesh for shock capturing



Jingjing Yang^{a,*}, Bin Zhang^a, Chunlei Liang^a, Yongwu Rong^b

^a Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC 20052, United States

^b Department of Mathematics, The George Washington University, Washington, DC 20052, United States

ARTICLE INFO

Article history:

Received 13 November 2015

Revised 21 March 2016

Accepted 23 March 2016

Available online 1 April 2016

Keywords:

Flux reconstruction

Shock capturing

Artificial diffusivity

Adaptive mesh refinement

Moving/deforming mesh

H-adaptation

ABSTRACT

This article presents new progress of shock capturing on unstructured dynamic grids by using adaptive mesh refinement (AMR) method in conjunction with artificial diffusivity (AD) under the framework of flux reconstruction (FR)/correction procedure via reconstruction (CPR) method. The proposed AMR algorithm is applicable even when the grid undergoes dynamic motion. It features an innovative data structure that is capable of direct addressing in managing the system of cells on multiple levels of refinement. A conservative mortar method is used to handle non-conforming interfaces resulted from cell subdivision. The AD is added in regions of strong dilatation to counter the Gibbs oscillation that is disruptive to high-order solutions when shock discontinuities exist. Several test cases are used to verify that this FR-AMR-AD framework can achieve high-order accuracy for smooth flows, and obtain stable solution with the presence of shock discontinuity. By further limiting the shock within subdivided adjacent cells, a sharp shock solution can be obtained with reduced amount of AD addition, hence errors from adding dissipation, on dynamically refined grids.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The task of resolving shock discontinuity is an integral part of a wide range of numerical simulations, such as shock/boundary-layer interaction, scramjet and space weather modelling. Developing high-order accurate, non-dissipative numerical schemes to resolve shock discontinuities and various scales of turbulence is critical to understand and predict the complex physical phenomena. Constructing numerical schemes that are able to achieve high-order accuracy, non-dissipation and capture a sharp representation of shock discontinuities is especially challenging in the aforementioned applications where unsteady, non-linear effects are dominant. Besides, the formidable amount of grids in these simulations in order to capture the complex physics calls for more efficient, dynamic way of grid management methods.

High-order methods are known for their ability to achieve accurate solution on relatively coarse grids. FR/CPR [38] method is one of this family, which is related to spectral difference (SD) method introduced by Kopriva and Kolas [27]. SD method adopts

a differential form of governing equations, and locally approximates the solution with element-wise high degree polynomials. Huynh [18,20] introduced FR method as a new approach to achieve high-order accuracy and resolution. FR method improves the efficiency of SD method by using a number of flavours of correction functions. This method was applied in solving diffusion problems with quadrilateral meshes by Huynh [19]. Wang and Gao [44] extended it the triangular and mixed meshes under lifting collocation penalty framework. FR is considered one of the most efficient high-order schemes in terms of the number of operations. Liang [28,29] reported 27% of efficiency gain by FR over Spectral Difference for inviscid flow and 40% for viscous flow in some cases.

With the presence of shock waves, the lack of an effective mesh refinement mechanism often leads to divergence of the solution. Shock-fitting or using a size function during mesh generation can be effective, provided that where to refine is already known. In contrast, requiring no *a priori* knowledge, AMR method is capable of dynamically increasing mesh resolution near shock discontinuities, and capture a sharp profile. The difficulty arising with AMR is the management of the data structure. To overcome the immutability of preallocated (dynamic/static) arrays, a tree-like data structure is usually used to track the addition/deletion of cells on the fly. Data structures that fall in this category include list-based

* Corresponding author. Tel.: +170333030662.

E-mail addresses: umb_young@gwu.edu (J. Yang), bzh@gwu.edu (B. Zhang), chliang@gwu.edu (C. Liang), rong@gwu.edu (Y. Rong).

[5], cell-based [35] and block-based [40] structures. The newly created sub-cells, along with its associated information, will be bundled as a node, then appended or inserted to a tree-like data structure. Therefore, they will face the problem of indirect addressing when retrieving information on a certain node. The complexity grows substantially as the size of the hierarchical grid system grows. How often the remeshing occurs is also a factor. Berger and Oliger [11] firstly used a logically rectangular patch which is suitable for tracing moving discontinuities on Cartesian grids. Collela [2] combines this technique with boundary embedded method to be able to apply on complex geometries. Although it is still hard to achieve high-order accuracy or extend to three dimensional simulations. Ying [46] used space filling curve (SFC) to interpret the hierarchical information along the curve without resorting to linear search. Although indirect addressing is avoided, this method still has its limitation on unstructured or moving/deforming grid. The AMR method proposed here combines the properties of SFC and cell-based structure. It is capable of direct addressing on moving/deforming grids. Only minimal run-time cost occurs during data retrieving. Depending on the choice of refining criteria, a certain cell or a patch of cells can be easily refined to multiple levels of sub-cells. We term this property as spatio-hierarchical awareness. Details will be discussed in Section 3.2.

AD method can remove spurious oscillation and acquire stable and physical-related solutions by adding localized diffusion to shock regions. It can be easily extended to multi-dimensional, complex geometry, compared to developing a flux limiting scheme. The original idea for AD is pioneered by von Neumann and Richtmeyer [43]. Non-linear artificial viscosity is added uniformly, thus the removal of shock is at the expense of smearing discontinuity. Jameson et al. [23] constructed flexible addition of AD, which depends on local pressure gradient, in the context of unstructured finite volume method to produce steep representation of shocks. Cook and Cabot [13] proposed a method in a high-order centered differencing framework to dynamically add high-wave number biased AD. A sensor based on the magnitude of highest-order coefficients is used to combine high-order discretization and piecewise-constant AD. Persson and Peraire [34] used a sensor based on the highest-order coefficient of the solution to detect and introduce element-wise AD in discontinuous Galerkin (DG) and demonstrate sub-cell resolution. Also in DG method, Barter and Darmofal [6] proposed a PDE-based, isenthalpic formulation of AD. Haga and Kawai [17] extended localized AD to FR method with a filter for smooth representation on unstructured grid. Kawai and Lele [24] and Premasuthan et al. [36] extended and modified Cook and Cabot's idea to curvilinear and unstructured grids, wherein grid-dependent, localized transport coefficients were dynamically added. The current study follows their formulation and incorporates it onto moving/deforming mesh with dynamic grid adaptivity.

This article is organized as the following: Section 2 introduces the governing equations, Section 3 addresses FR method, the principles and working conditions of the AMR method, the mortar method, and followed by the details of the formula of the AD method. Section 4 includes the numerical tests.

2. Governing equations

2D compressible Euler equations in the conservative form are considered for modelling. The governing equations are

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0, \quad (1)$$

where $\mathbf{Q} = [\rho, \rho u, \rho v, E]^T$ is the vector of the conservative variables and \mathbf{F} , \mathbf{G} are inviscid fluxes, which have the vector

representation as the following:

$$\mathbf{F} = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(E + p) \end{Bmatrix} - \mathbf{F}_{\text{ad}}, \quad (2)$$

$$\mathbf{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(E + p) \end{Bmatrix} - \mathbf{G}_{\text{ad}}, \quad (3)$$

in which, total energy can be expressed as $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2)$, and $\gamma = 1.4$ is the ratio of specific heats. \mathbf{F}_{ad} and \mathbf{G}_{ad} are artificially added viscous flux, and will be discussed in the Section 3.4.

Iso-parametric mapping is employed to relate each physical cell of an unstructured mesh (x, y) to a standard computational element $(\xi, \eta) \in [0, 1] \times [0, 1]$, which allows universal polynomial reconstruction regardless the actual size/shape of the grids. The transformation can be described as

$$x(\xi, \eta, \tau) = \sum_{i=1}^K M_i(\xi, \eta) x_i(\tau) \quad (4)$$

where K is the number of points used to define the physical element, and M_i are shape functions. The derivatives of Eq. 4 forms the Jacobian matrix and its inverse

$$\mathcal{J} = \frac{\partial(x, y, t)}{\partial(\xi, \eta, \tau)} = \begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$$\mathcal{J}^{-1} = \frac{\partial(\xi, \eta, \tau)}{\partial(x, y, t)} = \begin{bmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & 1 \end{bmatrix} = \frac{1}{|\mathcal{J}|} S, \quad (6)$$

where S and $|\mathcal{J}|$ are the transpose of the cofactor matrix and the Jacobian of \mathcal{J} , respectively.

The transformed equations on the computational domain take the forms

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial \tau} + \frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} = 0, \quad (7)$$

where

$$\begin{bmatrix} \tilde{\mathbf{F}} \\ \tilde{\mathbf{G}} \\ \tilde{\mathbf{Q}} \end{bmatrix} = |\mathcal{J}| \mathcal{J}^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \\ \mathbf{Q} \end{bmatrix},$$

Take into account the Geometric Conservation Law (GCL)

$$\begin{cases} \frac{\partial |\mathcal{J}| \xi_x}{\partial \xi} + \frac{\partial |\mathcal{J}| \eta_x}{\partial \eta} = 0, \\ \frac{\partial |\mathcal{J}| \xi_y}{\partial \xi} + \frac{\partial |\mathcal{J}| \eta_y}{\partial \eta} = 0, \\ \frac{\partial |\mathcal{J}|}{\partial \tau} + \frac{\partial}{\partial \xi} (|\mathcal{J}| \xi_\tau) + \frac{\partial}{\partial \eta} (|\mathcal{J}| \eta_\tau) = 0. \end{cases} \quad (8)$$

The first two in Eq. 8 are satisfied automatically if theisoparametric mapping is defined analytically. Correctly enforcing the GCL depends on the calculation of the evolving metrics and $|\mathcal{J}|$. If we expand Eq. 7, the solution variables $\tilde{\mathbf{Q}}$ and the fluxes $\tilde{\mathbf{F}}$, $\tilde{\mathbf{G}}$ in the computational space will acquire the following expression,

$$\begin{cases} \tilde{\mathbf{Q}} = |\mathcal{J}| \mathbf{Q}, \\ \tilde{\mathbf{F}} = |\mathcal{J}| (\mathbf{Q} \xi_t + \mathbf{F} \xi_x + \mathbf{G} \xi_y), \\ \tilde{\mathbf{G}} = |\mathcal{J}| (\mathbf{Q} \eta_t + \mathbf{F} \eta_x + \mathbf{G} \eta_y). \end{cases} \quad (9)$$

Notice that in free-stream situation where \mathbf{Q} is constant, $\frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta}$ will contain a term,

$$\mathbf{Q} \left(\frac{\partial |\mathcal{J}| \xi_t}{\partial \xi} + \frac{\partial |\mathcal{J}| \eta_t}{\partial \eta} \right). \quad (10)$$

Failure to enforce the GCL will be equal to introduce additional source term, and make the governing equations non-conservative. In [1,33], the metrics are solved as a conservative variable with the same time integrator. What we used in the current study, similar to [47,48], is to introduce a source term to cancel Eq. 10 at every stage of the Runge-Kutta scheme, and calculate $|\mathcal{J}|$ directly according to the grid motion.

This idea can be illustrated by examining the formulation of residual. The first term in Eq. 7 can be expressed as

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial \tau} = \frac{\partial (|\mathcal{J}| \mathbf{Q})}{\partial \tau} = |\mathcal{J}| \frac{\partial \mathbf{Q}}{\partial \tau} + \mathbf{Q} \frac{\partial |\mathcal{J}|}{\partial \tau}. \quad (11)$$

Substitute the GCL into the above equation and rearrange, we will obtain the residual \mathcal{R}

$$\mathcal{R} = \frac{\partial \mathbf{Q}}{\partial \tau} = \frac{1}{|\mathcal{J}|} \left\{ - \left[\frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} \right] + \mathbf{Q} \left[\frac{\partial (|\mathcal{J}| \xi_t)}{\partial \xi} + \frac{\partial (|\mathcal{J}| \eta_t)}{\partial \eta} \right] \right\} \quad (12)$$

$$= \frac{1}{|\mathcal{J}|} \left\{ - \left[\frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} \right] + source \right\}. \quad (13)$$

Therefore, GCL is satisfied by using the source term above to counter the Eq. 10, which arises from the unsteady transformation. The spatial convergence tests can be found in Section 4.1.

3. Numerical methods

3.1. FR method

In the current FR method, the conservative variables and fluxes are interpolated through Legendre-Gauss polynomial. Solution points are placed on the roots of Legendre-Gauss polynomial, which are entirely interior points of the computational cell. N quadrature points are used to construct $(N-1)$ order polynomial in each direction through Lagrange interpolating basis function

$$h_i(x) = \prod_{s=1, s \neq i}^N \left(\frac{X - X_s}{X_i - X_s} \right). \quad (14)$$

The reconstructed polynomial for conservative variables in the standard element is a tensor product of two one-dimensional polynomials

$$\mathbf{Q}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \frac{\tilde{\mathbf{Q}}_{i,j}}{|\mathcal{J}_{i,j}|} h_i(\xi) \cdot h_j(\eta). \quad (15)$$

The reconstructed flux polynomials take the form of element-wise continuous function

$$\tilde{\mathbf{F}}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \tilde{\mathbf{F}}_{i,j} h_i(\xi) \cdot h_j(\eta), \quad (16)$$

$$\tilde{\mathbf{G}}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \tilde{\mathbf{G}}_{i,j} h_i(\xi) \cdot h_j(\eta). \quad (17)$$

Fig. 1 shows that a 4th-order FR has collocated solution/flux points in the cell interior, which shows \mathbf{F} and \mathbf{G} are stored on the same set of quadrature points. And the boundary flux points are used to calculate $\tilde{\mathbf{F}}_{ij}^c$ (continuous flux representation within a cell) through the correction function.

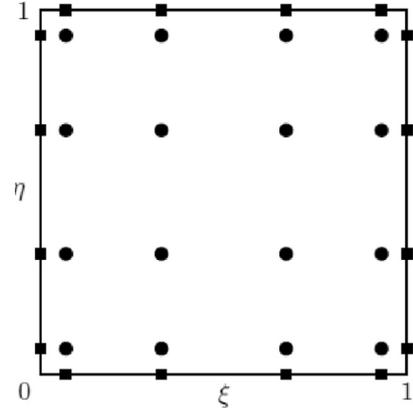


Fig. 1. 4th Order FR, solution (circle) and flux (square) points placement.

Those flux points on the boundary are responsible for calculating the common fluxes on the interfaces of two adjacent cells. Inviscid common fluxes at interfaces are calculated using Riemann solver, for example, Rusanov solver

$$\tilde{\mathbf{F}}_{ij}^{com} = \frac{1}{2} \{ \mathbf{F}_L + \mathbf{F}_R - (|\bar{\mathbf{V}}_n| + \bar{c}) \cdot (\mathbf{Q}_R - \mathbf{Q}_L) \cdot |\bar{S}_\xi| \cdot \text{sign}(\bar{n} \cdot \bar{S}_\xi) \}, \quad (18)$$

where \bar{n} is the normal direction of interfaces, $\bar{\mathbf{V}}_n$ is the normal velocity component and \bar{c} is the sound of speed. A correction procedure is then applied to form a continuous flux. A continuous flux can be mathematically described in ξ direction as

$$\tilde{\mathbf{F}}_{ij}^c(\xi) = \tilde{\mathbf{F}}_{ij}(\xi) + \left[\tilde{\mathbf{F}}_{j-\frac{1}{2}}^{com} - \tilde{\mathbf{F}}_j(0) \right] g_{LB}(\xi) + \left[\tilde{\mathbf{F}}_{j+\frac{1}{2}}^{com} - \tilde{\mathbf{F}}_j(1) \right] g_{RB}(\xi), \quad (19)$$

in which $\tilde{\mathbf{F}}_{j-\frac{1}{2}}^{com}$ and $\tilde{\mathbf{F}}_{j+\frac{1}{2}}^{com}$ are common flux on the left and right cell interface, $\tilde{\mathbf{F}}_j(0)$ is discontinuous flux at cell interface, and i, j are solution/flux points indices. The correction function g of interest is casted as g_{DG} in [19]. It is defined on the left cell boundary as

$$g_{DG, LB} = \frac{(-1)^N}{2} (P_N - P_{N-1}), \quad (20)$$

where P_N represents n -th order Legendre polynomial. $g_{DG, LB}$ is required to be 1 at $\xi = 0$ and 0 at $\xi = 1$. N -th order g_{DG} is required to vanish at $N-1$ Legendre-Gauss quadrature points. These points are roots of P_{N-1} , and are different locations from the roots of P_N , which is the choice of solution points on N -th order. In other words, g_{DG} requires correction to all interior solution points. The continuous flux \mathbf{G}_{ij}^c and its derivative in η direction can be obtained with the same procedures.

3.2. Direct-addressing AMR on moving/deforming mesh

In the proposed AMR data structure, each geometry entity – cell, face or vertex – has a unique ID in global integer indexing space. This approach is similar to the one used in Bell et al. [9], while we extended it to unstructured, dynamic grids from Cartesian grids. The representation of geometry entities takes into accounts its geometrical and hierarchical dependency. To demonstrate how this way of representation leads to direct-addressing, an example of an arbitrary quadrilateral two dimensional grid is given in Fig. 2. If cell k on level i is about to be refined with a refinement ratio of 4, then its sub-cells on level i will be $4k + N_i + p$, in which N_i is the cell numbers on level i and $1 \leq p \leq 4$. This simple incremental relationship completely eliminates the need for recursive search in order to locate the sub-cells of any selected cell. The connectivity within each level can be precalculated and stored during the initialization stage of the code. Face/vertex tagging can

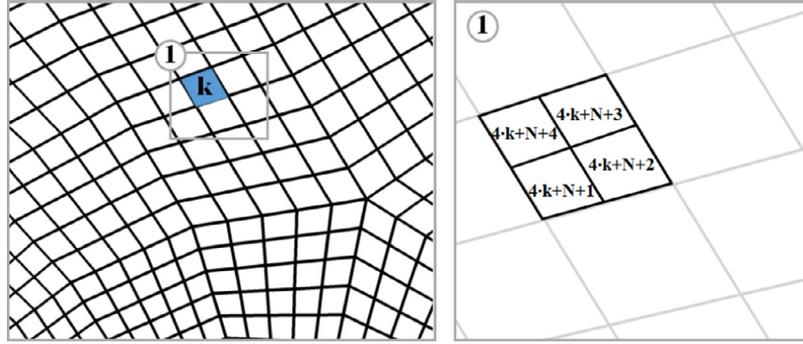


Fig. 2. Cell k on level i (left) and its sub-cells on level $i + 1$ (right). N is the number of cells on level i .

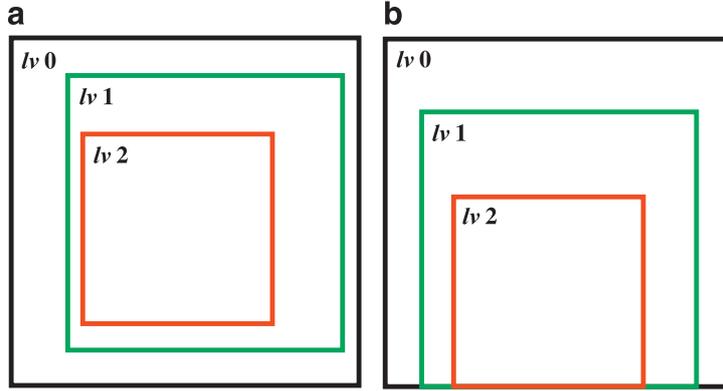


Fig. 3. Proper-nesting assumption.

be achieved in the same fashion. The proposed AMR data structure results in a clean and robust code. We proceed to discuss the procedures and conditions of implementing the AMR method, which is followed by an analysis of the merits and trade-off.

The procedures for performing AMR on moving/deforming grid is given in Algorithm 1. Currently, as a precursory study, we use the same time step size for grids on all levels of refinement. The maximum level of refinement is arbitrarily determined at the beginning of the calculation.

Algorithm 1 AMR Refining

```

 $maxLoR \leftarrow a$  ▷ maximum level of refinement
 $\Delta t \leftarrow b$  ▷ global time step size
 $\sigma \leftarrow c$  ▷ an very small positive number
procedure ADAPTIVE REFINING ON DYNAMIC GRIDS( $q, \beta_{\Delta}, m_o$ )
  Input conservative variables  $q$ , artificial bulk viscosity  $\beta_{\Delta}$  and
  motion parameter  $m_o$ 
  repeat
    for  $i \leftarrow 0, maxLoR - 1$  do ▷ refine
      for  $j \leftarrow 1, N_i$  do
        if  $\beta_{av}(j, i) \geq \sigma$  then
          Refine
           $q(4 \cdot j + N_i + k, i + 1) \leftarrow q(j, i), 1 \leq k \leq 4$  ▷  $L^2$ 
        projection
      end if
    end for
    end for
    check proper-nesting principle
    check inter-level boundary alignment
    update Jacobian etc. according to grid motion  $m_o$ 
  until  $t = t_{terminal}$ 
end procedure

```

It is shown that in Algorithm 1, non-zero artificial bulk viscosity β_{Δ} is chosen as the sensor to trigger refinement. We will show in Section 3.4 that β_{Δ} is regulated by a switch s_{β} , which acts as a good indication for strong dilatation in the cases where shocks are present. The procedure for coarsening can be defined in a similar fashion. The criterion for coarsening is simply the opposite, which is that β_{Δ} of all sub-cells are arbitrarily small at the same time. How frequently Algorithm 1 and its coarsening counterpart are enabled in response to the change in flow field is arbitrarily set.

3.2.1. Four enabling conditions

Furthermore, four conditions are enforced to enable this method. The first condition is called the properly-nested assumption [10,16,46], which is illustrated in Fig. 3. This condition consists of three requirements. First, a level $i + 1$ refined region of cells is contained in the interior of level i refined region, unless the fine and coarse regions coincide with each other on the boundaries. Second, the coinciding boundaries can only align on the boundary of the entire computational domain, instead of the interior, as in Fig. 3(b). Third, the difference in level of refinement (LoR) between adjacent cells can not be greater than one. Fig. 4 shows how to apply this requirement to maintain grid validity, when attempting to make a level $i + 1$ cell, all the neighboring cells have to be on level i . This requirement also simplifies the type of mortar interface that are encountered in implementing our AMR method. Since the refinement ratio is 4 (i.e. one targeted cell is subdivided into 4 sub-cells), the only possible mortar is the one shown in Fig. 11(b). Specifically, the non-conforming interfaces maintain a 2-to-1 configuration, that is 2 contributing sub-domain interfaces on one side, and 1 contributing sub-domain interface on the other side.

The second condition also concerns grid quality. As shown in Fig. 5, when refining cells on curved boundary, the newly created

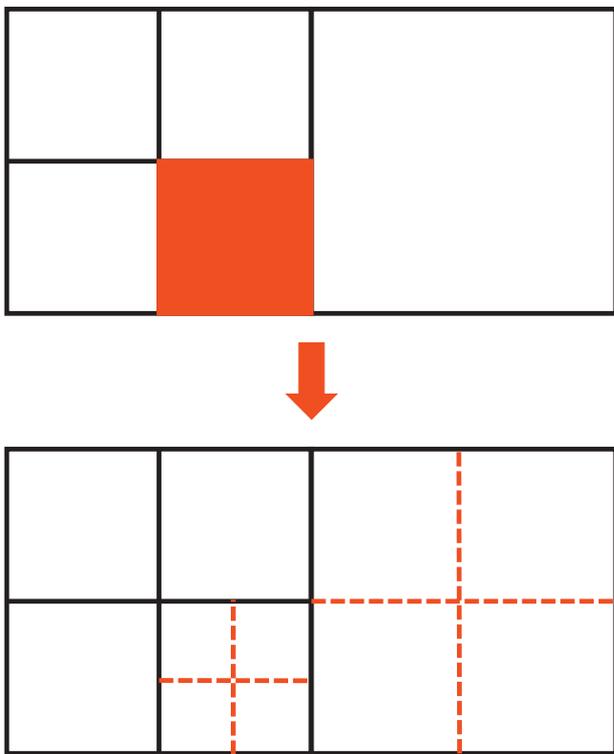


Fig. 4. Applying proper-nesting assumption to maintain grid validity.

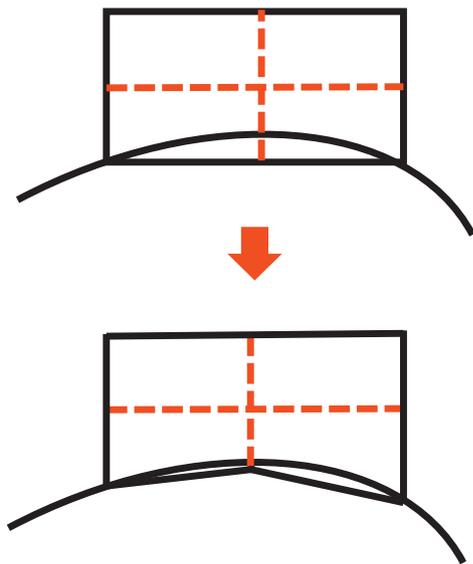


Fig. 5. Curve-fitting newly created mid-node.

mid-face point should be curve-fitted in order to yield accurate results.

The third condition, arising during the moving/deforming grids, is called inter-level boundary synchronization condition [3]. When all the vertex points are allowed to move freely, it could result in non-quadrilateral cells, shown in Fig. 6. This would violate the calculation of the Jacobian and the fourth condition, which is L^2 projection of the solutions between different levels.

The fourth condition is inter-level data transferring condition. During accuracy test in Section 4.1 we found that, two-dimensional interpolation, which is usually a choice of smoother for the solutions, produces less accurate results. Instead, a two-dimensional L^2 projection between fine and coarse levels is employed[25]. The

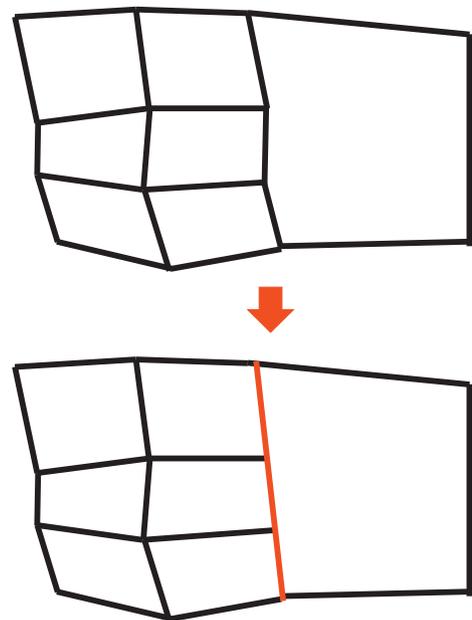


Fig. 6. Realignment of non-quad element occurred during mesh motion.

improvement is substantial, and makes the test case able to retain high-order accuracy. It is a natural extension of the techniques described in Section 3.3. All test cases in Section 4 use two-dimensional L^2 projection.

3.2.2. Compact, contiguous memory layout

The storage strategy of the current data structure is designed to achieve low redundancy and low operation count on a chunk of contiguous memory space, while maintaining direct addressing at the same time. An example of a domain having 4 cells originally is given in Fig. 7 to illustrate the storage strategy. The original cells, i.e. level 0 cells, are stored in an array in Fig. 7(a), which is allocated on a chunk of contiguous memory. The cells are assumed to be numbered counter-clockwise, starting from the lower left one. The cell numbering on level 0 is entirely up to the meshing software/algorithm used, does not affect our memory management.

In Fig. 7(b), if cell 4 on level 0 is flagged by the sensor as the first cell needs to be refined. Its sub-cells, 17–20, are stored at the beginning of the continuous memory space allocated for level 1, instead of the 17th–20th entries of the level 1 array. The size of the contiguous memory space for level i , $i \geq 1$ is predetermined. For cases of shock capturing, 20% of the total cells on level i will usually be sufficient. The location of a newly created sub-cell in the memory space of level i is mapped to a separate sequence array for direct locating.

In Fig. 7(c), if the sensor proceeds to identify cell 1 on level 0 as the next cell to be refined. Its sub-cells, 5–8, simply trails the exiting cells on level 1. The tail indicator, which is set to be the number of employed cells, is increased to 8.

In Fig. 7(d), cell 7 on level 1 needs to be refined. Due to the properly nested condition introduced in Section 3.2.1, cell 2 on level 1 needs to be refined as well. The newly created cell 29–32 on level 2 will be placed in the beginning of the memory space for level 2. Since cell 7 on level 1 is no longer employed, the last employed cell in the memory space of level 1 is copied to its location, therefore the memory space maintains contiguous. After the copying operation, the tail indicator and the sequence array need to be updated accordingly.

The property of the current memory management techniques can be described as “virtually existing everywhere, but selectively

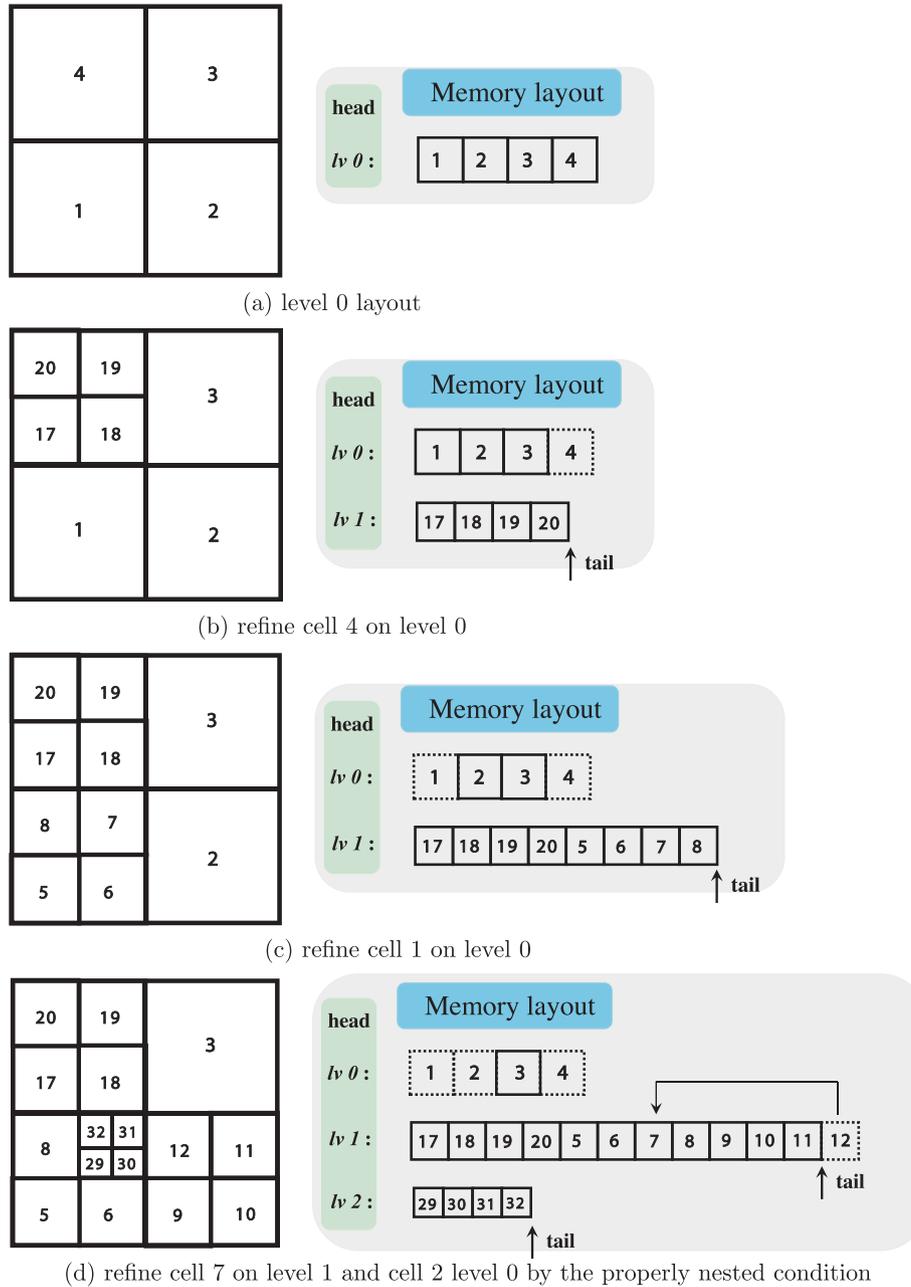


Fig. 7. An illustration of compact and contiguous memory layout during AMR process.

and compactly stored”. Since the connectivities on each level of refinement are precomputed and stored, the hierarchical dependency of a certain sub-cell is also known at the beginning of the calculation due to the direct-addressing nature described in Fig. 2, there is no need to search and insert it to a hierarchical location as in tree-based data structures, hence “virtually existing”. If we take another look at the cell sequence on level 1 in Fig. 7(d), we can get the following observations. The sequence does not necessarily starts from the first sub-cell on level 1. Although the memory space is contiguous, the cells stored in it are not necessarily continuously numbered, hence “compactly stored”.

3.2.3. Robustness and efficiency

The benefits of this AMR method is its simplicity and robustness. In the current data structure, there is a main grid on each LoR, which forms the object (as in the context of object-oriented programming). Regardless the object is the global array (FORTRAN)

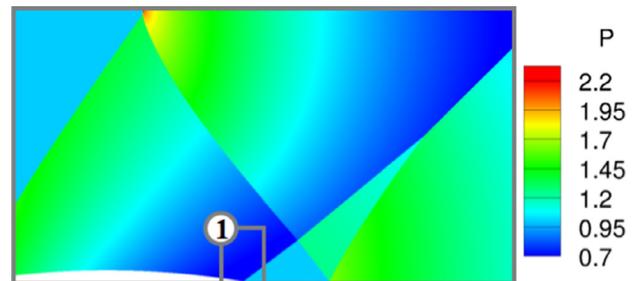


Fig. 8. Pressure contour in the supersonic bump case (Section 4.3) and zoom-in location, where the second shock stems.

or pointer (C/C++) to the computational cells, the non-AMR and AMR simulation share the same set of subroutines, i.e. numeri-

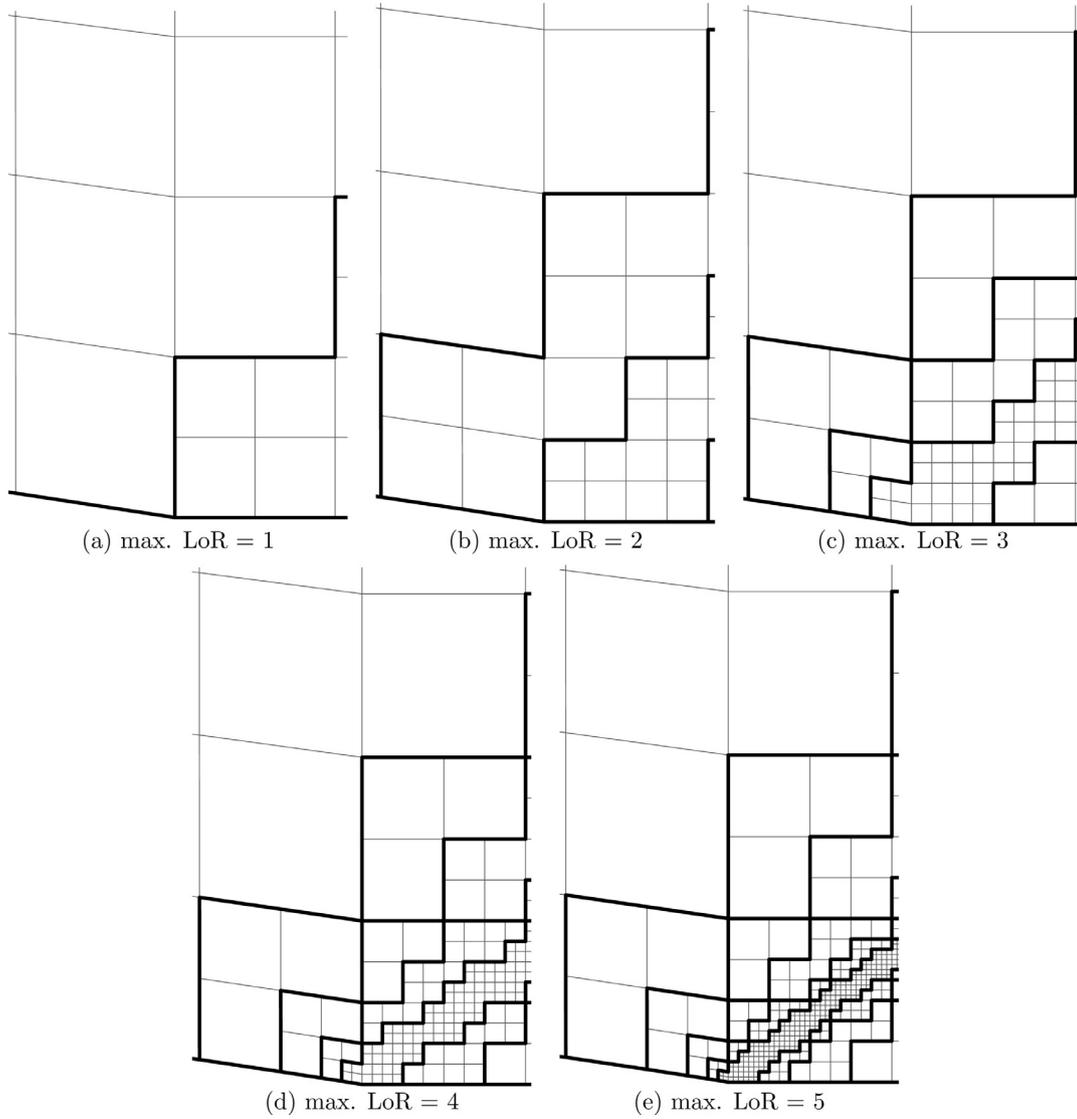


Fig. 9. Hierarchical grid structure in multilevel AMR near the shock emitting from the end of the bump. The inter-level boundary is highlighted with bold lines.

cal flux, time stepping, residual and etc. There is no need for the subroutines to differentiate on which level a certain cell is, which means they behave exactly the same for original cells and refined sub-cells, e.g. the program is capable of starting from either cells on the original level or cells on any other levels of refinement. As pointed out by Bell et al. [9], the robustness of an AMR method is measured by how few changes are required when extending it from two dimensions to three dimensions. Due to its integration with the original code, this AMR method would result in the almost identical code for two dimensional and three dimensional codes, with appropriate modularizations/object-orientations. It naturally incorporates into the moving/deforming mesh strategies and imposes no extra restriction, simply through updating its Jacobian in Eq. 6.

There are concerns that arise in the fast growth of cell ID within the global integer indexing space. The fast growing ID number resulted from the hierarchical relationship in Fig. 2 is crucial to ensure direct addressing. We can use a simple calculation to determine the limit of available levels of refinement in this AMR method. The global integer indexing space, i.e. cell IDs, is bounded by integer representation in the compilers. If using 8-bit integer representation, the total number of cells is limited by the

following condition in order to avoid array index overstepping,

$$\sum_{i=0}^{n_l} 4^i \cdot N_0 \leq 2^{31} - 1, \quad (21)$$

where n_l is the total levels of refinements, N_0 is the number of cells in the original level. For a 2D case with $N_0 = 10,000$, which is usually sufficient to capture the vortical structures in low to moderate Mach number flows, it would take 24 levels of refinement before it runs out of integer space. If more degrees of freedom are needed, which is already a considerably heavy load for a single processor, one can resort to parallelization for optimal speed. Due to its integration with the original code, it will be straightforward to parallelize the AMR data structure.

To quantitatively study the efficiency of using AMR method, We will compare the required wall time of AMR grids and uniformly refined grids, by testing an example case in Section 4.3, a circular bump in a supersonic channel flow on a 20×60 grid. The pressure contour $t = 10.0s$ is shown in Fig. 8, where two shocks stem from the front and back of the circular bump. The highlighted area is the location that we show the effect of multilevel AMR on the original grid. Five sets of AMR-uniform comparison are conducted. The AMR cases have maximum LoR set to 1–5 (the original grid is

Table 1
Comparison of wall time (second) between AMR grids and uniformly refined grids .

	Lv 1	Lv 2	Lv 3	Lv 4	Lv 5
AMR	857	1440	5904	27,896	61,882
Uniform	2439	9386	72,881	675,770	2,700,500
Speed-up	2.85	6.52	12.34	24.23	43.64

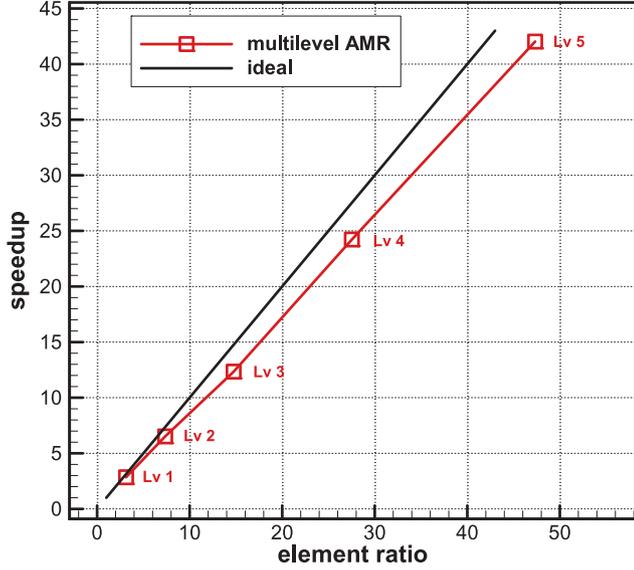


Fig. 10. Comparison of speed-up across multiple maximum LOR.

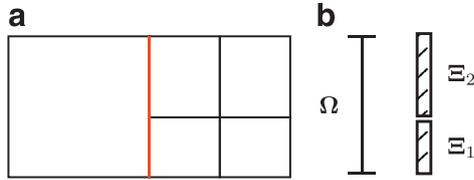


Fig. 11. (a) non-conforming interface emerges from subdomain refinement (b) illustration of contributing subdomain Ω and mortars Ξ .

considered on level 0). The reference group are uniformly refined to achieve the same resolution around the shock discontinuity, i.e. grid resolution from 40×120 to 640×3840 . The wall times are measured by the time elapsed from the start to $t = 10.0s$. A consistent time step for the cases with various levels of refinement is employed, which is the one that is able to stabilize the 5-level AMR case. Refinement criterion is non-zero artificial bulk viscosity, which is evaluated over iterations for $\Delta t = 0.1s$. Detailed analysis will be given in Section 4.3. The wall time and speed-up effect are tabulated in Table 1. Flow region with high gradients such as shock discontinuities demands higher grid density for better resolution as well as stabilization. AMR method can dynamically concentrate computing resources near shock discontinuities, therefore it can achieve the same resolution as a uniformly refined mesh with far less cell count. When using 5-level AMR (6 levels in total), as shown in Table 1, a gain of speed-up can be achieved as high as 43.

We define element ratio as $N_{ref, i}/N_{avg, i}$ on level i , where $N_{avg, i}$ is the average cell count occurred during the simulation. The correlation between speed-up and element ratio across level 1–5 AMR is plotted Fig. 10. Each new level of refined grid will create AMR overheads and non-conforming interfaces. The AMR overheads include time spent on evaluating refining/coarsening criteria, projecting data between cells on different LoR and maintaining compact

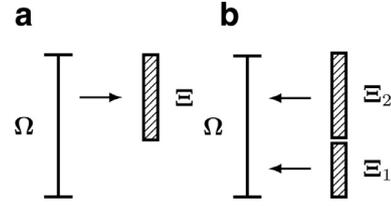


Fig. 12. Communication (a) left subdomain to mortars (b) mortars to left subdomain.

memory layout. It is expected that the speed-up across multilevel AMR will be lower than the ideal speed-up slope as more levels are involved. Nevertheless, as shown in Table 1, the gain in speed-up as the maximum LoR allowed increases always outweighs cost of the resulted AMR overheads and non-conforming interfaces. In fact, the more levels are involved, the more it will benefit the speed-up performance.

3.3. Mortar method

Mavriplis [31] first introduced this method to handle communication between non-conforming interface. Kopriva [26] adapted it for solving compressible flows on fixed, structured grids. Zhang and Liang [48,49] extended it to curved sliding interface. An example of subdomain refinement is given in Fig. 11 to demonstrate the procedure of Mortar method. Instead of interpolating flux points on either side of the interface to ensure a point-to-point communication, a L^2 projection is used to project the flux back and forth between the subdomains and the mortars. It subsequently satisfies global conservation and outflow condition in a hyperbolic system [26]. The mortar is denoted by Ξ , two contribution on each side of the mortar is denoted by L and R . The complete procedure to compute flux on non-conforming interfaces are detailed as below.

The solution variables on Ω can be represented by the following nodal form with a Lagrange polynomial basis:

$$\mathbf{Q}^{\Omega} = \sum_{j=1}^N \mathbf{Q}_j^{\Omega} h_j^{\Omega}(\xi), \quad (22)$$

where $\xi \in [0, 1]$. If maintaining the same solution/flux points configuration as well as polynomial order, solution variables on mortars can be defined as

$$\mathbf{Q}^{\Xi} = \sum_{j=1}^N \mathbf{Q}_j^{\Xi} h_j^{\Xi}(z), \quad (23)$$

where $z \in [0, 1]$, z and ξ are related by $\xi^L = o^L + s^L z$ in which o^L and s^L are the offset and scale of the mortar with respect to left subdomain as shown in Fig. 12(b).

3.3.1. Subdomain \rightarrow mortar projection

An unweighted L_2 projection is used to approximate the extrapolated solution on the mortar flux points $\mathbf{Q}^{\Xi, L}$ from the contributing subdomain interface. We use $\mathbf{Q}^{\Xi, L}$ to denote the discrete values in the vector $[\mathbf{Q}_1^{\Xi, L}, \mathbf{Q}_2^{\Xi, L}, \dots, \mathbf{Q}_N^{\Xi, L}]$, and use $\mathbf{Q}^{\Xi, L}(z)$ to denote the continuous function. The error between $\mathbf{Q}^{\Xi, L}(z)$ and $\mathbf{Q}^{\Omega}(o + sz)$ is required to be orthogonal to the polynomial space

$$\int_0^1 (\mathbf{Q}^{\Xi, L}(z) - \mathbf{Q}^{\Omega}(o + sz)) h_j^{\Xi} dz = 0, \quad j = 1, 2, \dots, N. \quad (24)$$

In matrix form, the approximation on mortar flux points is computed by

$$\mathbf{Q}^{\Xi, L} = \mathbf{P}^{\Omega \rightarrow \Xi} \mathbf{Q}^{\Omega} = \mathbf{M}^{-1} \mathbf{S}^{\Omega \rightarrow \Xi} \mathbf{Q}^{\Omega}. \quad (25)$$

The projection matrix \mathbf{P} is the product of \mathbf{M}^{-1} and \mathbf{S}

$$\mathbf{M}_{i,j} = \int_0^1 h_j^{\Xi} h_i^{\Xi} dz, \quad i, j = 1, 2, \dots, N, \quad (26)$$

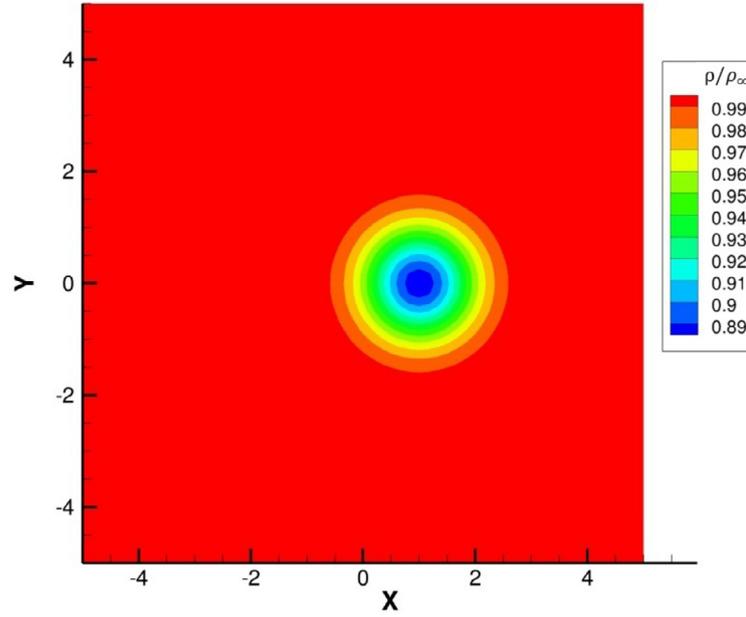


Fig. 13. Euler vortex flow. Vortex moves rightward and horizontally.

Table 2
Order of accuracy, Euler vortex, lv1, AMR.

lv1	N = 3		N = 4	
	L2	Order	L2	Order
400	1.052E-4	-	1.487E-5	-
1600	1.270E-5	3.050	8.576E-7	4.115
3600	4.122E-6	2.949	1.662E-7	4.047

Table 3
Order of accuracy, Euler vortex, lv1, AMR + grid motion.

lv1	N = 3		N = 4	
	L2	Order	L2	Order
400	2.556E-4	-	3.106E-5	-
1600	3.495E-5	2.870	2.091E-6	3.893
3600	1.114E-5	2.819	4.432E-7	3.826

$$\mathbf{S}_{i,j}^{\Omega \rightarrow \Xi} = \int_0^1 h_j^{\Xi} h_i^{\Xi} (o + sz) dz, \quad i, j = 1, 2, \dots, N. \quad (27)$$

Since the right subdomains in Fig. 12(b) have cell boundaries that coincide with the mortars, the projection matrix will become identity matrix. Applying the same principle, we can also get the computational inviscid flux $\tilde{\mathbf{F}}_{inv}^{\Xi,L}$, $\tilde{\mathbf{F}}_{inv}^{\Xi_1,R}$ and $\tilde{\mathbf{F}}_{inv}^{\Xi_2,R}$, which are required to construct the Riemann solver.

3.3.2. Mortar → subdomain projection

Projecting back to the left subdomain involves two contributing mortars. After obtaining the common inviscid flux by a Riemann solver, the same procedure is applied to maintain the best approximation in a least-squares sense

$$\int_0^{o_2} (\tilde{\mathbf{F}}_{inv}^{\Omega}(\xi) - \tilde{\mathbf{F}}_{inv}^{\Xi_1}(z)) h_j^{\Xi} d\xi + \int_{o_2}^1 (\tilde{\mathbf{F}}_{inv}^{\Omega}(\xi) - \tilde{\mathbf{F}}_{inv}^{\Xi_2}(z)) h_j^{\Xi} d\xi = 0, \quad j = 1, 2, \dots, N. \quad (28)$$

The fluxes on the subdomain interfaces and mortars are computed by

$$\tilde{\mathbf{F}}_{inv}^{\Omega} = \sum_{k=1}^2 \mathbf{P}^{\Xi_k \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_k} = \sum_{k=1}^2 s_k \mathbf{M}^{-1} \mathbf{S}^{\Xi_k \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_k}, \quad (29)$$

where s_k is the scaling, \mathbf{M} is identical to Eq. 25, and $\mathbf{S}^{\Xi_k \rightarrow \Omega}$ is the transpose of $\mathbf{S}^{\Omega \rightarrow \Xi_k}$.

3.4. Artificial diffusivity

High-order method gives rise to spurious oscillation near shock discontinuities. The main idea of AD method is to artificially add numerical dissipation to counter the spurious oscillation in the troubled region. In the present formulation, the artificial viscous stress assume the same form as Navier-Stokes viscous stress, which makes it physically consistent on the PDE level, expressed as

$$\mathbf{F}_{ad} = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{yx} + kT_x \end{Bmatrix}, \quad (30)$$

$$\mathbf{G}_{ad} = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xy} + v\tau_{yy} + kT_y \end{Bmatrix}, \quad (31)$$

where the viscous stress tensor is

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) + \beta \frac{\partial u_k}{\partial x_k} \delta_{ij}. \quad (32)$$

The discretization for artificially added viscous flux $\tilde{\mathbf{F}}_{ad}^{vis}(Q, \nabla Q)$ (afterisoparametric mapping, Section 2) is similar to the BR2 scheme by Bassi et al [7]. The procedures are listed in the following steps. We use $\tilde{\mathbf{F}}^v$ to represent the viscous flux for abbreviation.

1. Compute the common value of Q^{com} at the cell interface with a simple averaging $\frac{1}{2}(Q^- + Q^+)$
2. Compute the corrected ∇Q with the similar approach as described in Eq. 19 from the draft paper, $\nabla Q^{c-} = \nabla Q^- + r^-$ and

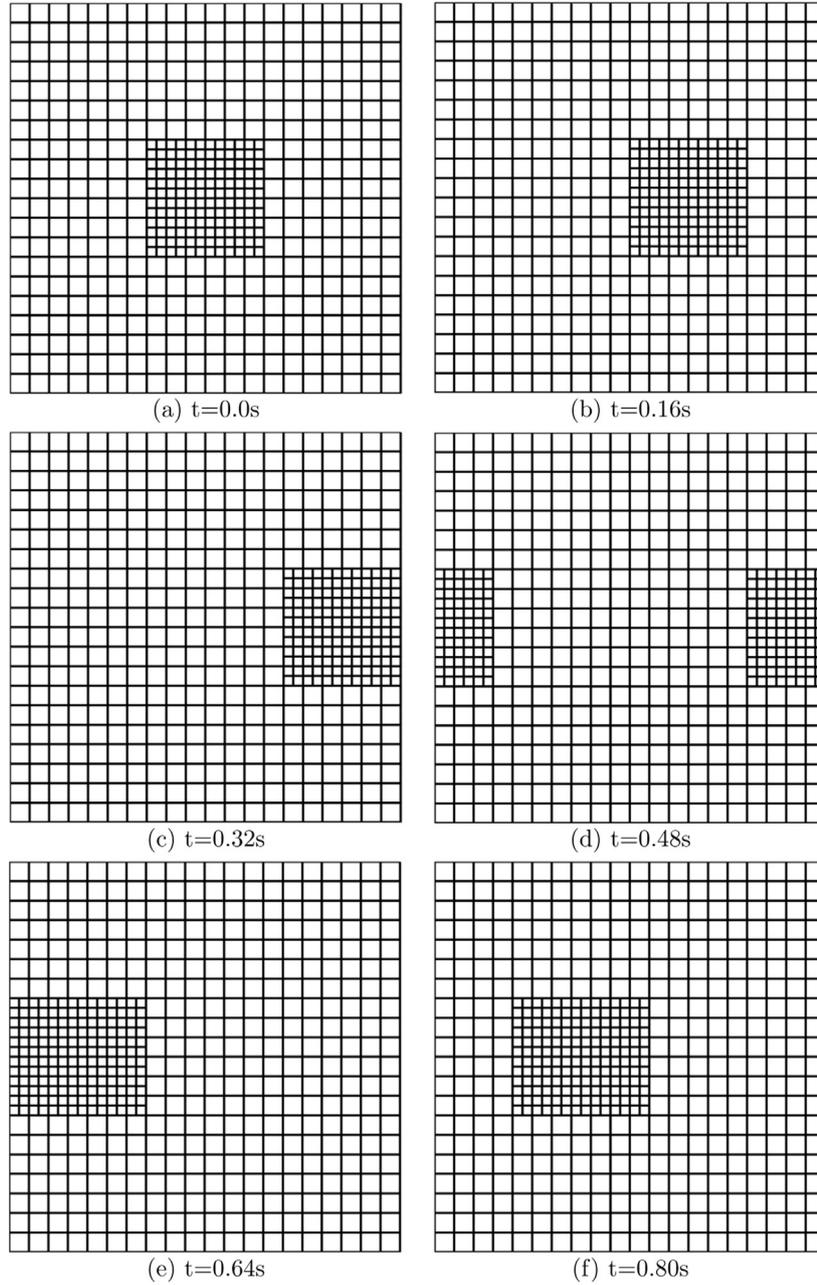


Fig. 14. Euler vortex flow. Vortex moves rightward and horizontally.

$\nabla Q^{c,+} = \nabla Q^+ + r^+$. r^- is determined by Eq. 19, can be expressed as $\left[Q_{j-\frac{1}{2}}^{com} - Q_j(0) \right] g'_{LB}(\xi)$

3. Compute the common value of $\nabla Q^{com} = \frac{1}{2}(\nabla Q^{c,+} + \nabla Q^{c,-})$, and obtain the common value $\tilde{\mathbf{F}}^{v,com}(Q^{com}, \nabla Q^{com})$
4. Reconstruct the viscous flux by

$$\tilde{\mathbf{F}}^{v,c}(\xi) = \tilde{\mathbf{F}}^v(\xi) + \left[\tilde{\mathbf{F}}^{v,com} - \tilde{\mathbf{F}}_j(0) \right] g_{LB}(\xi) + \left[\tilde{\mathbf{F}}^{v,com} - \tilde{\mathbf{F}}_j(1) \right] g_{RB}(\xi). \quad (33)$$

The localized artificial transport coefficients (subscript Δ) are added to the physical transport coefficient (subscript f)

$$\begin{aligned} \mu &= \mu_f + \mu_\Delta, \\ \beta &= \beta_f + \beta_\Delta, \\ k &= k_f + k_\Delta, \end{aligned} \quad (34)$$

in which μ is the shear viscosity, β is the bulk viscosity, and k is the thermal conductivity. And these artificial transport coefficients are defined as follow:

$$\begin{aligned} \mu_\Delta &= C_\mu \rho \left| \sum_{l=1}^2 \sum_{m=1}^2 \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m} \right)_r \frac{\partial r S_{rt}}{\partial \xi_l^r} \right|, \\ \beta_\Delta &= C_\beta \rho \left| \sum_{l=1}^2 \sum_{m=1}^2 \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m} \right)_r \frac{\partial r (\nabla \cdot u)}{\partial \xi_l^r} \right|, \\ k_\Delta &= C_k \frac{\rho c_s}{T} \left| \sum_{l=1}^2 \sum_{m=1}^2 \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m} \right)_r \frac{\partial r e}{\partial \xi_l^r} \right|, \end{aligned} \quad (35)$$

where C_μ , C_β , C_k are user-defined value, Δ_l is the physical grid spacing along a grid line in ξ_l direction, and magnitude of strain rate tensor S_{rt} , dilatation $\nabla \cdot u$ and internal energy e are used

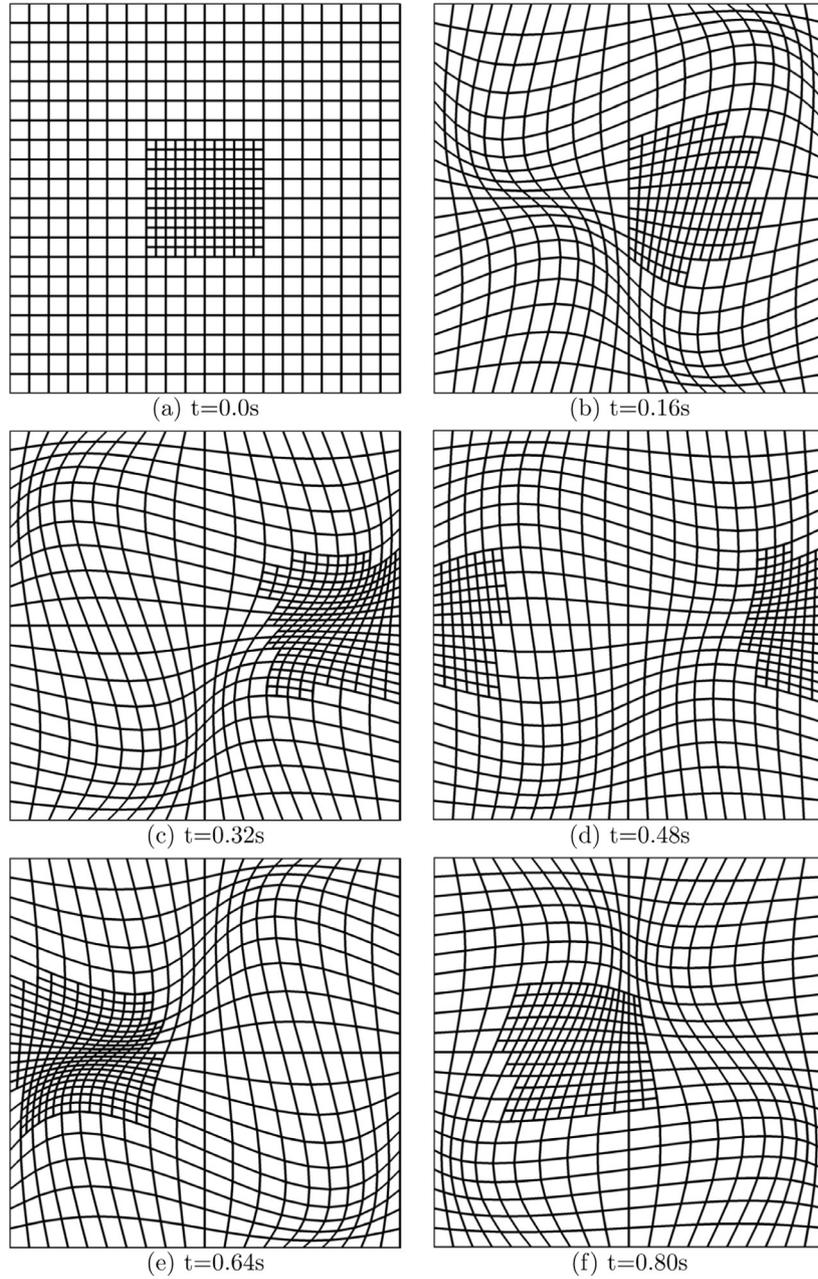


Fig. 15. AMR + prescribed grid motion, tracing the vortex within a cycle of its movement.

as sensors. The overbar denotes a restriction-prolongation filter [36] to smooth the representation of the artificial transport coefficients. The AD terms calculated by above formulation is essentially of order $\mathcal{O}(\Delta^{r+2})$ in smooth region and $\mathcal{O}(\Delta)$ around shocks. In the case of sufficiently high r (4 and above), it eliminates the need for a switch to turn off the artificial bulk viscosity in regions of expansion fan. It is suggested to use $r = 0$ for curvilinear or unstructured grids due to the inaccurate calculation of the laplacian term in the formulation. For $r = 0$, artificial conductivity becomes invalid and the formulation is reduced to

$$\begin{aligned} \mu_{\Delta} &= C_{\mu} \overline{|\Delta^2 S_{rr}|}, \\ \beta_{\Delta} &= C_{\beta} \rho s_{\beta} \overline{|\Delta^2 (\nabla \cdot u)|}, \end{aligned} \quad (36)$$

where s_{β} the switch, following Bhagatwala and Lele [12], it will enact bulk viscosity only when detecting strong dilatation

$$s_{\beta} = 0.5(1 - \tanh(20 \cdot \frac{\Delta}{c} \nabla \cdot u)). \quad (37)$$

4. Numerical tests

In this section, a five-stage fourth order Runge-Kutta method for time stepping [39] is used for all test cases.

4.1. Euler vortex flow

In an Euler vortex problem [15], the flow field, which is essentially a superposition of an isentropic vortex and a background flow, can be prescribed analytically

$$u = U_{\infty} \left\{ \cos \theta - \frac{\epsilon y_r}{r_c} \exp \left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\} \quad (38)$$

$$v = U_{\infty} \left\{ \sin \theta + \frac{\epsilon x_r}{r_c} \exp \left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\} \quad (39)$$

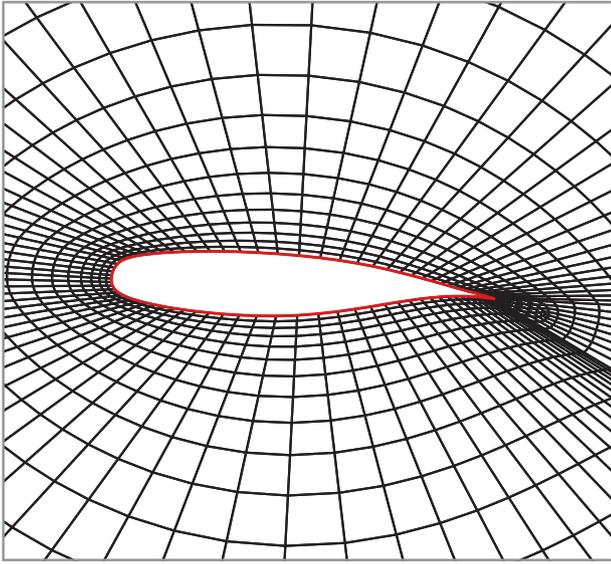


Fig. 16. Grid of NLR7301.

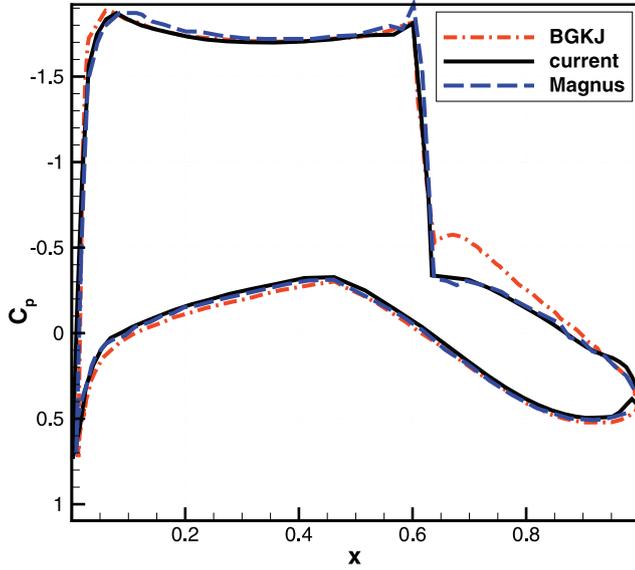


Fig. 17. Verification on shock-capturing with AD method, comparison of pressure coefficient.

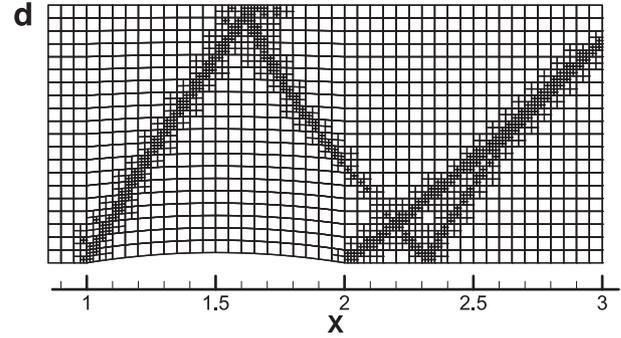
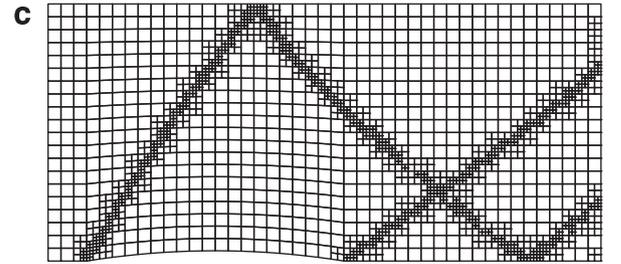
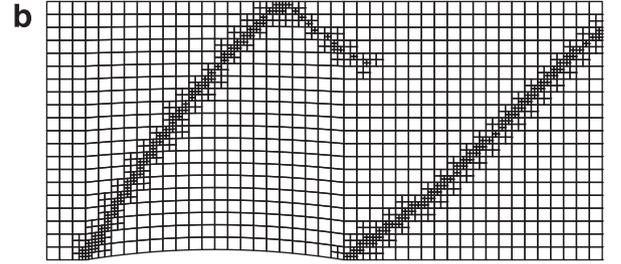
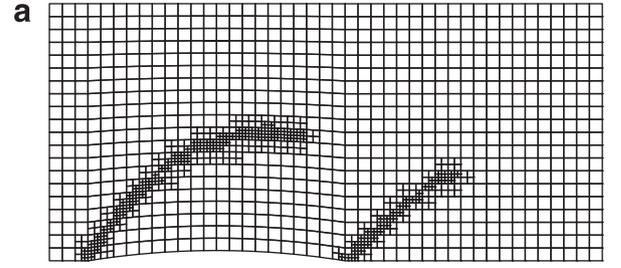


Fig. 18. Grid motion of 2-level refinement.

where $\bar{u} = U_\infty \cos \theta$, $\bar{v} = U_\infty \sin \theta$ are the x and y components of the mean velocity, (x_0, y_0) is the initial position of the vortex. The exact solution of Euler vortex within a square domain ($0 \leq x, y \leq L$) with periodic boundary conditions can be achieved by correcting the relative coordinates as following:

$$x_r = x_r - \lfloor \frac{x_r + x_0}{L} \rfloor \cdot L, \quad (44)$$

$$y_r = y_r - \lfloor \frac{y_r + y_0}{L} \rfloor \cdot L, \quad (45)$$

where $\lfloor x \rfloor$ is the floor operator, it gives the largest integer that is not greater than a real number x , the x_r and y_r on the right hand side are from Eqs. (42) and (43).

This test case is set up as $(U_\infty, \rho_\infty, p_\infty) = (1, 1, 1)$, Mach number of $M_\infty = 0.3$, flow direction $\theta = \arctan(0)$, vortex strength and size $\epsilon = 1$, $r_c = 1$. The domain size is $0 \leq x, y \leq 10$ (i.e. $L = 10$). The vortex is initially placed at the domain center. Periodic bound-

$$\rho = \rho_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2}\right) \right\}^{\frac{1}{\gamma-1}} \quad (40)$$

$$p = p_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2}\right) \right\}^{\frac{\gamma}{\gamma-1}} \quad (41)$$

where U_∞ , ρ_∞ , p_∞ , M_∞ represent mean flow speed, density, pressure and Mach number, respectively. θ is the direction of the mean flow, along which the vortex will be carried, ϵ and r_c are the vortex strength and size. The relative coordinates (x_r, y_r) are defined as

$$x_r = x - x_0 - \bar{u}t, \quad (42)$$

$$y_r = y - y_0 - \bar{v}t, \quad (43)$$

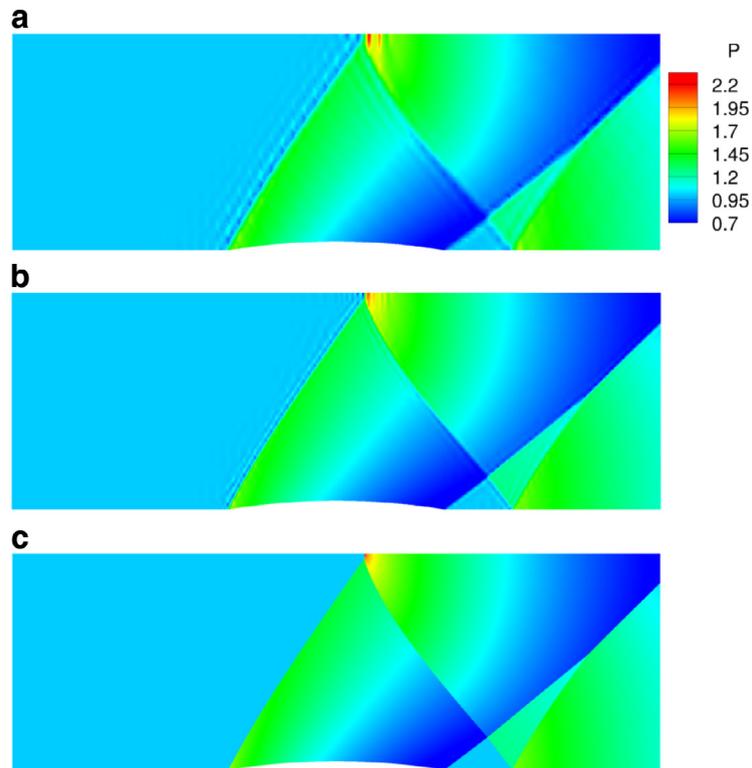


Fig. 19. Pressure contour comparison, (a) no refinement (b) 1-level refinement (c) 2-level refinement.

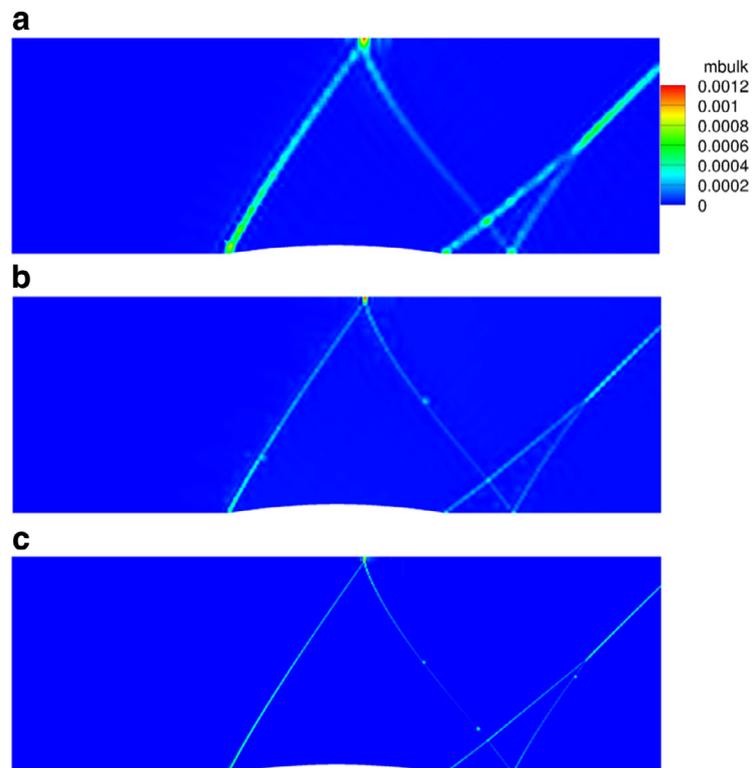


Fig. 20. Bulk viscosity contour comparison, (a) no refinement (b) 1-level refinement (c) 2-level refinement.

ary conditions are applied in both x and y directions. A density contour plot under this setting is shown in Fig. 13.

The purpose of this case is to test that if the $L2$ norm error is reduced as expected when the mesh is refined. In the first set of tests, AMR is tested on stationary grids, employing a sensor of density gradient. Therefore, AMR will refined the region occupied

by the Euler Vortex and follows its motion. The resulted AMR grid is shown in Fig. 14 as the vortex completes a cycle of its periodic motion. The $L2$ norm errors and orders are tabulated in Table 2. It indicates that order of accuracy is maintained in the AMR-enabled solver, with the help of the proposed way of managing AMR cells and the L^2 projection between fine and coarse level of grids.

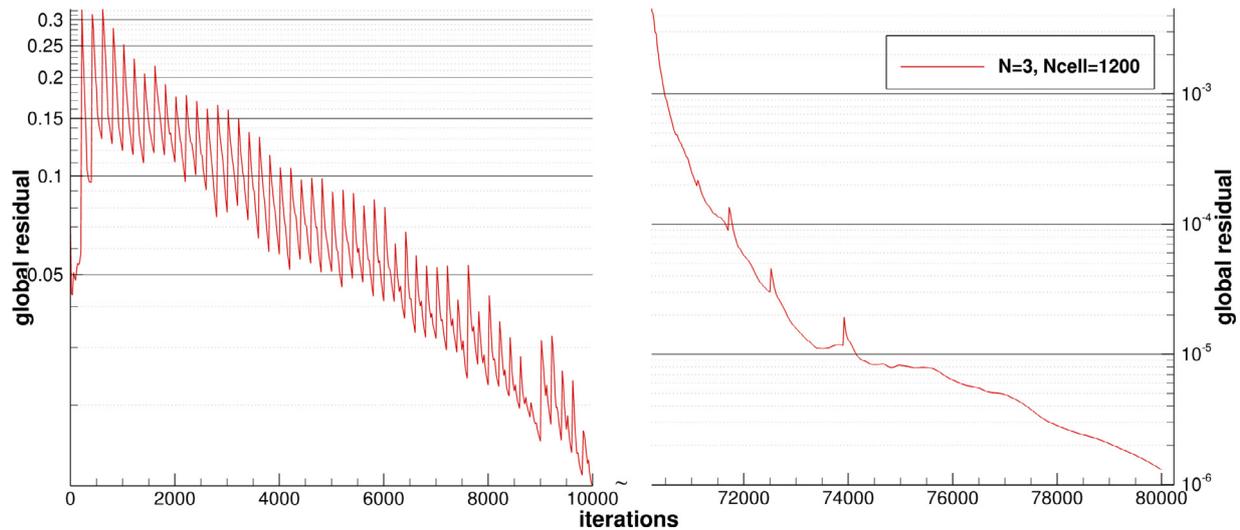


Fig. 21. Convergence history of the supersonic bump.

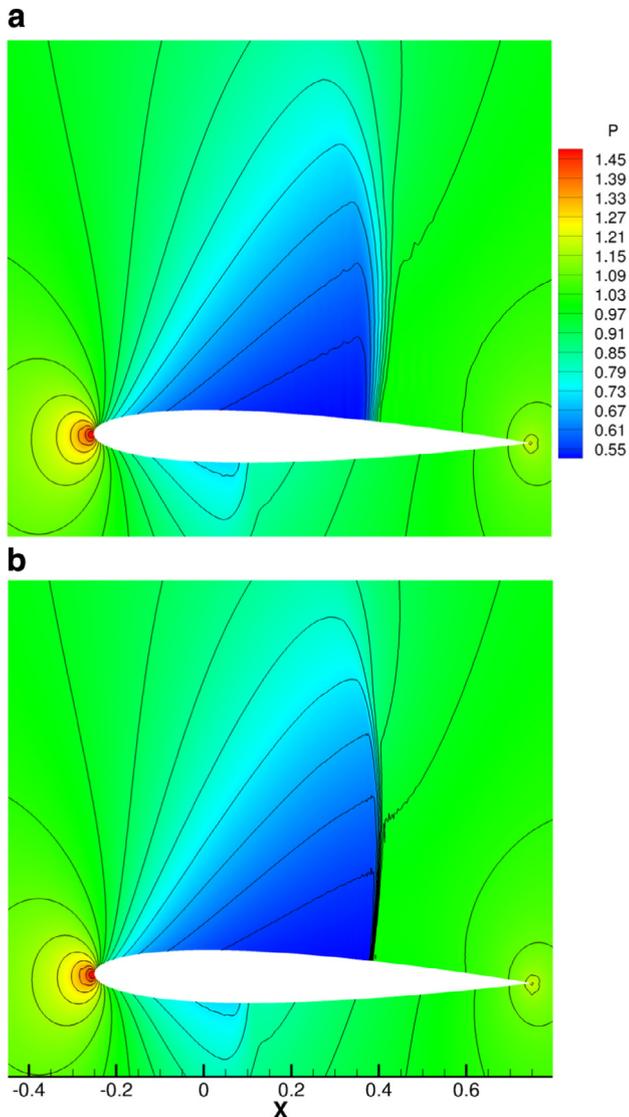


Fig. 22. Comparison pressure contour between (a) no refinement vs. (b) 2-level refinement. NACA0012, $Ma = 0.8$, $\alpha = 1.25^\circ$.

The second set of tests aims to further test the integration of AMR and mortar method, higher-order solver and moving/sliding grids. The grid motion is prescribed analytically

$$x(x_0, y_0, t) = x_0 + \sin(\pi x_0/5) \sin(\pi y_0/5) \sin(4\pi t/t_0) \quad (46)$$

$$y(x_0, y_0, t) = y_0 + \sin(\pi x_0/5) \sin(\pi y_0/5) \sin(8\pi t/t_0) \quad (47)$$

The same gradient based sensor is used to identify the region of refinement. The combined effect is captured in Fig. 15. The results in Table 3 shows that the integrated solver is able to retain high-order accuracy when the grid undergoes complicated alteration.

4.2. Transonic flow over NLR7301

This numerical verification aims to compare the effectiveness of the shock-capturing scheme, AD method, to two sources of established results. The flow field of transonic flow over a two-dimensional supercritical airfoil NLR7301 is calculated to see how well and accurately the shock profile can be captured. The boundary condition is set up as free-stream Mach number $Ma_\infty = 0.7$ and the airfoil is positioned at angle of attack $\alpha = 3^\circ$. The AD solver is set up with $C_\beta = 1.0$ at $N = 3$. The two sources are: Bauer, Garabedian, Korn and Jameson [8] approached this case with a potential solver, utilizing a non-conservative shock-capturing scheme, and Magnus [30] used finite difference method and Lax-Wendroff scheme to obtain the solution to the Euler equations. In our solution, the flow tangency condition is enforced on airfoil surface. A characteristic based far-field boundary condition [21] is used to prevent outgoing waves from being reflected back into the domain, where entropy and tangent velocity are specified as free-stream value at the inlet, while extrapolated at the outlet. A non-reflective boundary condition is even more important in unsteady transonic cases, it can eliminate unnecessary disturbance from the numerical boundary, Fig. 16.

The comparison of airfoil surface pressure coefficient is drawn in Fig. 17. It shows that the AD method leads to good agreement with the potential and Euler solvers. The only difference between the two numerical solvers and the potential solver lies right behind the shock wave. Pressure will increase and Mach number will decrease ahead of the shock as the flow approaches it. According to Rankine-Hugoniot jump conditions, the pressure behind the shock must be balanced by the flow curvature required by the airfoil. This is only possible if the shock wave has infinite curvature at

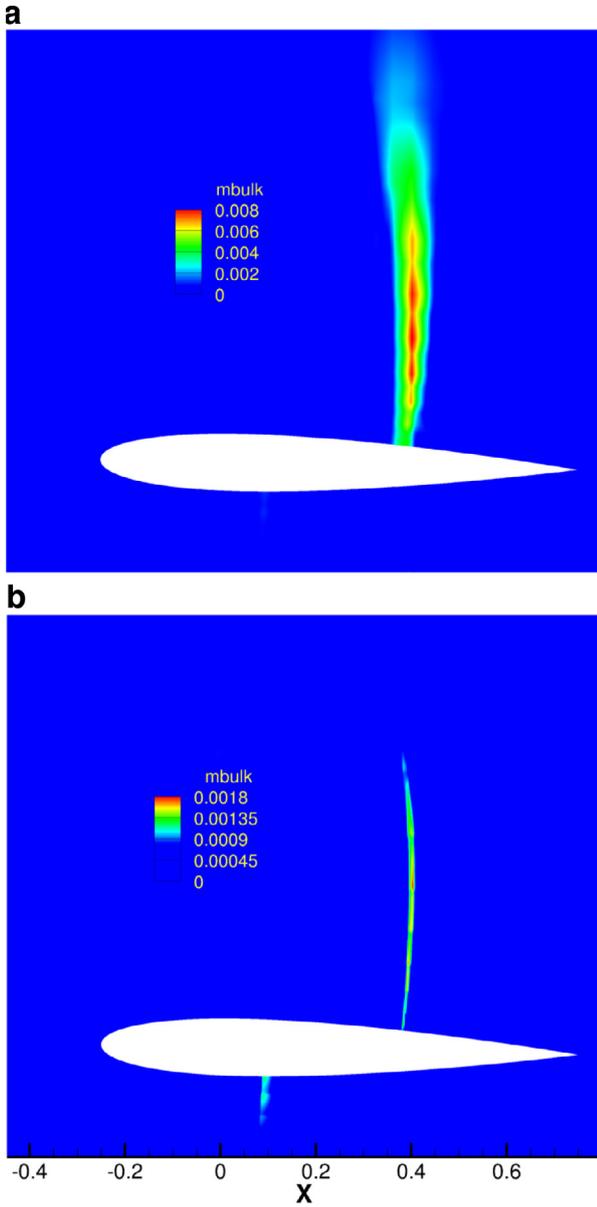


Fig. 23. Comparison bulk viscosity contour between (a) no refinement vs. (b) 2-level refinement. NACA0012, $Ma = 0.8$, $\alpha = 1.25^\circ$.

its foot (normal to the airfoil) [42]. This results in a rapid expansion at the shock foot, and is demonstrated as the so-called Zierep cusp. The discrepancy is due to numerical diffusion arising from the spatial discretization scheme of the Euler equations. Therefore the expansion does not occur as fast as in the potential solver.

The Roe solver [37] is tested for the influence of the choice of Riemann solver on the solution. With identical case settings in the current framework, we find no noticeable difference than the pressure coefficient plot shown in Fig. 17.

4.3. Supersonic flow over a bump

This test case relies on both AMR and AD to stabilize the solution and acquires better resolution of shock discontinuities. The flow field is a 2D channel with a circular bump in the middle, with a dimension of 1 by 3, and radius $r = 1.3$ respectively. The incoming flow has Mach number $Ma = 1.4$. After the flow field reaches stabilized state, it would present two shocks, initiating from two edges from the bump. The shocks will reflect when it travels to the

upper and lower walls. At $N = 3$ and $C_\beta = 0.3$, the pressure distribution is shown in Fig. 19(a). With the help of AD alone, we are able to stabilize the solution and capture the shocks fairly well. Although, the shocks still smear over a number of grids, and the resolution at the intercepting point and reflecting points need improvement (Fig. 9).

Then we join the AD with AMR to further facilitate capturing the shocks. As is shown in Fig. 18, the sensor accurately directs the AMR solver to subdividing cells along the path of the shocks. The results are shown in Fig. 19(b), (c) and Fig. 20(b), (c). The improvements over no AMR (Fig. 19(a) and Fig. 20) is substantial. The shock profile is sharply defined, no smearing can be seen in the solution. The amount of AD is further limited, therefore errors introduced by numerical dissipation is minimized as well. We evaluate the contour plots at the point that the solution has already been stabilized due to its complex shock structure to demonstrate the power of AD and AMR. A Local Mesh Refinement (LMR) may be able to achieve similar effect. But LMR acts on the flow field produced by the main solver which is not given any attention during the calculation, AMR is able to correct the solution all the way along. Besides, if any transient state is of interest, LMR is not as flexible as AMR, since it can only be applied on one specific time instant.

The convergence history is plotted in Fig. 21. The time step size $\Delta t = 1 \times 10^{-4}$, the duration of the simulation is 8×10^4 iterations, the convergence history of the first and last 10,000 steps are shown here. Towards the end, when the shock structure is stabilized, the shock sensor is triggered much less often. In the meanwhile, the residual is driven to a low magnitude (10^{-6}).

4.4. Transonic Flow over Oscillating NACA0012 Airfoil

In the transonic flow regime, the airfoil will experience complex fluid mechanical phenomena. Some of those, such as, flutter and buffeting, are detrimental and can lead to the system failure in the structural perspective. Others like unsteady pressure load, can still not be accurately predicted [14]. As a precursor to a fully-fledged simulation of Shock/Boundary-Layer interaction or transonic buffeting, this test case intends to study the modes of oscillating shocks. The capability of capturing shock on moving/deforming grids in our methods makes it well suited to an oscillating wing induced moving shock.

Tijdeman [41] gave an extensive account of different motions of shocks that will possibly occur on an airfoil that undergoes a pitching motion. He summarized three type of periodic shock oscillations from his pioneering experimental study. Magnus [30] recovered the shock motion numerically from an inviscid, transonic simulation. In this test, we also aim to recover two types of periodic shock oscillation, namely Type A and Type B. Following Tijdeman's classification, Type A shock is present during the entire sinusoidal motion of the airfoil, while Type B shock will be interrupted and partially disappear. A NACA0012 airfoil is placed in a flow field where the far-field Mach number $Ma_\infty = 0.8$. The airfoil has an initial angle of attack $\alpha = 1.25^\circ$, it can pitch at a center locates at $x_c = 0.4Chord$, with the maximum pitching angle $\Delta\alpha_{max} = 1^\circ$, at a frequency $\omega = 10Hz$. The same numerical boundary condition as in Section 4.2 is imposed. The pressure contour in Fig. 22(a) shows the flow field at neutral pitching angle. A strong shock on the upper surface and a weak shock at lower surface are present simultaneously. An additional validation of lift, drag and pressure coefficient of the static case with angle of attack fixed at $AoA = 1.25^\circ$ is documented in Appendix.

To capture the best representation of shocks and their motions, a combination of AD and 2-level AMR is firstly implemented. AD in this case has coefficient $C_\beta = 0.5$, $C_\mu = 0$ at order $N = 4$. The comparison of pressure and bulk viscosity contour at $\alpha = 1.25^\circ$ are

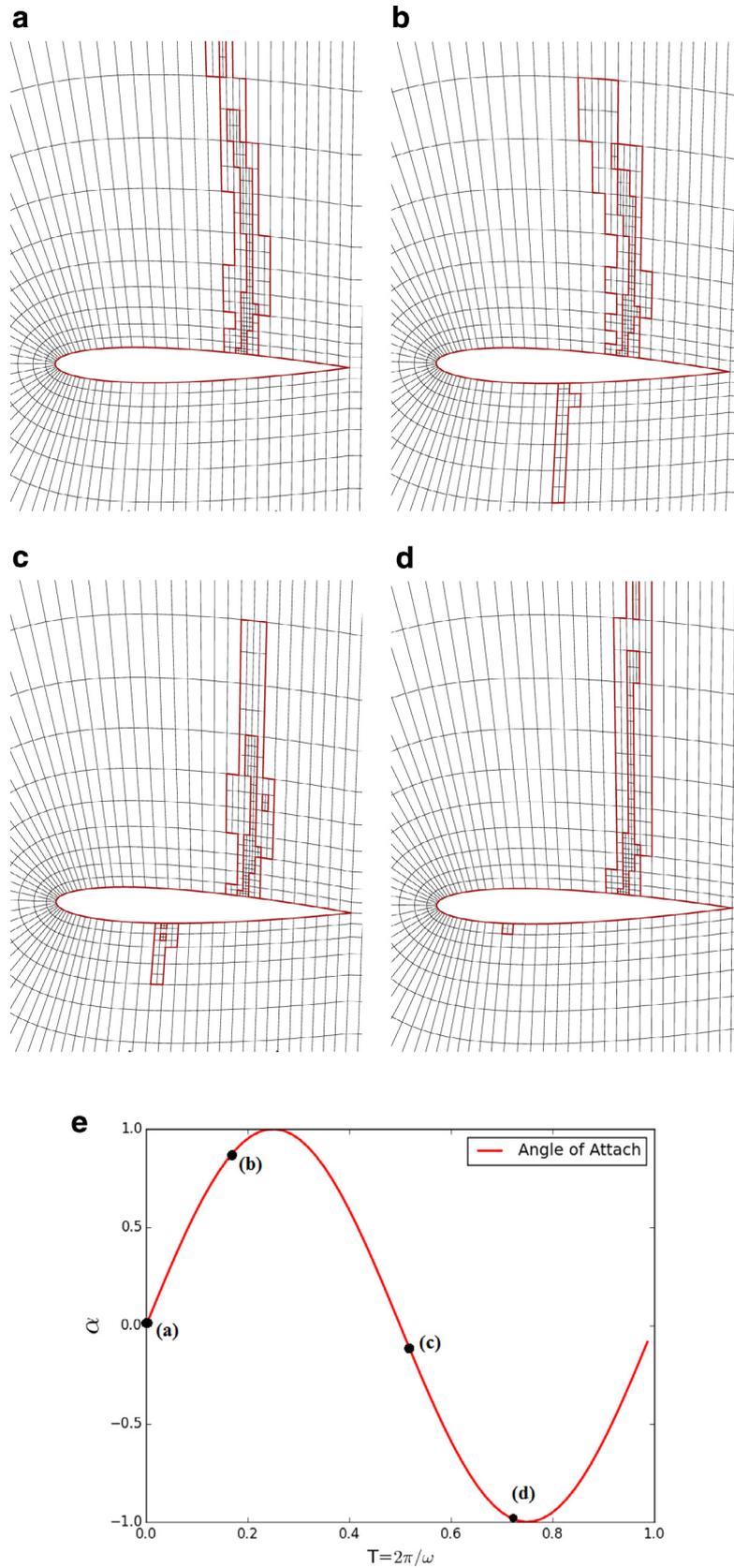


Fig. 24. Grid motion of 2-level refinement within a cycle of the oscillating airfoil. Angle of attack α in sub-figure (a)–(d) is indicated by its corresponding location in the graph of periodic motion in (e).

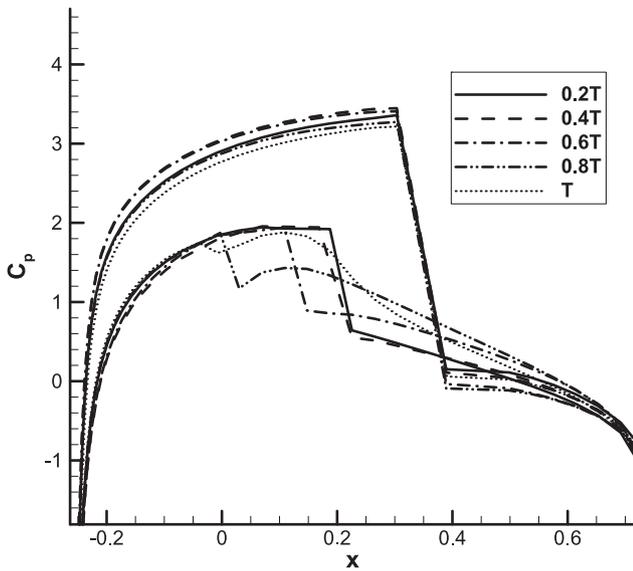


Fig. 25. Pressure coefficient C_p over a period of oscillating. The jumps on the upper and lower surfaces indicates that the shocks are moving periodically.

shown in Fig. 22 and 23, between no refinement and 2-level AMR. In Fig. 22(a), around the shock foot of the upper surface, pressure jump at shock discontinuity spans over a finite fraction of the chord. When 2-level AMR is enabled, the same region in Fig. 22(b). The isopressure lines are essentially clustered together, which will contribute to form a sharp shock front. The forward-inclined shock has a clearer presentation after 2-level AMR. Fig. 23 tells the story that how AMR is able to further assist capturing the shocks. The AD sensor directs AMR to do a 2-level refinement at the shock region. After this 2-level AMR, the AD solver is provided with a finer grid, on which the amount of AD needed is calculated. As shown in Fig. 23, the amount of numerical dissipation added to the flow field is reduced by an order of magnitude.

How AMR responds to the sensor and perform 2-level refinement continuously throughout the periodic motion of the airfoil is laid out in Fig. 24. The corresponding variation of angle of attack $\Delta\alpha$ in sub-figure (a)–(d) is illustrated in sub-figure (e). Due to the accurate detection of shocks, the refined regions closely follow the shocks, and themselves alone can describe the motion of shocks. The shock on the lower surface has a different behaviour than the one on the upper surface. Fig. 24(a)–(d) shows that this shock grows and diminishes as the airfoil oscillates. This shock will disappear and not be detected in part of the cycle.

In fact, its motion is periodic. This is made clear in the plot of pressure coefficient around the airfoil surface as in Fig. 25. C_p is recorded by a interval of $\Delta t = 0.2T$. The jump on the lower surface has a larger displacement than the jump on the upper surface. And it demonstrates evidently a periodic motion.

Time evolution of surface pressure is sampled over the lower surface in Fig. 26. In the middle region, the 3rd, 4th and 5th sampling points, where the shock sweeps back and forth, show periodic yet irregular pressure variation, due to the non-linear effect in unsteady transonic flow. Away from this region, the pressure variation is almost sinusoidal. This is consistent with observations in the experiments [32,42]. The overall aerodynamic loads (lift C_l and moment C_m) are recorded in Fig. 27. The aforementioned experimental observation and measurements also point out that the lift and moment will vary nearly sinusoidally at low to moderate reduced frequency, which is dependent on Mach number Ma and frequency ω . Despite the presence of strong shocks, the oscillating

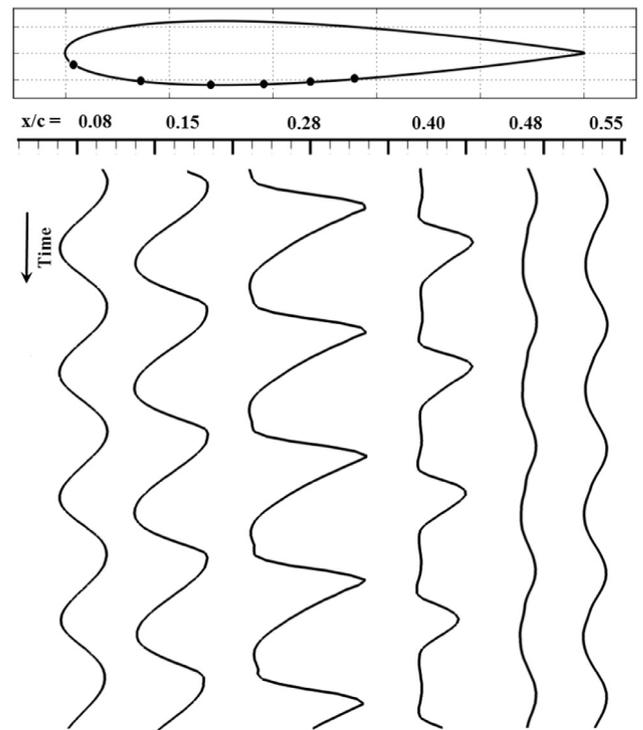


Fig. 26. The evolution of pressure coefficient along the lower surface. Locations that are being monitored are indicated by black dots in NACA0012 profile. C_p present periodic motion as the airfoil pitches sinusoidally.

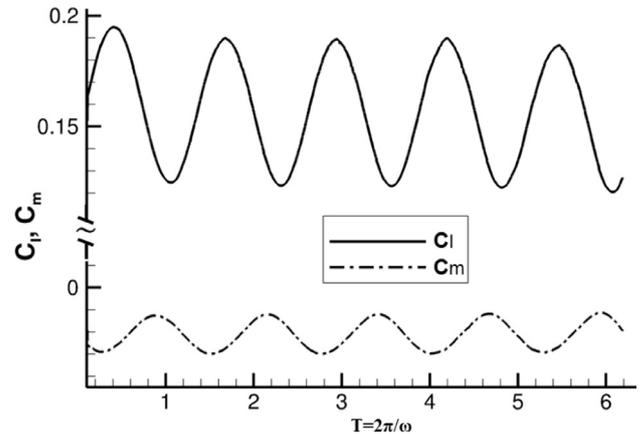


Fig. 27. Lift coefficient C_l and moment coefficient C_m moves sinusoidally despite the presence of shocks.

motion of the airfoil results in linear contribution to the overall aerodynamic load.

5. Concluding remarks

We have presented an integrated framework which combines high-order FR method, AMR and AD methods for the purpose of shock-capturing on dynamic grids. For this purpose, AMR and AD methods reinforce each other. AD detects the shock and directs AMR algorithm to refine the target regions, and AMR subdivides the neighbouring cells, lowers the required amount AD, therefore minimize the error from introducing numerical dissipation. Despite the complex manipulation of the mesh, this method is able to retain high-order accuracy for smooth flows. Its capability of acquiring accurate and sharp solutions of the shock discontinuities is demonstrated through various test cases, including

transonic oscillating airfoil. The robust data structure ensures performing AMR on moving/deforming mesh efficiently. Due to its direct-addressing nature and compact memory layout, the solver can achieve a speed-up as high as 43 compared to using a uniformly refined mesh. These merits make this framework well-suited for complex physical phenomenon, such as Shock/boundary-layer interaction and space weather modelling.

Acknowledgement

Chunlei Liang would like to thank the US Office of Naval Research (ONR) for a Young Investigator Program award administered by Dr. Ki-Han Kim. Jingjing Yang is partially supported by the George Washington University through the faculty start up fund for Prof. Chunlei Liang.

Appendix

The static transonic NACA0012 case shares the same setup as the case in Section 4.4 with angle of attack fixed at $AoA = 1.25^\circ$. The domain has a circular inlet placed $50c$ ahead of the airfoil. Far-field boundary condition is enforced to avoid reflection of sound wave back into the domain. In Fig. A.28, the resulted pressure coefficient is compared with [22].

Table A.4 compares the lift and drag coefficient with two other sources of literature [4,45]. A $N = 2$ simulation is used for comparison, given the three sources all use either finite volume method or low order polynomial base. The current approach of AMR + AD shows good agreement with the reference value from the literature.

Table A1

Lift and drag coefficient of NACA0012 at $Ma = 0.8$ and $AoA = 1.25^\circ$.

	current ($N = 2$)	Jameson [22]	Yano et al. [45]	Balan et al. [4]
C_l	0.35193	0.3688	0.35169	0.35291
C_d	0.022705	0.023600	0.022628	0.022746

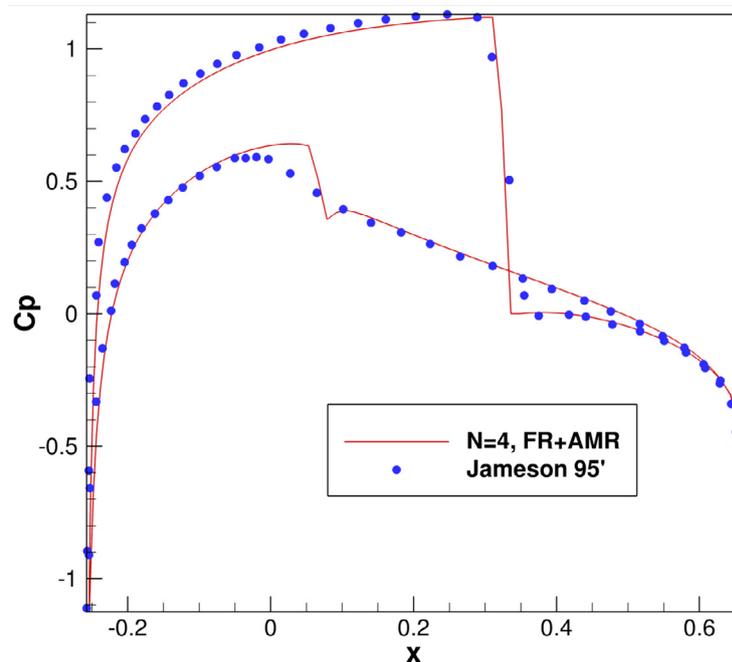


Fig. A.28. Pressure coefficient comparison.

References

- [1] Abe Y, Haga T. Fully-conservative high-order fr scheme on moving and deforming grids. AIAA2015;22nd computational fluid dynamic conference.
- [2] Almgren AS, Bell JB, Collela P, Marthaler T. A cartesian grid projection method for the incompressible Euler equations in complex geometry. SIAM J Sci Comput 1997;18:1289–309.
- [3] Anderson R, Elliot N, Pember R. An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the euler equations. J Comput Phys 2004;199:598–617.
- [4] Balan A, Schütz J, Woopen M, May G. Case c1. 3: flow over the naca 0012 airfoil: subsonic inviscid, transonic inviscid, and subsonic laminar flows. In: First international workshop on high-order CFD methods; 2012.
- [5] Bank RE. Efficient implementation of local mesh refinement algorithm. In: Adaptive computational method for partial differential equation. Philadelphia, PA: SIAM; 1983. p. 74–81.
- [6] Barter GE, Darmofal DL. Shock capturing with high-order, PDE-based artificial viscosity. AIAA Pap 2007.
- [7] Bassi F, Crivellini A, Rebay S, Savini M. Discontinuous galerkin solution of the reynolds-averaged navier-stokes and $k - \omega$ turbulence model equations. Comput Fluids 2005;507–40.
- [8] Bauer F, Garabedian P, Korn D, Jameson A. Supercritical wing sections II. New York: Springer-Verlag; 1975.
- [9] Bell J, Berger M, Saltzman J, Welcome M. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. SIAM J Sci Comput 1994;15(1):127–38.
- [10] Berger MJ, Collela P. Local adaptive mesh refinement for shock hydrodynamics. J Comput Phys 1989:64–84.
- [11] Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. J Comput Phys 1984;53:484–512.
- [12] Bhagatwala A, Lele SK. A modified artificial viscosity approach for compressible turbulence simulations. J Comput Phys 2009;228:4965–9.
- [13] Cook AW, Cabot WH. A high-wavenumber viscosity for high-resolution numerical methods. J Comput Phys 2004;195:594–601.
- [14] Dolling DS. Fifty years of shock-wave/boundary-layer interaction research: what next? AIAA J 2001;39(8):1517–31.
- [15] Erlebacher G, Hussaini MY, Shu CW. Interaction of a shock with a longitudinal shock. J Comput Phys 1997;128:475–88.
- [16] Fidkowski KJ, Roe PL. An entropy adjoint approach to mesh refinement. SIAM J Sci Comput 2010;32:1261–87.
- [17] Haga T, Kawai S. Toward accurate simulation of shockwave-turbulence interaction on unstructured meshes a coupling of high-order FR and LAD schemes. AIAA Pap 2013;AIAA-2013-3065.
- [18] Huynh HT. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. AIAA Pap 2007;AIAA-2007-4079.
- [19] Huynh HT. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods for diffusion. AIAA Pap 2009;AIAA-2009-403.

- [20] Huynh HT, Wang ZJ, Vincent PE. High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids. *Comput Fluids* 2014;98:209–20.
- [21] Jameson A. Solution of Euler equations for two-dimensional transonic flow by a multigrid method. *Appl Math Comput* 1983.
- [22] Jameson A. Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int J Comput Fluid Dyn* 1995;4(3–4):171–218.
- [23] Jameson A, Schmidt W, Turkel E. Numerical simulation of the euler equations by finite volume methods using runge-kutta time stepping schemes. *AIAA Pap* 1981.
- [24] Kawai S, Lele SK. Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes. *J Comput Phys* 2008;227(22):9498–526.
- [25] Kopera MA, Giraldo FX. Analysis of adaptive mesh refinement for imex discontinuous galerkin solutions of the compressible euler equations with application to atmospheric simulations. *J Comput Phys* 2014;275:92–117.
- [26] Kopriva DA. A conservative staggered-grid Chebyshev multidomain method for compressible flows, II. a semi-structured method. *J Comput Phys* 1996;128:475–88.
- [27] Kopriva DA, Koliass JH. A conservative staggered-grid Chebyshev multidomain method for compressible flows. *J Comput Phys* 1996;125:244–61.
- [28] Liang C, Cox C, Plesniak M. A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction. *J Comput Phys* 2013;239:138–46.
- [29] Liang C, Miyaji K, Zhang B. An efficient correction procedure via reconstruction for simulation of viscous flow on moving and deforming domains. *J Comput Phys* 2014;256:55–68.
- [30] Magnus RJ. Some numerical solution of inviscid, unsteady, transonic flows over the NLR7301 airfoil. *Tech. Rep.*. Office of Naval Research; 1978.
- [31] Mavriplis CA. Nonconforming discretization and a posteriori error estimates for adaptive spectral element techniques. *M.I.T.*; 1989.
- [32] McDevitt JB, Okuno AF. Static and dynamic pressure measurements on a NACA 0012 airfoil in the Ames high Reynolds number facility, 2485. *NASA, Scientific and Technical Information Branch*; 1985.
- [33] Minoli A, Kopriva DA. Discontinuous galerkin spectral element approximations on moving meshes. *J Comput Phys* 2011;230:1876–902.
- [34] Persson PO, Peraire J. Sub-cell shock capturing for discontinuous galerkin methods. *AIAA Pap* 2006;AIAA-2006-112.
- [35] Popinet S. Gerris: a tree-based adaptive solver for incompressible Euler equations in complex geometries. *J Sci Comput* 2003;190:572–600.
- [36] Premasathan S, Liang C, Jameson A. Computation of flows with shocks using the spectral difference method with artificial viscosity, II: modified formulation with local mesh refinement. *Comput Fluids* 2014;98:122–33.
- [37] Roe PL. Approximate riemann solvers, parameter vectors and difference schemes. *J Comput Phys* 1981;43:357–72.
- [38] Shi L, Wang ZJ. Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method. *J Comput Phys* 2015;295:261–84.
- [39] Spiteri RJ, Ruuth SJ. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J Numer Anal* 2002;40:469–91.
- [40] Stout QF, DeZeeuw DL, Gombosi TI, Groth CPT, Marshall HG, Powell KG. Adaptive blocks: a high performance data structure. *Tech Report*. University of Michigan; 1997.
- [41] Tijdeman H. Investigations of the transonic flow around oscillating airfoils. *TU Delft, Delft University of Technology*; 1977.
- [42] Tijdeman H, Seebass R. Transonic flow past oscillating airfoils. *Annu Rev Fluid Mech* 1980;12(1):181–222.
- [43] von Neumann J, Richtmeyer RD. A method for the numerical calculation of hydrodynamic shocks. *J Appl Phys* 1950:232–7.
- [44] Wang ZJ, Gao H. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. *J Comput Phys* 2009:228.
- [45] Yano M, Darmofal DL. Case c1. 3: Flow over the naca 0012 airfoil: Subsonic inviscid, transonic inviscid, and subsonic laminar flows. In: *First international workshop on high-order CFD methods*; 2012.
- [46] Ying W. A multilevel adaptive approach for computational cardiology. *Duke University*; 2005.
- [47] Yu ML, Wang ZJ, Hu H. A high-order spectral difference method for unstructured dynamic grids. *Comput Fluids* 2011;48:84–97.
- [48] Zhang B, Liang C. A simple, efficient, high-order accurate sliding-mesh interface approach to the spectral difference method on coupled rotating and stationary domains. *J Comput Phys* 2015;295:147–60.
- [49] Zhang B, Liang C, Yang J. A simple, efficient, high-order accurate approach to spectral difference method on coupled rotating and stationary/deforming domains with sliding interfaces. In: *Proceedings of ASME 34th international conference on Ocean, offshore and Arctic engineering*; 2015. Newfoundland, Canada.