

Accuracy of a 3D adaptive mesh refinement method with analytical velocity fields

*Zhenquan Li¹ and Rajnesh Lal²

¹School of Computing and Mathematics, Charles Sturt University, Thurgoona, NSW 2640, Australia.

²School of Mathematical and Computing Sciences, Fiji National University, Lautoka, Fiji.

*Presenting author and corresponding author: jali@csu.edu.au

Abstract

Meshing plays an important role on the accuracy and convergence of CFD solvers. The accuracy includes quantitative measures such as discretization and truncation errors and qualitative measures such as drawing closed streamline, identifying singular points, asymptotic lines/planes, and (symmetry) axis. The current study builds on previous work by further demonstrating the accuracy of the three-dimensional adaptive mesh refinement method by comparing the accuracy measures between the ones derived from analytical velocity fields and those identified by the refined meshes. The adaptive mesh refinement method presented in this study is proposed based on the law of mass conservation for three-dimensional incompressible or compressible steady fluid flows. The performance of the adaptive mesh refinement method is analysed using three-dimensional analytic velocity fields of four examples. The results provide evidence for the accuracy of the mesh refinement method in identifying the singular points, axes, and asymptote planes of the analytical velocity fields.

Keywords: Adaptive mesh refinement, Computational fluid dynamics (CFD), 3D velocity fields

Introduction

Discrete computational meshes are commonly employed in numerical high-performance computing modelling of physical processes to describe a specific problem's geometry or the general domain. To provide the requisite computation accuracy, discrete meshes are subjected to stringent requirements regarding the level and quality of discretization. Moreover, the simulation of computational fluid dynamics (CFD) problems is often largely reliant on mesh size for convergence and accuracy [1]. In many circumstances, the mesh size required grows to such proportions that the task becomes intractable for the computer resources available. For example, predicting the vortex trajectory of a large-scale flow, such as a tropical cyclone, may necessitate resolving the flow within and around the storm [2]. Furthermore, resolving localized features like vortex centers demands high resolution in areas where the numerical solution varies rapidly [3]. Such challenges can be addressed by adaptively refining the mesh during time-stepping.

There are three common adaptive techniques in CFD to reduce and control numerical error, such as local refinement and coarsening, known as h -refinement, adjusting the local order of discretisation of the numerical method, known as p -refinement, or optimising the distribution of the computational nodes via grid relocating or moving, known as r -refinement [4]. Other quantitative refinement approaches, such as numerical entropy generation schemes and weak

local residuals, include refinement and coarsening indicators [5][6]. To dynamically obtain high accuracy in a domain of interest based on some pre-defined criteria, the h -adaptivity technique is used in adaptive mesh refinement (AMR) [7]. Since the initial work by Berger and Oliger [8] on the application of block-structured AMR to two dimensional (2D) hyperbolic partial differential equations, the AMR approach has been widely extended to several multiscale domains of CFD [9]-[12]. AMR techniques have been effectively employed to minimize computational time and memory requirements for numerous applications in computational fluid dynamics (CFD), computational structural dynamics (CSD), and other fields of computational mechanics [4].

This study uses the AMR method proposed by Li [13][14] to refine a given mesh based on the three-dimensional (3D) velocity fields computed numerically. The AMR method in [13][14] is derived from a theorem in the qualitative theory of differential equations (Theorem 1.14, page 18, Ye [15]) for accurate numerical computation of 2D and 3D velocity fields. The refinement process can be repeated as many times as necessary until the desired level of accuracy or a certain threshold is reached. The mesh refinement technique [13][14] has previously been verified using the accurate locations of singular points, asymptotic lines, and closed streamlines [16]-[18]. Moreover, the accuracy of the 2D AMR method has also been verified against the commonly used CFD benchmark experiments such as the lid-driven cavity flow [19]-[22], the 2D unsteady flow past a square cylinder [23], and the backward-facing step flow [24]. Additionally, the AMR proposed by Li [13][14] has been shown to capture the centre of vortices within the refined cells of once refined meshes and within the twice refined cells after applying the AMR algorithm twice [21][25].

This paper builds on Li's [13] work by further demonstrating the accuracy of the 3D AMR method. Since the error of CFD simulation comes from both the error of numerical methods and the error of meshing, we demonstrate the accuracy of the AMR method using the computational velocity fields without computational errors; that is, the computational velocity fields are calculated by substituting the coordinates of the nodes of a mesh into analytical velocity fields. We show the accuracy of the 3D AMR method by comparing the accuracy measures between the analytical velocity fields and the refined meshes. We provide four examples of 3D AMR using 3D analytic velocity fields from [26]. The four examples presented in this paper provide evidence for the accuracy of the mesh refinement method in identifying the singular points, axes, and asymptote planes of the analytical velocity fields.

The Mass Conservation Conditions for Linear Interpolations of Vector Fields Over Tetrahedral Domains

The 3D AMR method is an extension of the 2D AMR method [14] derived from a theorem in the qualitative theory of differential equations [15].

The continuity equation for incompressible or steady-state fluid is

$$\nabla \cdot \mathbf{V} = 0$$

which is the statement of the law of mass conservation.

Let \mathbf{V}_l be the linear interpolation of the values at the four vertices of tetrahedra in the domain of the velocity field. It follows that for every tetrahedron, the velocity field can be computed as

$$\mathbf{V}_l = \mathbf{A}\mathbf{Y} + \mathbf{B}'$$

where $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$, $\mathbf{B}' = \begin{pmatrix} b_1' \\ b_2' \\ b_3' \end{pmatrix}$, and $\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$ is a matrix of constants, vector of constants, and the vector of spatial variables, respectively.

\mathbf{V}_l is unique if the volume of the tetrahedron is not zero [16]. Substitution of \mathbf{V}_l into \mathbf{V} of the continuity equation gets

$$\nabla \cdot \mathbf{V}_l = \text{trace}(\mathbf{A}) = 0 \quad (1)$$

for incompressible or steady-state fluid. However, the interpolated numerical velocity vector field \mathbf{V}_l generally does not satisfy Eq. (1).

Let f be a scalar function of spatial variables y_1, y_2 , and y_3 . We assume that $f\mathbf{V}_l$ satisfies the continuity equation

$$\nabla \cdot (f\mathbf{V}_l) = 0$$

and then calculate the expressions of f . Solving the above equation for the eight different Jacobian forms of the constant matrix \mathbf{A} results in eight distinct expressions of the function f that are given in Table 1 [13]. In Table 1, $(y_1, y_2, y_3)^T = \mathbf{V}^{-1}\mathbf{X}$ and $(b_1, b_2, b_3)^T = \mathbf{V}^{-1}\mathbf{B}$ where \mathbf{V} satisfies $\mathbf{A}\mathfrak{V} = \mathbf{V}\mathfrak{V}$, and \mathfrak{V} is one of the Jacobian matrices in Table 1. The Jacobian forms of the constant matrix \mathbf{A} and corresponding expressions of f for the eight cases in which the linear interpolations of the vector fields over tetrahedral domains do not hold the law of mass conservation is summarized in Table 1.

For $f \neq \{0, \infty\}$, the vectors \mathbf{V}_l and $f\mathbf{V}_l$ produce same streamlines (for more details we refer the readers to Section 2.2 of [17]).

The conditions (SH) for subdividing a hexahedron are as follows:

- for a hexahedron shown in Fig. 1, subdivide it into five tetrahedra as shown in Fig. 2.
- calculate the Jacobian form of \mathbf{A} in $\mathbf{V}_l = \mathbf{A}\mathbf{Y} + \mathbf{B}'$ for each of the five tetrahedra, respectively.
- if there exist at least one of the five expressions of f corresponding to the particular Jacobian of \mathbf{A} in Table 1 equalling zero or infinity, at some points on the corresponding tetrahedra, subdivision is performed on the hexahedron.

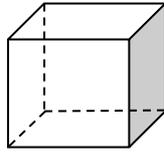


Fig. 1. A hexahedral cell.

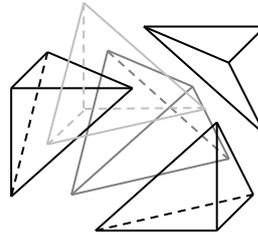


Fig. 2. Tetrahedral subdivision of a hexahedron.

Table 1. Jacobian forms of the constant matrix A and expressions of f for all possible cases of a non-mass conservative linear field.

Case	Jacobian (\mathfrak{J})	f
1	$\begin{pmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{pmatrix}$ $(0 \neq r_1 \neq r_2 \neq r_3 \neq 0)$	$\left(y_1 + \frac{b_1}{r_1}\right)^{-1} \left(y_2 + \frac{b_2}{r_2}\right)^{-1} \left(y_3 + \frac{b_3}{r_3}\right)^{-1}$
2	$\begin{pmatrix} \mu & \lambda & 0 \\ -\lambda & \mu & 0 \\ 0 & 0 & r \end{pmatrix}$ $(r \neq 0, \lambda \neq 0)$	$\left\{ \left(y_1 + \frac{\mu b_1 - \lambda b_2}{\mu^2 + \lambda^2}\right)^2 + \left(y_2 + \frac{\lambda b_1 + \mu b_2}{\mu^2 + \lambda^2}\right)^2 \right\}^{-1} \left(y_3 + \frac{b_3}{r}\right)^{-1}$
3	$\begin{pmatrix} a & \delta & 0 \\ 0 & a & 0 \\ 0 & 0 & r \end{pmatrix}$ $(a \neq 0, r \neq 0)$ $(\delta = 0 \text{ or } 1)$	$\left(y_2 + \frac{b_2}{a}\right)^{-2} \left(y_3 + \frac{b_3}{r}\right)^{-1}$
4	$\begin{pmatrix} \mu & \lambda & 0 \\ -\lambda & \mu & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $(\lambda \neq 0)$	$\left\{ \left(y_1 + \frac{\mu b_1 - \lambda b_2}{\mu^2 + \lambda^2}\right)^2 + \left(y_2 + \frac{\lambda b_1 + \mu b_2}{\mu^2 + \lambda^2}\right)^2 \right\}^{-1}$
5	$\begin{pmatrix} r & \delta & 0 \\ 0 & r & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $(r \neq 0, \delta = 0 \text{ or } 1)$	$\left(y_2 + \frac{b_2}{r}\right)^{-2}$
6	$\begin{pmatrix} r & \delta & 0 \\ 0 & r & \delta \\ 0 & 0 & r \end{pmatrix}$ $(r \neq 0, \delta = 0 \text{ or } 1)$	$\left(y_3 + \frac{b_3}{r}\right)^{-3}$
7	$\begin{pmatrix} r & 0 & 0 \\ 0 & 0 & \delta \\ 0 & 0 & 0 \end{pmatrix}$ $(r \neq 0, \delta = 0 \text{ or } 1)$	$\left(y_1 + \frac{b_1}{r}\right)^{-1}$
8	$\begin{pmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $(0 \neq r_1 \neq r_2 \neq 0)$	$\left(y_1 + \frac{b_1}{r_1}\right)^{-1} \left(y_2 + \frac{b_2}{r_2}\right)^{-1}$

The Adaptive Mesh Refinement Method

In practice, an unstructured mesh is typically employed, with most of the elements being hexahedra. The adaptive refinement approach is applied to every element in a mesh. A hexahedron to which the conditions (SH) applies can be decomposed into five or six

tetrahedra [27][28]. We divide a hexahedron into five tetrahedra in this study. The following algorithm describes how to refine a hexahedral cell in a mesh using the conditions (SH).

The refinement process of a hexahedral cell:

- 1) Subdivide the hexahedron into five tetrahedra and check if V_l satisfies the law of mass conservation on all five tetrahedra. If yes, no refinement for the hexahedron is required. If no, proceed to Step 2.
- 2) Apply the conditions (SH) to all tetrahedra. If the conditions (SH) are not satisfied on all tetrahedra, no subdivision is required. Otherwise, the cell is subdivided into a number of small elements such that the lengths of all sides of the small elements are truly reduced (e.g. half). Fig. 3 is an example that subdivides a hexahedron into eight smaller hexahedra by connecting the mid points of opposite sides on each of the six faces and O_1O_3, O_2O_4, O_5O_6 . The new nodes are $AB, BC, CD, AD, AE, BF, CG, DH, EF, FG, GH, EH$ and $O_1, O_2, O_3, O_4, O_5, O_6, O$ as shown in Fig. 3.

The following is the algorithm of the adaptive mesh refinement method.

Algorithm of adaptive mesh refinement:

1. Let $T = 0$.
2. Calculate the values of a velocity field at nodes of an initial hexahedral mesh.
3. Perform the refinement process one by one for all cells in initial mesh and let $T = T + 1$.
4. Take the smaller hexahedra in the subdivided hexahedra in Fig. 3 as new cells of the initial mesh by replacing the cell in Fig. 1 if a cell is refined in Step 3. Otherwise, keep the cell in Fig. 1 in the initial mesh.
5. Repeat steps 2-5 until a pre-specified threshold number T is reached.

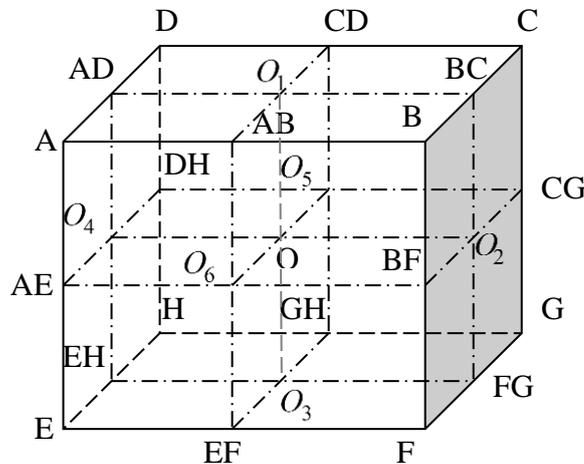


Figure 3. Subdivision of a hexahedron into eight small hexahedra.

In this study, we calculate the values of a velocity field in Step 2 of the algorithm at a point by substituting the coordinates of the nodes into the analytical velocity field. Since the number of refinements can be performed infinite times, we introduce a threshold number T in the algorithm. The choice of T depends on the required accuracy of the mesh, capacity of computers, or computational time.

Results

Four examples of 3D analytical velocity fields are shown here to demonstrate the effectiveness of the adaptive mesh refinement method. In these examples, the 3D analytical velocity fields, adopted from [26], are used to show that the values at the nodes of the refined meshes can present the fields very well by comparing the features shown in the refined meshes with the exact results. As the refinement process can be repeated as many times as necessary to achieve the desired level of accuracy or a certain threshold, we choose T as an integer. A higher threshold number, T , furnishes higher accuracy of numerical results based on the values at the nodes of the refined meshes.

Li [13] considered a different toroidal flow from the one considered in this paper. In [13], streamlines were drawn using the computational velocity fields on a refined mesh obtained by substituting the coordinates of nodes of the refined meshes into the analytical velocity field. For an exact closed streamline of the toroidal flow, a seed point was selected on the exact streamline. Then a streamline was drawn using the computational velocity field on a refined mesh, and the difference between the seed point and the end point (the x coordinate is the same as the x coordinate of the seed point) was compared. The distances between the seed and end points are smaller when the threshold number T is bigger.

In this section, we use four examples to provide evidence for the accuracy of the adaptive mesh refinement method in identifying the other qualitative measures for the accuracy of computational velocity fields.

Example 1: Helical flow

Velocity field: $\mathbf{V} = (-4y, x, 0.5)$.

Fig. 4 shows the initial mesh and three exact streamlines. These lines spiral around the z -axis. The variation of velocity fields at the points close to z -axis is smaller. Hence, more accurate computational velocity fields or a finer cell size are required for drawing accurate streamlines at the points close to the z -axis.

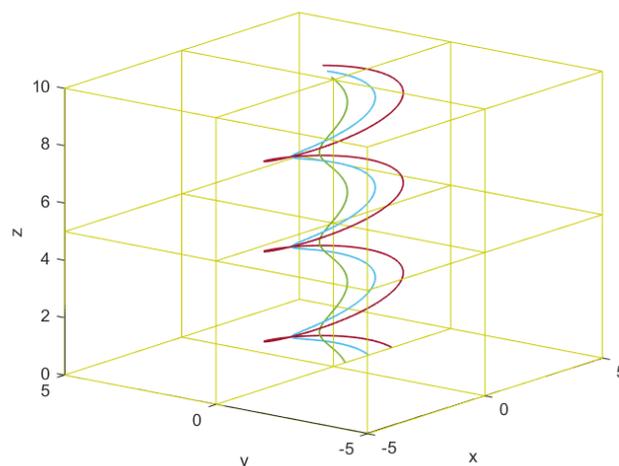


Figure 4. Initial mesh and the exact streamlines of helical flow.

Fig. 5 shows the refined mesh with the three streamlines. The cell sizes are smaller when cells are closer to z-axis. Fig. 6 shows the projection of refined mesh and the three streamlines on xy and yz -planes. The projection on the yz -plane indicates clearly that the cell sizes are getting smaller when cells are closer to the z-axis. The projection on the xz -plane is the same as that on the yz -plane. The projections of the three streamlines on the xy -plane are circles. The projection on the xy -plane again demonstrates that the cell sizes are getting smaller when cells are closer to the z-axis. This example demonstrates that the adaptive mesh refinement method can identify an axis accurately.

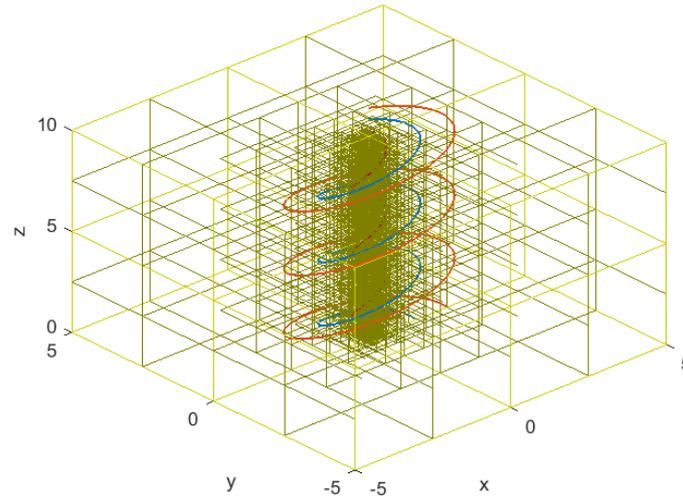


Figure 5. Refined mesh for $T = 7$ and streamlines of helical flow.

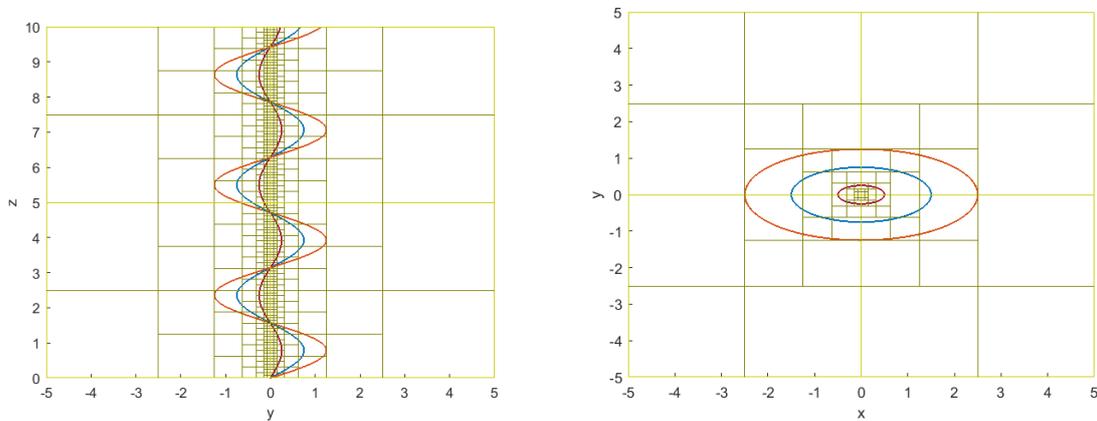


Figure 6. Projections of the graph in Fig. 5 on yz (left) and xy (right) planes.

Example 2: Saddle-spiral flow

Velocity field: $V = (-0.25xz - 5y, -0.25yz + 5x, -0.25z^2)$

Fig. 7 shows the initial mesh and two exact streamlines of the velocity field. The streamline on the top of xy -plane spirals down around the z -axis and gradually approximates the plane but never intersects with the plane. The streamline below xy -plane spirals up around z -axis and approximates the plane closer and closer but never intersects with the plane. Therefore,

xy -plane is asymptotic plane. Since the variations of velocity fields at the points closer to z -axis and xy -plane are smaller, the cell sizes must be smaller for drawing more accurate streamlines. Fig. 8 demonstrate that the finer cell size is closer to the z -axis and xy -plane. This example demonstrates that the adaptive mesh refinement method can identify axes and asymptotic planes. The singular point of this velocity field is the origin, and it is also identified in the refined mesh.

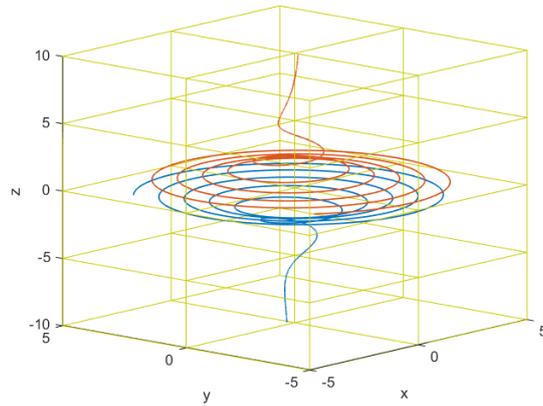


Figure 7. The initial mesh and the exact streamlines of Saddle-spiral flow.

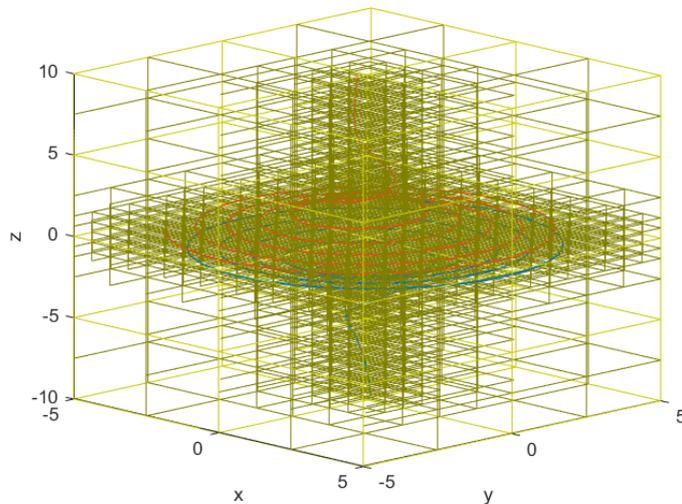


Figure 8. The refined mesh for $T = 5$ and the streamlines of Saddle-spiral flow.

Example 3: Toroidal flow

Velocity field:

$$\mathbf{V} = \left(\frac{-x(z - 4)}{r^2} - \frac{20y(r - 2)}{r}, \frac{-y(z - 4)}{r^2} + \frac{20x(r - 2)}{r}, \frac{r - 2}{r} \right)$$

where $r = \sqrt{x^2 + y^2}$.

Fig. 9 shows an exact streamline. When points are close to the z -axis, the velocity field at some of these points varies considerably, and when r is around 2, the variation of the

z component of velocity field is small. Therefore, we need a mesh with finer cells close to the z -axis and around $r = 2$.

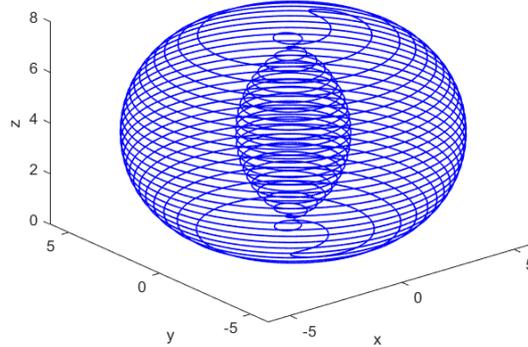


Figure 9. An exact streamline for Toroidal flow.

The refined mesh was generated for $T = 5$. The projections of the refined mesh on the yz - and xy -planes are shown in Fig. 10.

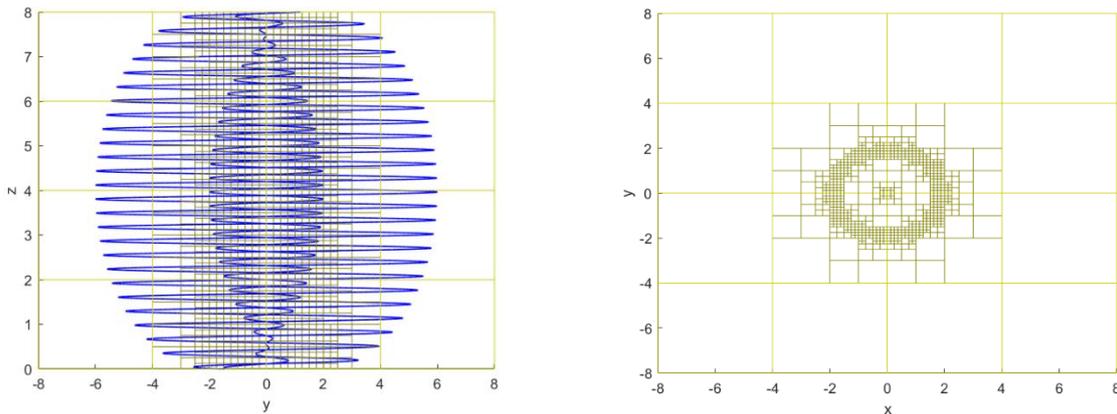


Figure 10. The projection on yz of the refined mesh with the streamline (left) and the projection on xy plane (right).

The projection of the refined mesh on the xy -plane shown on the right in Fig. 10 demonstrates that the cells sizes are getting smaller and smaller when r approaches 2 and cells are closer to the z -axis. The left figure in Fig. 10 shows that the refined cells are in the whole range of z coordinate in the domain. This example demonstrates that the adaptive mesh refinement can identify the areas where the velocity fields vary dramatically in value.

Example 4: Unstable focus-stretching flow

Velocity field: $\mathbf{V} = \left(\frac{-x}{2} - 40y, \frac{-y}{2} + 40x, -z \right)$.

Fig. 11 shows three exact streamlines. All streamlines spiral around the z -axis and towards the xy -plane. Some streamlines are close to the z -axis, and some of them are far away from the z -axis in the beginning and then close to the z -axis when they move close to the xy -plane

but never interact with the plane. Therefore, finer cells are required around the z -axis and xy -plane to compute a more accurate computational velocity field. The accuracy of computational velocity fields means the streamlines drawn using the computational velocity fields are accurate.

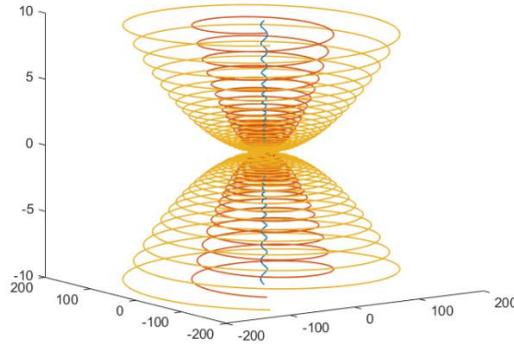


Figure 11. Three exact streamlines.

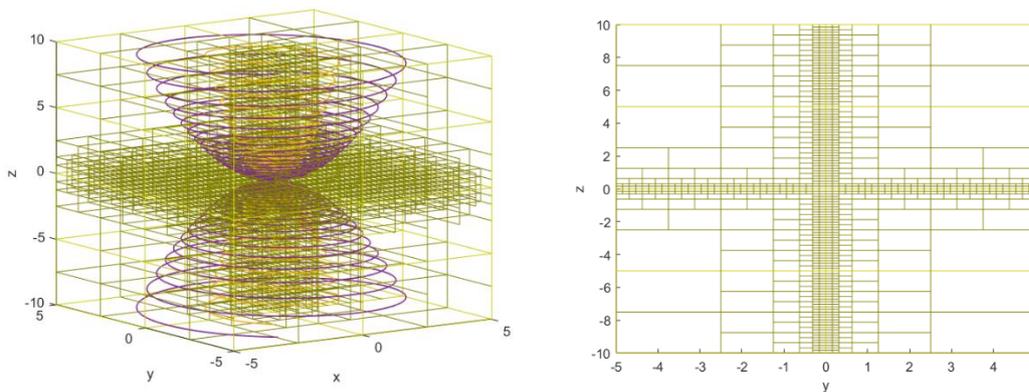


Figure 12. Refined mesh for $T = 7$ with streamlines (left) and the projection on yz plane (right).

The left figure in Fig. 12 shows the refined mesh for $T = 7$. Even though the refined mesh is similar to that of Saddle-spiral flow in Example 2, the two velocity fields are different. The right figure in Fig. 12 clearly shows that the z -axis and xy -plane are identified in the refined mesh. The singular point of this velocity is the origin, and it is identified in the refined mesh.

Discussion

The AMR method is implemented for three-dimensional unstructured meshes with hexahedra elements and extended to multi-level refinement. The results from the four examples are presented, which confirm the accuracy and efficiency of the 3D AMR method. Furthermore, we have assessed the performance of the adaptive mesh refinement method in identifying the accurate location of singular points, axes, asymptotic planes, and other features.

References

- [1] Chakravarthy, S., and Akdag, V. (2015) Importance of accuracy in CFD simulations. *6th BETA CAE International Conference Importance*, 1-9.
- [2] Fulton, S. R. (2001) An adaptive multigrid barotropic tropical cyclone track model. *Monthly weather review* **129(1)**, 138-151.

- [3] Amaziane, B., Bourgeois, M., and El Fatini, M. (2014) Adaptive mesh refinement for a finite volume method for flow and transport of radionuclides in heterogeneous porous media. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles* **69(4)**, 687-699.
- [4] Löhner, R. (2008) Applied computational fluid dynamics techniques: an introduction based on finite element methods. John Wiley & Sons.
- [5] Mungkasi, S., and Roberts, S. G. (2012) Behaviour of the numerical entropy production of the one-and-a-half-dimensional shallow water equations. *ANZIAM journal* **54**, C18-C33.
- [6] Mungkasi, S., Li, Z., and Roberts, S. G. (2014) Weak local residuals as smoothness indicators for the shallow water equations. *Applied Mathematics Letters* **30**, 51-55.
- [7] Rettenmaier, D., Deising, D., Ouedraogo, Y., Gjonaj, E., De Gerssem, H., Bothe, D., Tropea, C. and Marschall, H. (2019) Load balanced 2D and 3D adaptive mesh refinement in OpenFOAM. *SoftwareX*, **10**, p.100317.
- [8] Berger, M. J., and Olinger, J. (1984) Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics* **53(3)**, 484-512.
- [9] Bell, J., Berger, M., Saltzman, J., and Welcome, M. (1994) Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM Journal on Scientific Computing* **15(1)**, 127-138.
- [10] Friedel, H., Grauer, R., and Marliani, C. (1997) Adaptive mesh refinement for singular current sheets in incompressible magnetohydrodynamic flows. *Journal of Computational Physics* **134(1)**, 190-198.
- [11] Rendleman, C. A., Beckner, V. E., Lijewski, M., Crutchfield, W., and Bell, J. B. (2000) Parallelization of structured, hierarchical adaptive mesh refinement algorithms. *Computing and Visualization in Science* **3(3)**, 147-157.
- [12] Berger, M. J., & Leveque, R. J. (1998) Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis* **35(6)**, 2298-2316.
- [13] Li, Z. (2007) An adaptive three-dimensional mesh refinement method based on the law of mass conservation. *Journal of Flow Visualization and Image Processing* **14(4)**, 375-395.
- [14] Li, Z. (2008) An adaptive two-dimensional mesh refinement method based on the law of mass conservation. *Journal of Flow Visualization and Image Processing* **15(1)**, 17-33.
- [15] Ye, Y. (1986) *Theory of Limit Cycles*, American Mathematical Society, Providence, Rhode Island.
- [16] Li, Z. (2002) A mass conservative streamline tracking method for two-dimensional CFD velocity fields. *Journal of Flow Visualization and Image Processing* **9**, 75-87.
- [17] Li, Z. (2006a) An adaptive streamline tracking method for two-dimensional CFD velocity fields based on the law of mass conservation. *Journal of Flow Visualization and Image Processing* **13**, 1-14.
- [18] Li, Z. (2006b) An adaptive streamline tracking method for three-dimensional CFD velocity fields based on the law of mass conservation. *Journal of Flow Visualization and Image Processing* **13**, 359-376.
- [19] Li, Z. (2014) Accuracy analysis of a mesh refinement method using benchmarks of 2-D lid-driven cavity flows and finer meshes. *Journal of Mathematical Chemistry* **52**, 1156-1170.
- [20] Lal, R. and Li, Z. (2015) Sensitivity analysis of a mesh refinement method using the numerical solutions of 2-D steady incompressible driven cavity ow. *Journal of Mathematical Chemistry* **53**, 844-867.
- [21] Li, Z. and Wood, R. (2017) Accuracy verification of a 2D adaptive mesh refinement method for incompressible or steady flow. *Journal of Computational and Applied Mathematics* **318**, 259-265.
- [22] Li, Z. (2017a) Computational complexity of the algorithm for a 2D adaptive mesh refinement method using lid-driven cavity flows. *Computational Thermal Sciences* **9**, 395-403.
- [23] Li, Z. (2017b) Analysis of 2D unsteady ow past a square cylinder at low Reynolds numbers with CFD and a mesh refinement method. *WSEAS Transactions on Fluid Mechanics* **12**, 150-157.
- [24] Li, Z. and Li, M. (2021) Accuracy verification of a 2D adaptive mesh refinement method using backward-facing step ow of low Reynolds numbers. *International Journal of Computational Methods* **18**, 204-1012.
- [25] Li, Z. and Wood, R. (2015) Accuracy analysis of an adaptive mesh refinement method using benchmarks of 2-D steady incompressible lid-driven cavity flows and coarser meshes. *Journal of computational and applied mathematics* **275**, 262-271.
- [26] Reztsov, A. V., and Mallinson, G. D. (1998) Dual stream functions for 3D swirling flows. In *Proc. of 13th Australian Fluid Mech. Conference. Melbourne* 179-182.
- [27] Knight, D., and Mallinson, G. (1996) Visualizing unstructured flow data using dual stream functions. *IEEE Transactions on Visualization and Computer Graphics* **2(4)**, 355-363.
- [28] Nielson, G. M. (1994) Tools for Triangulations and Tetrahedrizations. In *Scientific Visualization* 429-525.
- [29] Reyn, J. W. (1964) Classification and description of the singular points of a system of three linear differential equations. *Zeitschrift für angewandte Mathematik und Physik ZAMP* **15(5)**, 540-557.