# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 題目(和文) | |
| Title(English) | A moving mesh framework based on multi-moment finite volume method and its application to fluid-structure interactions |
| 著者(和文) | JIN PENG |
| Author(English) | Peng Jin |
| 出典(和文) | 学位:博士(理学),<br>学位授与機関:東京工業大学,<br>報告番号:甲第11248号,<br>授与年月日:2019年9月20日,<br>学位の種別:課程博士,<br>審査員:肖 鋒,奥野 喜裕,青木 尊之,末包 哲也,長﨑 孝夫 |
| Citation(English) | Degree:Doctor (Science),<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:甲第11248号,<br>Conferred date:2019/9/20,<br>Degree Type:Course doctor,<br>Examiner:,,,, |
| 学位種別(和文) | 博士論文 |
| Type(English) | Doctoral Thesis |

TOKYO INSTITUTE OF TECHNOLOGY
DOCTORAL THESIS

# A moving mesh framework based on multi-moment finite volume method and its application to fluid-structure interactions

by

Peng Jin

Supervisor: Dr. Feng Xiao

*A Dissertation Submitted to Tokyo Institute of Technology in Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy*

August 2019

# *Abstract*

A wide spectrum of engineering applications are highly related to multi-material fluid flows or flows with sharp interfaces, including contact discontinuities in compressible flows, moving boundaries of fluid-structure interaction (FSI), free surfaces in two-phase flow, etc. However, accurately simulating these flows is very difficult due to the moving interfaces where a variety of boundary conditions or jump conditions may be enforced. In this work, we want to track the moving interfaces as far as possible by using moving grids in order to ensure the accurate solutions at interfaces while leaving extremely flexible interfaces, e.g. free surfaces in two-phase flows, to be captured.

However, development of numerical model for tracking interface is rather challenging since it is always associated with moving unstructured grids which results in the difficulties for achieving adequate solution accuracy, superior numerical stability and robustness. In this thesis, we proposed a moving mesh framework for fluid flows based on multi-moment finite volume method. The characteristics of this framework are presented as follows:

- The motion of fluid flows is described in Arbitrary Lagrangian Eulerian (ALE) viewpoint.
- Both Volume Integrated Average (VIA) and Point Value (PV) are treated as computational variables and used for high order polynomial reconstruction.
- Multi-moments, namely, VIA and PV, are solved separately on arbitrarily moving mesh.

Based on above framework, we developed numerical models for both compressible flows and incompressible flows on moving grids. For applications, we also constructed a practical and reliable numerical solver for Fluid-Structure Interaction (FSI) problems. The details and our major efforts are divided into following four parts:

1. The core idea of the moving mesh framework is interpreted in Chapter 3 for solving compressible Euler equations. Governing equations are firstly cast into ALE integral form and differential form for updating VIA and PV respectively. The VIAs are computed by a finite volume method (FVM) that ensures the rigorous numerical conservativeness, while the PVs are defined at cell vertices and updated efficiently by a point-wise finite difference formulation. By employing

both VIA and PV as computational variables, it becomes convenient for generating quadratic or high order polynomials within a compact stencil. An uniformly third-order accuracy is achieved on arbitrarily moving grids for 1D and 2D smooth solutions and geometrical conservation law is well satisfied. Moreover, numerical tests reveal that it has superior capability for resolving moving discontinuities in compressible flows in a Lagrangian way.

2. In Chapter 4, we extended the numerical model to viscous compressible flows and solved a simple one-way coupling FSI problem where the fluid state is affected by a prescribed varying boundary. Due to the moving boundary of fluid domain, we implemented the radial basis function (RBF) interpolation to transfer the movement of boundary to internal mesh points, which is of great significance for putting our moving mesh numerical model to practical use. In addition to several basic benchmark tests with high Mach number, we also investigate the phenomenon of flows past over an oscillating cylinder, which verifies the feasibility of present model for FSI application.

3. In Chapter 5, multi-moment finite volume method for incompressible flows on unstructured moving grids was developed, where incompressible Navier-Stokes equations are solved by the pressure projection procedure. We then built a fluid-structure interaction solver by integrating the fluid model with an elastically mounted rigid-body. Two coupling schemes, i.e. an explicit weak coupling scheme and a semi-implicit strong coupling scheme, have been devised to formulate the interactions between fluid and moving solid with a wide range of mass ratios. By employing PVs at cell vertices as additional variables updated at each time step, the present multi-moment finite volume method shows great advantage when applied to the ALE framework for interactions among multiple materials, where the solution points can always coincide with the interfaces between different materials.

4. In Chapter 6, we presented a preliminary numerical model for moving body in free surface flow. In view of the flexibility of free surface of two-phase flow as well as breakup and coalescence phenomena, the gas and liquid are described by one-fluid model where the free surface is captured by an algebraic interface capturing method, THINC-QQ scheme. Governing equations for two-phase flow are solved on moving grids by multi-moment finite volume method and coupled with the movement of solid body. Numerical tests, such as wave generation by a moving paddle, water exit of a circular cylinder and free falling of a wedge, show that the present solver is capable for simulating fluid flows with three-phase material interaction (i.e. gas, liquid and solid).

As mentioned above, the proposed numerical model has been verified by a wide range of benchmark tests and analyzed from many aspects including numerical error, convergence rate, conservativeness, computational cost, as well as reliability for practical applications. Numerical results demonstrate the appealing performance and feasibility of present framework for FSI application.

# *Acknowledgements*

I would like to express my gratitude to all those who helped me during the finish of this thesis.

First and foremost, I want to give my sincere thanks to my supervisor, Prof. Xiao, who provided me the opportunity and good environment for fulfilling my Phd course. His instructive advice and continuous encouragement helped me get through many difficulties encountered in the research. His passion and serious attitude for CFD and truth significantly affect me, which will be an invaluable treasure in the rest of my life. Secondly, I deeply thank Prof. Aoki, Prof. Okuno, Prof. Suekane and Prof. Nagasaki for becoming my thesis defense committee. Their constructive suggestions and comments helped me to improve the thesis.

I would also like to express gratitude to my labmates, Xie, Sun, Che, Deng, Ikebata, Abe, Inaba, Nakayama, Wang, Jin Jonghoon, Kondo, Tsubone, Saito, Hashiguchi, Siengdy, Andres, Zhang, Qi, Qian, Ronit, Cheng. They not only gave me kind supports in academic study, but also enriched my overseas life. Specially, I want to thank Dr. Xie, who guided me to study CFD since I entered the lab. and provided the pioneering works of this research. When I got into troubles, he was patient to help and can always come up with practical solutions. Also, I want to express my thanks to Dr. Sun, who have ever recommended a valuable reference which helped me to find a feasible way to my research objective. Besides, I would like to thank Dr. Deng, who helped me a lot in the field of compressible flows.

Last but not least, I thank my beloved family from the bottom of my heart for their loving considerations and great confidence on me through these years. May health and happiness always around you.

Peng Jin

August 2019

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **ADER** | **A**rbitrary high-or**DER** accurate |
| **ALE** | **A**rbitrary **L**agrangian **E**ulerian |
| **CIP-CSL3** | **C**onstraint **I**nterpolation **P**rofile-**C**onservative **S**emi-**L**agrangian scheme with **3**rd-order polynomial function |
| **CFD** | **C**omputational **F**luid **D**ynamics |
| **CSF** | **C**ontinuum **S**urface **F**orce |
| **CSS** | **C**onventional **S**erial **S**taggered |
| **DG** | **D**iscontinuous **G**alerkin |
| **DOF** | **D**egree **O**f **F**reedom |
| **ENO** | **E**ssentially **N**on-**O**scillatory |
| **EOS** | **E**quation **O**f **S**tate |
| **FDM** | **F**inite **D**ifference **M**ethod |
| **FSI** | **F**luid **S**tructure **I**nteraction |
| **FVM** | **F**inite **V**olume **M**ethod |
| **GCL** | **G**eometrical **C**onservation **L**aw |
| **HLL** | **H**arten **L**ax van**L**eer |
| **HLLC** | **H**arten **L**ax van**L**eer **C**ontact |
| **KC** | **K**eulegan-**C**arpenter number |
| **LS** | **L**evel **S**et |
| **MAC** | **M**arker-**A**nd-**C**ell |
| **MLP** | **M**ulti-dimensional **L**imiting **P**rocess |
| **MOOD** | **M**ulti-dimensional **O**ptimal **O**rder **D**etection |
| **MUSCL** | **M**onotonic **U**pstream-**C**entered **S**cheme for **C**onservation **L**aws |
| **ODE** | **O**rdinary **D**ifferential **E**quation |
| **PDE** | **P**artial **D**ifferential **E**quation |
| **PV** | **P**oint **V**alue |
| **RBF** | **R**adial **B**asis **F**unction |
| **SIA** | **S**urface **I**ntegrated **A**verage |
| $\mathcal{TEC}$ | $\mathcal{T}$ime $\mathcal{E}$volution $\mathcal{C}$onverting |

| | |
|---|---|
| **THINC** | **T**angent of **H**yperbola for **IN**terface **C**apturing |
| **THINC-QQ** | **THINC** method with **Q**uadratic surface representation and Gaussian **Q**uadrature |
| **TVD** | **T**otal **V**ariation **D**iminishing |
| **VIA** | **V**olume **I**ntegrated **A**verage |
| **VIV** | **V**ortex **I**nduced **V**ibration |
| **VOF** | **V**olume **O**f **F**luid |
| **VPM** | **V**olume-integrated-average and **P**oint-value based **M**ulti-moment |

*To my family*

# Chapter 1

# Introduction

## 1.1 Background

Fluid flows with multi-materials or sharp interfaces, including shock waves in compressible flows, fluid-structure interactions (FSI), multiphase flows, etc., occur in a wide spectrum of scientific researches and engineering applications. The moving interfaces, which refer to a broad concept covering contact discontinuity, solid-fluid interface and also the interface between two immiscible fluids, significantly affect the fluid flow and attract much attention in fluid mechanics, thus is of primary interest in the current study.

To study this problems, both theoretical and experimental approaches have been considered from the very beginning and achieved many successes. However, though the equations of fluid mechanics, generally Navier-Stokes equations, have been known for over a century, they can only solve a limited number of flows [9]. General solutions to Navier-Stokes equations with given initial conditions and boundary conditions are still unsolved. Even more, the existence and smoothness of solution of Navier-Stokes equations are still not mathematically proved, which were stated as one of the Millennium Prize Problems by Clay Mathematics Institute [10]. For experimental study, the database is insufficient for gradually increased requirements of careful optimization of industrial design or prediction of fluid flows nowadays. Moreover, it may be too costly or time consuming, and difficult if not impossible for certain extreme cases. As a result, computational fluid dynamics (CFD) has become a popular and important auxiliary tool for studying such kind of problems and many other related fields.

## 1.2 Interface tracking and interface capturing techniques

There are a variety of boundary conditions or jump conditions need to be enforced at interfaces in CFD code, thus the interfaces are expected to be treated accurately. To represent and evolve multi-material interfaces or moving boundaries, interface tracking and interface capturing techniques are commonly employed. Interface tracking technique conforms the computational mesh with the moving interfaces, thus the solutions are updated in a moving mesh. With the body-fitted grids, the kinematic and dynamics conditions on the interface can be explicitly specified or computed which ensures the accuracy on the moving interfaces. Moreover, the mesh resolution of the boundary layer around the interfaces can be easily refined. However, remapping process is needed for the case where the interfaces are largely deformed or topologically changed, which may be computationally expensive and technically trivial. On the other hand, interface capturing techniques describe the interface by an indicator function on a fixed spatial domain, which avoid problems of mesh distortion. Nevertheless, the advection of interface inevitably produces the numerical diffusion and the exact implementation of interface conditions is not straightforward. As stated by Tezduyar [11], "for comparable levels of spatial discretization, interface-capturing methods yield less accurate representation of the interface". Due to the inherent superiorities and weaknesses of both techniques, the specific adoptions depend on problems confronted in practical applications.

### 1.2.1 Some related research topics

We discuss a few active research fields involving moving interfaces or boundaries in CFD based on these two techniques as follows:

• Contact discontinuities in compressible flows

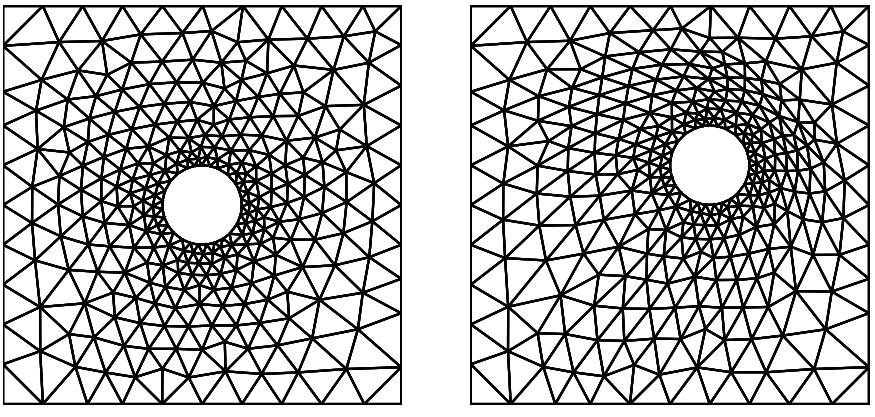
Discontinuous solutions commonly exist in compressible flows, which makes it tough to devise a high accurate numerical scheme without non-physical oscillations. Generally, the solution of Riemann problem contains waves of three types: shock waves, contact discontinuities and rarefaction waves. The accurate reproduction of complex waves are significant for designing high-speed aircraft and some devices like gas turbine, combustion chamber and so on. However, contact discontinuities, across which both pressure and particle velocity are constant but density and some related variables (such as specific internal energy, sound speed, entropy, etc.) jumps discontinuously, are observed to be easily smeared in many numerical tests [12].

Traditionally, numerical methods in CFD are developed in either Eulerian or Lagrangian framework. Eulerian schemes describe the flow in a fixed coordinate, which are convenient but result in inevitable extra errors of convective fluxes or non-physical oscillations near material interfaces. Lagrangian schemes follow moving fluid particles, thus inherently have superiority in tracking multi-material interfaces, particularly contact discontinuity in compressible flows. Unfortunately, it may suffer from mesh tangle due to large flow distortions. Due to the shortcomings of purely Lagrangian and purely Eulerian descriptions, one solution is to devise numerical schemes by combining strong points of both, i.e. the underlying idea of Arbitrary Lagrangian Eulerian (ALE) method [13, 14].

• Fluid Structure interaction

Fluid structure interaction (FSI), such as flapping wings, airfoil oscillations, blood vessels, is an ubiquitous phenomena in nature and engineering applications. In FSI, the spatial domain occupied by fluids always varies with time since structure moves or deforms, and its boundary is a subsequent result of fluid-structure interaction. The accurate treatment of moving boundary of fluid and structure plays an essential role in FSI field, since they are the passage for information communication.

The ways for representing the moving boundaries of fluid and structure can also be classified into interface tracking and interface capturing techniques. Interface tracking technique, also referred to conforming mesh method, as shown in Figure 1.1, treats the solid-fluid interface as the boundary of both physical fields, thus the boundary conditions can be simply and accurately imposed. Interface capturing technique, also called

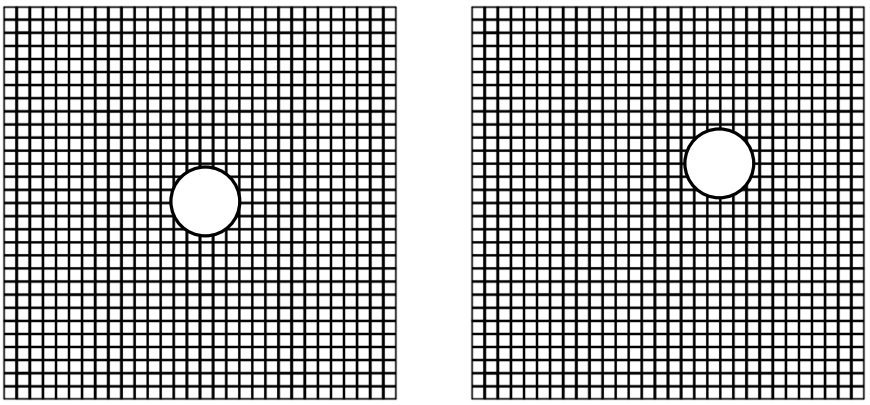


(a) $t = t_0$ (b) $t = t_1$

FIGURE 1.1: Conforming mesh method

as non-conforming mesh method, represents and evolves the interface in a fixed domain

by a color function, such as Level Set (LS) as shown in Figure 1.2. The interface conditions are treated as constraints imposed on the model equations. The practical choice



(a) $t = t_0$ (b) $t = t_1$

FIGURE 1.2: Non-conforming mesh method

for industrial applications lies on the specific engineering problems. For example, interface tracking technique is popularly chosen for studying flows involving mild structure movement and deformation in coastal and ocean engineering, while remapping procedures are required for large movements or distortions. As a contrast, interface capturing technique has the advantage for large deformation problems while with a reduced accuracy for the solution near the interface due to interpolation.

• Two-phase flows

Two-phase flows, referred to gas-liquid flows or two immiscible fluids in this study, is common in nature world, our daily life and also engineering applications, such as: ocean waves, a jet of tap water, fuel spray in engines, etc.. The numerical simulation of two-phase flows are very challenging, since they are usually accompanied with the phenomena of large interface deformation, breakup and coalescence, thus is an active research field in CFD.

The oldest and the most popular approach in this field is to capture the front directly on a regular and stationary grid, such as Marker-And-Cell (MAC) [15], Level Set (LS) [16], Volume-Of-Fluid (VOF) methods [17]. In these approaches, different fluids are distinguished by indicators like signed distance function, VOF function, etc. The evolution of interfaces is realized by moving the indicators with the velocity field of fluids. Currently, most achievements on two-phase flows are obtained based on the concept of capturing interfaces on a stationary grids since it is more available for practical use. However, there are also some researchers devoted to tracking the interface for reaching

higher accuracy by using moving grids [18, 19, 21]. Obviously, it is best suited for relatively simple geometries, and the applications to complex three dimensional problems are rare. The reason may attribute to the extremely flexible movement of free surface.

From the issues discussed above, the strongness and weakness of interface tracking and interface capturing schemes are obvious. Therefore, the basic methodology in this study is to track the multi-material interfaces as far as possible for practical applications, since it inherently ensures the accurate solution at interfaces, while moving interfaces with large deformation, such as free surfaces in multiphase flows, are expected to be captured.

### 1.2.2 Requirements under interface tracking technique

Under interface tracking technique, some requirements for devising numerical model are discussed as follows:

- Adaptability for complex geometry
  Since moving grids are used to fit the multi-material boundaries, the elementary elements are supposed to be adaptable for complex geometrical structures. To this end, unstructured grids, such as triangular and tetrahedral elements, are demanded for the partition of computational domain. However, numerical algorithm on unstructured grids may suffer from the algorithmic complexity, rapid increased computational costs and so on, thus needs some efforts.

- Adequate solution accuracy
  One direction in CFD is to devise accurate and reliable algorithm with high efficiency. However, high order reconstruction polynomials on structured grids built from wide stencils can not be straightforwardly extended to unstructured grids. Moreover, the alleviation of accuracy degradation by mesh quality is pretty important for moving grids.

- Robustness on moving domain
  Numerical robustness on moving domain is a basic requirement for providing friendly user experience. To achieve accurate solutions, stable moving mesh strategies are required to avoid the extremely distorted mesh or tangled mesh. In addition, conservation laws of mesh geometry and physical quantities need to be satisfied, which may benefit numerical stability.

- Stability for interface coupling
  With accurate representation of interface by tracking technique, the boundary conditions or jump conditions on the interface need to be imposed. However, the displacements and velocities of interfaces are the interactive or equilibrious results of two neighbouring materials. For FSI, multi-material equations are supposed to be solved simultaneously for extreme cases, since the the staggered coupling algorithm may cause the disequilibrium of the energy transfer through the interface.

With these in mind, we hope to explore a practical moving mesh framework under interface tracking technique.

## 1.3 Previous and related studies

Two popular methods under interface tracking technique are: Arbitrary Lagrangian Eulerian (ALE) method [13, 22] and space-time method [23, 24]. In ALE formulation, the temporal and spatial derivatives are cast in different descriptions, namely, referential and spatial configuration respectively. By doing this, numerical formulation can be simply written in semi-discrete form, which simplifies the procedure for devising algorithms and thus has gained much popularity in current CFD application. Nevertheless, due to the variation of mesh geometry, efforts are needed to devise the numerical algorithm satisfying several conditions to ensure the numerical stability and accuracy, such as geometrical conservation law (GCL) [25]. Besides, developing high order accurate numerical algorithm is also a challenging work on moving unstructured grid. On the other hand, space-time method, firstly proposed by Hughes [26] in the finite element formulation, exhibits superior advantages for solving moving boundary problems. By integrating the governing equation in a space-time slab, the 3D practical problem becomes a 4D problem including the time dimension, which results in full conservativeness in space and time. It is unconditionally stable and can achieve arbitrarily high order accuracy in time. However, a space-time control volume requires the numerical scheme to be implicit, which is not flexible for time integration. Moreover, the representation of high-dimensional computational element is rather challenging and the computational system may be huge with the largely increased degrees of freedom from all elements. In this study, we prefer to employ ALE method for studying fluid flows on moving domain.

### 1.3.1 ALE method

In ALE-based interface tracking technique, the interface is desired to be tracked in a Lagrangian way. However, as aforementioned, purely Lagrangian schemes are likely to suffer from the mesh tangle for large deformation in practical applications. In the ALE formulations, the grids of the computational domain can move arbitrarily independent of the fluid motions, thus the mesh movement can be arranged artificially. This flexibility of mesh motion makes it more robust compared to the purely Lagrangian framework.

Efforts have been made so far to develop practical ALE algorithms. The existing works may be divided into two classes, i.e. indirect method and direct method. The indirect method was firstly proposed by Hirt et al. [13], which consists of three phases: (1) a Lagrangian phase where the solution and the grid are updated; (2) a rezoning phase that regularizes the tangled or heavily distorted mesh cells; (3) a remapping phase in which the Lagrangian solution is transferred to the rezoned mesh adjusted in step (2). However, rezoning and remapping procedures can be computationally expensive especially for two and three dimensional realistic calculations. The direct method [27, 28] does not need the remapping as a separate step because the mesh velocity is already taken into account in the numerical formulations that are consistent with the governing equations in the moving mesh framework. Both types of methods are widely implemented in the simulations of fluid dynamics.

In ALE schemes, the mesh geometry changes in time, which requires the geometrical quantities, such as volumes, boundary surfaces and vertices of moving cells, to be updated at each time step. Some existing works started from purely Lagrangian framework for developing mesh moving problems. Munz et al. [29] devised a cell-centered Godunov-type scheme using Roe [30] and HLL [31] flux solvers for Lagrangian hydrodynamics equations. Maire et al. [32–34] built a relationship between nodal displacement and flux formulation and presented a robust cell-centered Lagrangian method on multi-dimensional meshes with first and second order accuracy. To achieve high order accuracy, Cheng et al. [35] developed a class of Lagrangian type schemes based on high order essentially non-oscillatory (ENO) [36] reconstruction and obtained third order accuracy with curved mesh in [37]. Dumbser et al. [38] proposed a one-dimensional high order Lagrangian ADER (arbitrary high-order accurate) finite volume method. Besides, the Discontinuous Galerkin (DG) method has also been used to solve the gas dynamics equations in the total Lagrangian formulation for high-order accuracy [39]. These purely Lagrangian methods move mesh cells with the flow velocity so that the advection terms are removed from the governing equations. Particular care must be paid when the

mesh is distorted to an unacceptable extent. A remedy is to improve the mesh quality with subsequent rezoning and remapping steps, which results in the indirect ALE method.

These Lagrangian methods can also be extended to direct ALE methods. In the conventional finite volume framework, the direct ALE scheme can be constructed upon cell-centered Lagrangian solver completed with an edge-based upwinded formulation of the numerical fluxes as presented in [40]. Boscheri et al. developed high-order direct ALE ADER finite volume schemes in [28, 41] with WENO (weighted essentially non-oscillatory) reconstruction and MOOD (multi-dimensional optimal order detection) approach. Moreover, high order discontinuous Galerkin (DG) direct ALE method has been implemented in [27, 42, 43] for compressible Euler equations. In this paper, we focus on direct ALE method where the hydrodynamics equations are solved in an arbitrarily moving coordinate frame.

### 1.3.2 Spatial reconstruction

As stated above, most numerical algorithms has been proposed under the framework of three types: finite difference method (FDM), finite element method (FEM) and finite volume method (FVM). Among these, FVM has gained a great popularity in CFD due to the conservativeness and flexibility for unstructured grids, which is served as the basic framework for devising numerical algorithms in this study.

In the framework of finite volume method, there are mainly two methodologies to devise high order ALE schemes. One is based on the conventional finite volume method, where the computational variables are defined as volume integrated averages (VIA) as shown in Figure 1.3 (a). The reconstruction originally started from piecewise constant and extended to linear reconstruction such as monotonic upstream-centered scheme for conservation laws (MUSCL) [44, 45], which is popularly implemented in mainstream commercial softwares. In recent years, development of high order scheme is an active research direction which is generally based on polynomial-based reconstruction, such as the essentially non-oscillatory (ENO) [36] and weighted ENO (WENO) [46, 47] schemes. These high order schemes presented the excellent performance on structured grids, while choosing the reliable cell stencils for high order polynomial reconstruction on moving unstructured grids is not an easy task.

The other methodology is to use compact stencils by adding the degrees of freedoms (DOF) locally on each cell as shown in Figure 1.3 (a), such as the discontinuous Galerkin

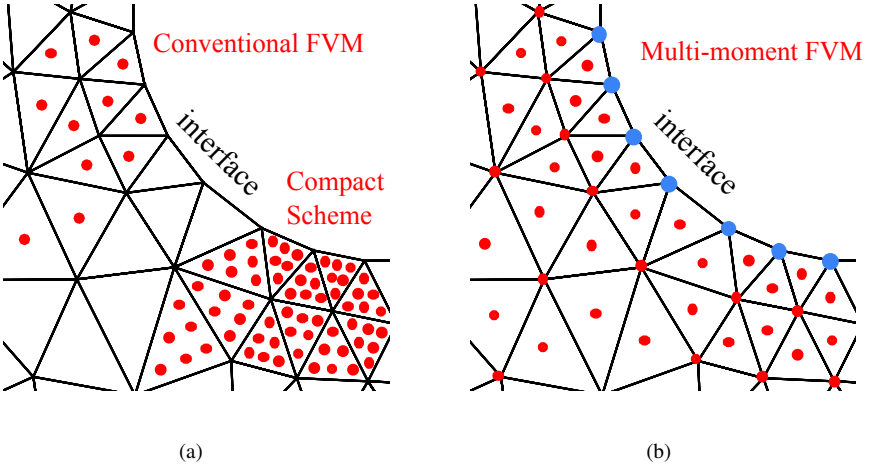



FIGURE 1.3: Definition of degrees of freedom (DOF) for (a) conventional finite volume method and compact scheme, (b) Multi-moment finite volume method.

(DG) method [48, 49] and the spectral finite volume (SV) method [50, 51]. Such methods have superior properties for achieving high order accuracy. As proved in [52], the p-refinement (polynomial refinement) achieves a variable convergence rate with increased DOFs, while the h-refinement (mesh size refinement) only produces a constant convergence rate. However, some issues [53] like more restrictive time step for stability, relative high computational cost and storage cost, need to be considered for industrial applications.

On the other hand, for moving boundary problems, especially for fluid-structure interaction applications, the solutions, as well as the coupling conditions, on the fluid-structure interfaces are crucial and require particular attention. We notice that the aforementioned methods based on increased local DOFs, as well as the conventional FVM, define the computational variables inside each cell, thus extra numerical steps are needed to determine values and conditions on body surfaces or multi-material interfaces. This observation motivates us to use the point values (PV) at cell vertices as computational variables. Making use of both VIA and PVs at the vertices of each mesh cell as the computational variables, the underlying idea of the multi-moment finite volume method, as shown in Figure 1.3(b), possesses at least the following advantages when used for the FSI problems with moving meshes.

- With locally increased DOFs, high order reconstructions can be easily built on unstructured grids;
- PVs are always available at grid points, and can be directly used in moving mesh and FSI computations with efficiency and accuracy;

- The VIA is solved by an FVM formulation, and thus rigorously conserved.

Multi-moment finite volume method based on local reconstruction was developed by Xie et al. [54, 55] on unstructured grids to achieve high order accuracy for incompressible flows in Eulerian representation with fixed grids. In multi-moment finite volume method, both VIA and PVs are used for local polynomial reconstruction, and simultaneously they are treated as computational variables which are solved by integral form and differential form of the governing equations respectively. Successive works for Euler equations have been implemented in [56, 57]. Since the high order polynomial is based on well-determined local cells, it is straightforward to extend previous studies to moving grids. The first objective in this work is to establish an efficient numerical model for fluid flows on moving grids by employing multi-moment finite volume method.

### 1.3.3 Fluid structure interaction

The second objective of this study is to construct an accurate and stable numerical model for fluid-structure interaction based on the aforementioned fluid solver. The coupling strategies for FSI problems can be classified into either the monolithic approach or the partitioned approach. The monolithic approach solves the nonlinear algebraic systems discretized from fluid and solid equations simultaneously. Since the equations for different materials are solved together, the monolithic approach inherently guarantees the synchronism which benefits numerical accuracy and stability, but is usually more computationally cost and requires a great deal of effort to develop the numerical models. The partitioned approach separates different physical fields and solves them independently with the information communicated through the material interfaces. The separate computation makes different numerical schemes be able to be chosen for different fields according to the requirements, which makes the partitioned approach popular in the industrial applications. The partitioned method can be further divided into loosely coupled [58, 59] and strongly coupled schemes [60]. For loosely coupled schemes, the fluid and solid equations are solved only once in sequence at each time step, thus it is simple and efficient. Obviously, it does not rigorously ensure dynamic equilibrium across the interface, which may lead to the instability for a range of problems especially for incompressible flows with small solid/fluid density ratios. Strong coupling scheme enforces the exact coupling conditions between fluid and solid by means of iterations with the implicit numerical schemes devised for both fields. For incompressible flows, the fractional step method [15, 61, 62] is widely used for pressure-velocity coupling. Based on the fractional step method, semi-implicit methods [63] are considered to be

efficient and stable method even for light body involved simulations, where the implicit coupling is only employed into the terms associated with the added-mass effect. In this research, the semi-implicit coupling scheme is devised in the present FSI numerical model for problems with small solid/fluid density ratio and the loosely coupled scheme is used for large solid/fluid density ratio for computational efficiency.

## 1.4 Purpose of current study

In this paper, aiming at various interfaces in fluid flows, like contact discontinuities in compressible flows, boundaries of fluid and structures, etc., we want to track the moving interfaces by using moving grids in order to ensure the accurate solutions. The extremely flexible moving interfaces, such as free surface in two-phase flows, is expected to be captured by transporting indicator functions (i.e. VOF, LS, etc.) on stationary grids or moderately moving grids. Toward this objective, we give the purpose of this study from two parts.

1. Firstly, we will put effort to establish an accurate and robust moving mesh framework for fluid flows based on multi-moment finite volume method. The major concerns are put on following points:

   (a) The motion of fluid flows is described in ALE viewpoint. Thus, the governing equations are formulated on a moving mesh configuration, where the computational grids are able to move arbitrarily.

   (b) To achieve adequate solution accuracy on moving unstructured grids, multi-moment finite volume method will be adopted, where both VIA and PV are treated as computational variables. With locally defined DOFs, quadratic polynomials will be constructed over triangular and quadrilateral elements for 2D and tetrahedral elements for 3D domain which are flexible for fitting complex geometrical structures.

   (c) Multi-moments, namely, VIA and PV, are solved separately on arbitrarily moving grids, where the VIA is calculated by finite volume formulation to ensure the rigorous numerical conservativeness, while the PVs are pointwisely computed through a finite difference method.

   (d) Conservation laws of mesh geometry should be well satisfied to ensure the robustness and accuracy.

We hope to develop numerical models for fluid flows on moving grids based on this framework. The proposed numerical model is expected to achieve 3rd-order accuracy for smooth solutions on varying domain and present appealing performance for tracking moving interfaces by Lagrangian way.

2. Furthermore, we want to present a practical and reliable numerical model for fluid-structure interaction problems of engineering applications. Some issues and our efforts are given as follows:

   (a) During the process of tracking fluid-structure interface, computational grids are supposed to keep validity and good quality during the mesh movement. An efficient and robust mesh movement strategies, radial basis function (RBF) interpolation [64–66], will be implemented to transfer the given boundary point displacements to internal point displacements.

   (b) Interface coupling strategy affects the computational efficiency and stability. Loosely coupled scheme is simple and fast for computation but does not ensure the dynamic equilibrium across the interface. A semi-implicit strong coupling scheme will be presented to calculate problems with small solid/fluid density ratio.

   (c) A preliminary numerical model for a more complex FSI phenomenon, moving body in free surface flow, will be presented. The proposed fluid model on moving mesh will be used for tracking the moving body, while gas-liquid surface will be captured with an algebraic VOF method, THINC-QQ scheme, by formulating two-phase flow into one-fluid model.

Various benchmark tests will be finally calculated to extensively verify the performance of present numerical solver in solving a large class of FSI problems.

## 1.5 Outline of thesis

The rest of the paper is organized as follows. In chapter 2, we first give the basic knowledge of ALE description, and then formulate the Navier Stokes equations of conservation laws on a moving mesh configuration in ALE integral form and differential form, which is prepared for updating VIA and PV moments respectively.

Chapter 3 presents the multi-moment finite volume ALE scheme for Euler equations of compressible gas in 1D and 2D space, which interprets the core idea of the multi-moment based moving mesh framework. The VIA of the conservative variables are

solved by a finite volume method in the integral form of the governing equations to ensure the numerical conservativeness, where the numerical fluxes are calculated directly from the cell boundary PVs in 1D and from the high-order multi-moment reconstruction in 2D. Whereas, the governing equations of differential form are solved for the PVs of the primitive variables by using Roe's Riemann solver. Finally, a series of benchmark tests are computed to verify the convergence rate and the performance for tracking the contact discontinuities.

Chapter 4 extends the numerical model to the viscous compressible flows with a forced moving body. Different from Chapter 3, the PVs are directly computed with differential form in respect of conservative variables which is straightforward for extension to viscous flows. With a forced moving body in the flow, a radial basis function (RBF) interpolation is implemented to transfer the displacement of given boundary points to internal mesh points. Numerical tests, including flow over a oscillating cylinder, are computed to verify the capability of the present solver.

In Chapter 5, we firstly presents the multi-moment finite volume method for unsteady incompressible flows on unstructured moving grids and then construct a numerical model for fluid-solid body interaction. We devise two coupling scheme, i.e. and explicit weak coupling scheme and a semi-implicit strong coupling scheme, to formulate the interactions between fluid and moving solid with a wide range of mass ratios. Various benchmark tests are finally present to verify the numerical model.

In Chapter 6, we present a preliminary numerical model for a more complex FSI phenomenon, moving body in free surface flow. Moving body is tracked by the moving mesh fluid model, while the gas-liquid interface is captured with an algebraic VOF method, THINC-QQ scheme, by formulating two phase flow into one-fluid model. Numerical tests, such as wave generation and water exit, are calculated to verify the numerical model.

Chapter 7 summarizes the present study and offers some conclusions regarding the major contributions. We also mention some deficiencies and provide suggestions for future research.

# Chapter 2

# Governing Equations

## 2.1 Basic knowledge of ALE

### 2.1.1 ALE description

Lagrangian and Eulerian viewpoints are two basic methodologies for describing the motion in continuum mechanics. For the interest of completeness, we start from Lagrangian and Eulerian descriptions, and then give the general formulation, Arbitrary Lagrangian Eulerian description, by following [67–69].

There are two commonly used domains in continuum mechanics: the material domain $\Omega_{\mathbf{X}}$ and the spatial domain $\Omega_{\boldsymbol{x}}$ which consist of material particles $\mathbf{X}$ and $\boldsymbol{x}$ respectively. The Lagrangian viewpoint describes the motion by following the material particles, thus relates to the material coordinate, $\mathbf{X}$ ($\mathbf{X} \in \Omega_{\mathbf{X}}$), or called as Lagrangian coordinate. The Eulerian viewpoint observes the motion from current space, which is related to spatial coordinate, $\boldsymbol{x}$ ($\boldsymbol{x} \in \Omega_{\boldsymbol{x}}$). Therefore, the motion of a specific material point at time $t$ relates the material coordinate $\mathbf{X}$ to the spatial coordinate $\boldsymbol{x}$ by a mapping function $\boldsymbol{\phi}$ as

$$\boldsymbol{x} = \boldsymbol{\phi}(\mathbf{X}, t). \tag{2.1}$$

Then, the material velocity $\boldsymbol{u}$ is defined as

$$\boldsymbol{u} = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\mathbf{X}} \tag{2.2}$$

with $\big|_{\mathbf{X}}$ meaning holding the material coordinate $\mathbf{X}$ fixed.

In ALE description, a referential domain $\Omega_{\boldsymbol{\chi}}$ made up of reference coordinates $\boldsymbol{\chi}$ is introduced, which represents the configuration of mesh movement. Its relationship with spatial domain and material domain can be built by adding two other mapping function $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ as:

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{\chi}, t), \quad \boldsymbol{\chi} = \boldsymbol{\psi}(\mathbf{X}, t). \tag{2.3}$$

Similarly with Eq.(2.2), two velocities are defined as

$$\boldsymbol{u}_g = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{\chi}}, \quad \boldsymbol{w} = \left.\frac{\partial \boldsymbol{\chi}}{\partial t}\right|_{\mathbf{X}}, \tag{2.4}$$

where $\boldsymbol{u}_g$ is called as referential velocity, or grid velocity with $\big|_{\boldsymbol{\chi}}$ meaning holding the referential coordinate $\boldsymbol{\chi}$ fixed, $\boldsymbol{w}$ the particle velocity in referential coordinates with $\big|_{\mathbf{X}}$ meaning holding the material coordinate $\mathbf{X}$ fixed. The three velocities are connected by

$$\left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\mathbf{X}} = \left.\frac{\partial \boldsymbol{\varphi}(\boldsymbol{\chi}, t)}{\partial t}\right|_{\mathbf{X}} = \left.\frac{\partial \boldsymbol{\varphi}(\boldsymbol{\chi}, t)}{\partial t}\right|_{\boldsymbol{\chi}} + \frac{\partial \boldsymbol{\varphi}(\boldsymbol{\chi}, t)}{\partial \boldsymbol{\chi}} \left.\frac{\partial \boldsymbol{\chi}}{\partial t}\right|_{\mathbf{X}}, \tag{2.5}$$

that is

$$\boldsymbol{u} = \boldsymbol{u}_g + \boldsymbol{w}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}}. \tag{2.6}$$

Readers may refer to [68] for rigorous proof.

### 2.1.2 Material, spatial and referential time derivatives

Considering a scalar physical variable $f$, we define the material, referential and spatial time derivatives as $\frac{\partial f}{\partial t}\big|_{\mathbf{X}}$, $\frac{\partial f}{\partial t}\big|_{\boldsymbol{\chi}}$ and $\frac{\partial f}{\partial t}\big|_{\boldsymbol{x}}$ respectively. The material time derivative relates to spatial and referential time derivatives as

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{X}} = \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{u}\frac{\partial f}{\partial \boldsymbol{x}} \tag{2.7}$$

and

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{X}} = \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{w}\frac{\partial f}{\partial \boldsymbol{\chi}}. \tag{2.8}$$

By using Eq.(2.6), Eq.(2.8) can be transformed to

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{X}} = \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{\chi}} + (\boldsymbol{u} - \boldsymbol{u}_g)\frac{\partial f}{\partial \boldsymbol{x}}. \tag{2.9}$$

In this study, we denote $\frac{d\cdot}{dt} = \frac{\partial\cdot}{\partial t}\big|_{\mathbf{X}}$ and $\frac{\partial\cdot}{\partial t} = \frac{\partial\cdot}{\partial t}\big|_{\boldsymbol{x}}$ for convenience, thus Eq.(2.7) and Eq.(2.9) can be rewritten as

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \boldsymbol{u}\frac{\partial f}{\partial \boldsymbol{x}} \tag{2.10}$$

and

$$\frac{df}{dt} = \frac{\partial f}{\partial t}\bigg|_{\boldsymbol{\chi}} + (\boldsymbol{u} - \boldsymbol{u}_g)\frac{\partial f}{\partial \boldsymbol{x}}. \tag{2.11}$$

From Eq.(2.10) and Eq.(2.11), the relation between the referential and spatial time derivatives can be formulated as

$$\frac{\partial f}{\partial t}\bigg|_{\boldsymbol{\chi}} = \frac{\partial f}{\partial t} + \boldsymbol{u}_g\frac{\partial f}{\partial \boldsymbol{x}}, \tag{2.12}$$

which is quite useful in ALE formulations.

### 2.1.3 Time derivatives of integrals over moving control volumes

To establish the integral form of conservation laws for physical variables, the rate change of integrals over moving control volumes is of great interest. We consider a material volume $\Omega_{\mathbf{X}}(t)$ bounded by its boundary $\Gamma_{\mathbf{X}}(t)$, which means all particles are permanently contained in $\Omega_{\mathbf{X}}(t)$. Then, we denote a spatial control volume (fixed in space) $\Omega(t)$ enclosed by it boundary $\Gamma(t)$, which coincides with the material volume $\Omega_{\mathbf{X}}(t)$ at the considered time t. The material time derivative of the integral of a scalar function $f(\boldsymbol{x}, t)$ over the time-varying control volume $\Omega(t)$ is given by the following well-known expression, which is often referred as Reynolds transport theorem,

$$\frac{d}{dt}\int_{\Omega_{\mathbf{X}}(t)} f(\boldsymbol{x}, t)d\Omega = \int_{\Omega(t)} \frac{\partial f(\boldsymbol{x}, t)}{\partial t}d\Omega + \int_{\Gamma(t)} f(\boldsymbol{x}, t)\boldsymbol{u}\cdot\boldsymbol{n}d\Gamma, \tag{2.13}$$

where **n** denotes the surface normal vector of the control volume boundary $\Gamma(t)$, the detailed proof can be found in [70].

Analogically, the referential time derivative of a moving control volume is formulated as

$$\frac{\partial}{\partial t}\int_{\Omega_{\boldsymbol{\chi}}(t)} f(\boldsymbol{x}, t)d\Omega = \int_{\Omega(t)} \frac{\partial f(\boldsymbol{x}, t)}{\partial t}d\Omega + \int_{\Gamma(t)} f(\boldsymbol{x}, t)\boldsymbol{u}_g\cdot\boldsymbol{n}d\Gamma, \tag{2.14}$$

where the spatial control volume $\Omega(t)$ coincides with the referential moving volume $\Omega_{\boldsymbol{\chi}}(t)$ at the considered time $t$ and the moving velocity is given by $\frac{\partial \boldsymbol{x}}{\partial t}\big|_{\boldsymbol{\chi}} = \boldsymbol{u}_g$.

## 2.2 ALE formulations for Navier Stokes equations

### 2.2.1 Navier Stokes equations in Eulerian coordinates

The motion of fluid flows is generally described by the basic principles of conservation of mass, momentum and energy and cast into Navier-Stokes equations with the Newtonian model of viscous stresses, which can be formulated in Eulerian coordinates as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathcal{F}(\mathbf{U}) - \mathcal{G}(\mathbf{U}, \nabla \mathbf{U})) = 0, \tag{2.15}$$

where $\mathbf{U}$ represents the vector of conservative variables, $\mathcal{F}(\mathbf{U})$ and $\mathcal{G}(\mathbf{U}, \nabla \mathbf{U})$ represent the vectors of convective flux and viscous flux respectively. In this formulation, the vectors of conservative variables and fluxes take the form as follows

$$\mathbf{U} = \begin{pmatrix} \rho \\ \mathbf{M} \\ \rho E \end{pmatrix}, \quad \mathcal{F}(\mathbf{U}) = \begin{pmatrix} \boldsymbol{u}\rho \\ \boldsymbol{u} \otimes \mathbf{M} + p\bar{\bar{I}} \\ \boldsymbol{u}\rho E + p\boldsymbol{u} \end{pmatrix}, \quad \mathcal{G}(\mathbf{U}, \nabla \mathbf{U}) = \begin{pmatrix} 0 \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \boldsymbol{u} - \mathbf{Q} \end{pmatrix}, \tag{2.16}$$

with $\rho, \boldsymbol{u}, p, \mathbf{M}$ and $E$ being the density, velocity, pressure, momentum and specific total energy of the fluid respectively. The symbol $\otimes$ denotes the tensor product and $\bar{\bar{I}}$ is the unit tensor. For Newtonian fluid, the stress tensor $\bar{\bar{\tau}}$ is a linear function of velocity gradient based on Stoke's hypothesis, given as

$$\bar{\bar{\tau}} = \mu \left( \nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T - \frac{2}{3} \nabla \cdot \boldsymbol{u} \bar{\bar{I}} \right). \tag{2.17}$$

The dynamic viscosity $\mu$ is represented as a function of temperature $T$ following Sutherland's law

$$\mu(T) = \mu_0 \left( \frac{T}{T_0} \right)^{\beta} \frac{T_0 + s}{T + s}. \tag{2.18}$$

The heat flux vector $\mathbf{Q}$ is given by

$$\mathbf{Q} = -\kappa \nabla T, \tag{2.19}$$

where $\kappa = \frac{c_p \mu}{Pr}$ is the thermal conductivity with $Pr$ and $c_p$ being Prandtl number and the specific heat capacity at constant pressure respectively. We also include the equation of state (EOS) to establish the relationship among three thermodynamic variables in the following general form

$$p = p(\rho, e), \tag{2.20}$$

where $e = E - \frac{1}{2}|\boldsymbol{u}|^2$ is the specific internal energy. We consider the ideal gas in the research with the simpler form as $p = (\gamma - 1)\rho e$. The adiabatic index $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heat capacities at constant presssure and constant volume conditions and we set $\gamma = 1.4$ in this research unless otherwise stated.

### 2.2.2 ALE integral form

Following [71], we integrate Eq.(2.15) over a spatial control volume $\Omega(t)$ (coinciding with a referential moving control volume $\Omega_{\boldsymbol{\chi}}(t)$ at the time $t$) which is enclosed by its boundary $\Gamma(t) \equiv \partial\Omega$ as

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Gamma(t)} \mathcal{F}(\mathbf{U}) \cdot \mathbf{n} d\Gamma - \int_{\Gamma(t)} \mathcal{G}(\mathbf{U}, \nabla \mathbf{U}) \cdot \mathbf{n} d\Gamma = 0, \tag{2.21}$$

where $\mathbf{n}$ denotes the surface normal vector of the control volume boundary $\Gamma(t)$. By using Eq.(2.14), Eq.(2.21) becomes

$$\frac{\partial}{\partial t} \int_{\Omega_{\boldsymbol{\chi}}(t)} \mathbf{U} d\Omega + \int_{\Gamma(t)} (\mathcal{F}(\mathbf{U}) - \boldsymbol{u}_g \mathbf{U}) \cdot \mathbf{n} d\Gamma - \int_{\Gamma(t)} \mathcal{G}(\mathbf{U}, \nabla \mathbf{U}) \cdot \mathbf{n} d\Gamma = 0. \tag{2.22}$$

In this research, we denote the vectors of convective flux and viscous flux as $\mathbf{F}(\mathbf{U}) = (\mathcal{F}(\mathbf{U}) - \boldsymbol{u}_g \mathbf{U}) \cdot \mathbf{n}$ and $\mathbf{G}(\mathbf{U}, \nabla \mathbf{U}) = \mathcal{G}(\mathbf{U}, \nabla \mathbf{U}) \cdot \mathbf{n}$ respectively with $\boldsymbol{u}_g = \frac{\partial \boldsymbol{x}}{\partial t}\big|_{\boldsymbol{\chi}}$ representing the grid moving velocity. Thus, we formulate the ALE integral form of the conservative form of Navier-Stoke equations as

$$\frac{\partial}{\partial t} \int_{\Omega_{\boldsymbol{\chi}}(t)} \mathbf{U} d\Omega + \int_{\Gamma(t)} \mathbf{F}(\mathbf{U}) d\Gamma - \int_{\Gamma(t)} \mathbf{G}(\mathbf{U}, \nabla \mathbf{U}) d\Gamma = 0. \tag{2.23}$$

In the case of uniform flow, where all fluid variables are uniform and constant, Eq.(4.1) should be reduced to

$$\frac{dV}{dt} - \int_{\Gamma(t)} \boldsymbol{u}_g \cdot \mathbf{n} d\Gamma = 0, \tag{2.24}$$

which is the original definition of geometrical conservation law (GCL) [25] in the integral form. Here, $V = \int_{\Omega_{\boldsymbol{\chi}}(t)} d\Omega$ is the control volume.

### 2.2.3 ALE differential form

With the relation between spatial time derivatives and referential time derivatives Eq.(2.12), Eq.(2.15) can be written as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\boldsymbol{\chi}} + \nabla \cdot (\mathcal{F}(\mathbf{U}) - \mathcal{G}(\mathbf{U}, \nabla \mathbf{U})) - \boldsymbol{u}_g \cdot \nabla \mathbf{U} = 0, \tag{2.25}$$

where $\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\boldsymbol{\chi}}$ denotes the time derivative of conservative variables with respect to grid moving frame.

Since the grid velocity $\boldsymbol{u}_g$ is considered in both integral and differential formulations, the system can be discretized on arbitrarily moving grids. Two common cases are: (1) Eulerian framework with $\boldsymbol{u}_g = 0$ in which mesh keeps stationary; (2) Lagrangian framework with $\boldsymbol{u}_g = \boldsymbol{u}$ where the control volume moves with the local fluid velocity.

# Chapter 3

# Multi-moment finite volume ALE scheme for Euler equations

In CFD, PDEs are traditionally discretized on a computational mesh, where the computational variables are defined as moments like Volume Integrated Average (VIA), Point Values (PV), etc.. Different from the conventional FVM or FDM, where only single moment (VIA or PV) is employed, multi-moment finite volume method makes use of multiple moments. In this study, we are interested in building a moving mesh framework based on multi-moment finite volume method by employing both VIA and PV, so called as VPM (Volume integrated average and Point value based Multi-moment) method. To interpret the basic idea, we start from inviscid compressible flows, which are governed by Euler equations. The numerical formulations to compute two kinds of moments, i.e. VIA and PV, are derived respectively from the integral and differential forms of Euler equations.

## 3.1 Governing euqation

### 3.1.1 Integral form

The VIA is updated by a finite volume formulation where we use the integral form of Euler equations. By diminishing the effect of viscosity and heat conduction, we rewrite Eq.(2.23) as

$$\frac{d}{dt}\int_{\Omega_{\boldsymbol{\chi}}(t)} \mathbf{U} d\Omega + \int_{\Gamma(t)} \mathbf{F}(\mathbf{U}) d\Gamma = 0, \tag{3.1}$$

with the vectors of conservative variables **U** and fluxes **F(U)** given below

$$\mathbf{U} = \begin{pmatrix} \rho \\ \mathbf{M} \\ \rho E \end{pmatrix}, \mathbf{F}(\mathbf{U}) = \begin{pmatrix} (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\rho \\ (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\mathbf{M} + p \cdot \mathbf{n} \\ (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\rho E + p\boldsymbol{u} \cdot \mathbf{n} \end{pmatrix}, \tag{3.2}$$

where $\rho$, $\boldsymbol{u}$, $\mathbf{M}$, $p$ and $E$ are the density, velocity, momentum, pressure and specific total energy of the fluid respectively. $\Omega_{\boldsymbol{\chi}}(t)$ is a referential moving control volume with moving velocity $\boldsymbol{u}_g$ which can be set arbitrarily. If $\boldsymbol{u}_g = 0$, the system reduces to a Eulerian framework; and if $\boldsymbol{u}_g = \boldsymbol{u}$, it corresponds to the Lagrangian framework where the control volume moves with the local fluid velocity.

Considering the ideal gas, the equation of state (EOS) becomes

$$p = (\gamma - 1)\rho e, \tag{3.3}$$

with $e = E - \frac{1}{2}|\boldsymbol{u}|^2$ being the specific internal energy, $\gamma$ the ratio of specific heats.

Besides, the volume variation is given by following geometric conservation law (GCL) [25]

$$\frac{dV}{dt} - \int_{\Gamma(t)} \boldsymbol{u}_g \cdot \mathbf{n} d\Gamma = 0, \tag{3.4}$$

where $V$ denotes the control volume.

### 3.1.2 Differential form

In this study, differential form of compressible Euler equations is used to predict the PV moment, which can be obtained from Eq.(2.25) as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\boldsymbol{\chi}} + \nabla \cdot \mathcal{F}(\mathbf{U}) - \mathbf{u_g} \cdot \nabla \mathbf{U} = \mathbf{0}. \tag{3.5}$$

As we can see from Eq.(3.5), there is an extra term related to the gradient of variables, $\boldsymbol{u}_g \cdot \nabla \mathbf{U}$, in the conservation laws, which can not be calculated separately from the divergence term. In practice, we cast Eq.(3.5) into two kind of quasi-linear forms in 2D.

***Form* 1**

Eq.(3.5) is cast in terms of primitive variables:

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_{\chi} + \boldsymbol{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} + \boldsymbol{B}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial y} = 0, \tag{3.6}$$

where $\mathbf{W} = [\rho, u, v, p]$ is the vector of primitive variables, $\boldsymbol{A}(\mathbf{W})$ and $\boldsymbol{B}(\mathbf{W})$ are the coefficient matrices for spatial derivatives of $\mathbf{W}$ with respect to $x$ and $y$ coordinates respectively. They take the following form

$$\boldsymbol{A}(\mathbf{W}) = \begin{bmatrix} u-u_g & \rho & 0 & 0 \\ 0 & u-u_g & 0 & \frac{1}{\rho} \\ 0 & 0 & u-u_g & 0 \\ 0 & \gamma p & 0 & u-u_g \end{bmatrix},$$

$$\boldsymbol{B}(\mathbf{W}) = \begin{bmatrix} v-v_g & 0 & \rho & 0 \\ 0 & v-v_g & 0 & 0 \\ 0 & 0 & v-v_g & \frac{1}{\rho} \\ 0 & 0 & \gamma p & v-v_g \end{bmatrix},$$

where $\rho$ is the density, $p$ the pressure, $u$ and $v$ the $x$ and $y$ components of fluid velocity $\boldsymbol{u} = (u, v)$, $u_g$ and $v_g$ $x$ and $y$ components of the grid velocity $\boldsymbol{u}_g = (u_g, v_g)$ respectively. Note that in (3.6), $\frac{\partial \mathbf{W}}{\partial t}\big|_\chi$ denotes the time derivative with respect to the grid moving at speed $\boldsymbol{u}_g$. The grid velocity $\boldsymbol{u}_g = \frac{\partial \boldsymbol{x}}{\partial t}\big|_{\boldsymbol{\chi}}$ can be formulated more directly as

$$\boldsymbol{u}_g = \frac{d\boldsymbol{x}_g}{dt}, \tag{3.7}$$

where $\boldsymbol{x}_g$ is the coordinate of the grid. This is the so-called local kinematic equation, which corresponds to GCL in the integral form (3.4). If $\boldsymbol{u}_g = 0$, the system reduces to the differential form of Euler equations in the Eulerian framework; and if $\boldsymbol{u}_g = \boldsymbol{u}$, the system becomes the Lagrangian framework.

***Form* 2**

Eq.(3.5) can also be formulated with respect to conservative variables $\mathbf{U} = [\rho, \rho u, \rho v, \rho E]$ as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_\chi + (\mathcal{A}(\mathbf{U}) - u_g \boldsymbol{I})\frac{\partial \mathbf{U}}{\partial x} + (\mathcal{B}(\mathbf{U}) - v_g \boldsymbol{I})\frac{\partial \mathbf{U}}{\partial y} = 0, \tag{3.8}$$

where $\mathcal{A}(\mathbf{U}) = \frac{\partial \mathcal{F}^x(\mathbf{U})}{\partial \mathbf{U}}$ and $\mathcal{B}(\mathbf{U}) = \frac{\partial \mathcal{F}^y(\mathbf{U})}{\partial \mathbf{U}}$ are the Eulerian Jacobian matrices, $\boldsymbol{I}$ is the unit matrix, $u_g$ and $v_g$ are x and y components of grid velocity $\boldsymbol{u}_g$ respectively. Thus, we formulate the ALE differential form of compressible Navier-Stoke equations as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_\chi + \boldsymbol{A}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} + \boldsymbol{B}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial y} = 0 \tag{3.9}$$

with the ALE Jacobian matrices $\boldsymbol{A}(\mathbf{U}) = \mathcal{A}(\mathbf{U}) - u_g\boldsymbol{I}$ and $\boldsymbol{B}(\mathbf{U}) = \mathcal{B}(\mathbf{U}) - v_g\boldsymbol{I}$. Jacobian matrices take the form as

$$\boldsymbol{A}(\mathbf{U}) = \begin{bmatrix} -u_g & 1 & 0 & 0 \\ -u^2 + \frac{1}{2}(\gamma-1)\boldsymbol{V}^2 & (3-\gamma)u - u_g & (1-\gamma)v & \gamma - 1 \\ -uv & v & u - u_g & 0 \\ u\left[\frac{1}{2}(\gamma-1)\boldsymbol{V}^2 - H\right] & H - (\gamma-1)u^2 & (1-\gamma)uv & \gamma u - u_g \end{bmatrix}, \quad (3.10)$$

$$\boldsymbol{B}(\mathbf{U}) = \begin{bmatrix} -v_g & 0 & 1 & 0 \\ -uv & v - v_g & u & 0 \\ -v^2 + \frac{1}{2}(\gamma-1)\boldsymbol{V}^2 & (1-\gamma)u & (3-\gamma)v - v_g & \gamma - 1 \\ v\left[\frac{1}{2}(\gamma-1)\boldsymbol{V}^2 - H\right] & (1-\gamma)uv & H - (\gamma-1)v^2 & \gamma v - v_g \end{bmatrix}, \quad (3.11)$$

with the specific kinetic energy $\frac{1}{2}\boldsymbol{V}^2 = \frac{1}{2}(u^2+v^2)$ and the total specific enthalpy $H = e + \frac{p}{\rho}$.

Finally, both ***Form* 1** and ***Form* 2** can be summarized into a general quasi-linear form

$$\left.\frac{\partial \mathbf{Q}}{\partial t}\right|_{\chi} + \boldsymbol{A}\frac{\partial \mathbf{Q}}{\partial x} + \boldsymbol{B}\frac{\partial \mathbf{Q}}{\partial y} = 0 \quad (3.12)$$

with $\boldsymbol{Q}$ being $\boldsymbol{W}$ or $\boldsymbol{U}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ corresponding matrices, thus the numerical procedures for two forms are identical. In this study, we tried both ***Form* 1** and ***Form* 2** for solving PV moment of Euler equations and don't see much difference in our practice. However, ***Form* 1** can not be directly extended to viscous compressible flows due to the transformation between conservative variables and primitive variables. Thus, We will present the numerical scheme for Euler equations based on ***Form* 1** for solving PV moments in this Chapter, and use ***Form* 2** for viscous compressible flows as shown in Chapter 4.

## 3.2 Multi-moment finite volume ALE scheme for 1D Euler equation

The one-dimensional computational domain is divided into $I$ non-overlapping cells $\Omega_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ of size $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ with $i = 1, 2, \cdots, I$. For a given cell $\Omega_i$, two kinds of moments are defined for field variable $\phi(x,t)$. They are the volume

integrated average (VIA) over each cell

$$\overline{\phi}_i = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(x,t)dx \tag{3.13}$$

and the point value (PV) at cell boundary

$$\phi_{i+\frac{1}{2}} = \phi(x_{i+\frac{1}{2}}, t), \tag{3.14}$$

where $\phi$ stands for all state variables, including conservative and primitive variables.

### 3.2.1 The CIP-CSL3 reconstruction

Given one VIA $\overline{\phi}_i$ and two PVs $\phi_{i\pm\frac{1}{2}}$ over $\Omega_i$, as well as the first order cell center derivative $d_i$ approximated from the VIA and PV moments from $\Omega_i$ and its neighboring cells, a piecewise cubic interpolation function for conservative variables $\phi(x,t)$ can be constructed, following the CIP-CSL3 (constrained interpolation profile-conservative semi-Lagrangian scheme with third-order polynomial function) scheme [72, 73], for each element as

$$\Phi_i(x) = a_3(x - x_{i-\frac{1}{2}})^3 + a_2(x - x_{i-\frac{1}{2}})^2 + a_1(x - x_{i-\frac{1}{2}}) + a_0, \tag{3.15}$$

and the unknown coefficients are computed by imposing the following constraint conditions:

$$\begin{cases} \Phi_i(x_{i-\frac{1}{2}}) = \phi_{i-\frac{1}{2}}, \\ \Phi_i(x_{i+\frac{1}{2}}) = \phi_{i+\frac{1}{2}}, \\ \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Phi_i(x)dx = \overline{\phi}_i, \\ \frac{d\Phi_i(x)}{dx}\Big|_{x_i} = d_i. \end{cases} \tag{3.16}$$

Thus, we have coefficients of (3.15) as

$$\begin{cases} a_0 = \phi_{i-\frac{1}{2}}, \\ a_1 = \frac{2(3\overline{\phi}_i - 3\phi_{i-\frac{1}{2}} - \Delta x_i d_i)}{\Delta x_i}, \\ a_2 = \frac{3(-2\overline{\phi}_i - \phi_{i+\frac{1}{2}} + 3\phi_{i-\frac{1}{2}} + 2\Delta x_i d_i)}{\Delta x_i^2}, \\ a_3 = \frac{4(\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}} - \Delta x_i d_i)}{\Delta x_i^3}. \end{cases} \tag{3.17}$$

When the cell center derivative $d_i$ is simply computed by differencing the PVs at the two boundaries of cell $\Omega_i$ as

$$d_i = \frac{\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}}}{\Delta x_i}, \tag{3.18}$$

the coefficient of the third-order term $a_3$ vanishes in (3.17), and the interpolation function then degrades to a piecewise quadratic interpolation function for cell $\Omega_i$ and the CIP-CSL2 (constrained interpolation profile-conservative semi-Lagrangian scheme with second-order polynomial function) scheme [74] is retrieved. As shown in [72, 73], the cell center derivative can be approximated by conventional slope limiter schemes to suppress the numerical oscillations. We use the superbee limiter [75] as follows

$$d_i = maxmod\left(minmod\left(d_{i-\frac{1}{2}}, 2d_{i+\frac{1}{2}}\right), minmod\left(2d_{i-\frac{1}{2}}, d_{i+\frac{1}{2}}\right)\right), \tag{3.19}$$

where $d_{i-\frac{1}{2}} = \frac{2(\overline{\phi}_i - \overline{\phi}_{i-1})}{\Delta x_{i-1} + \Delta x_i}$ and $d_{i+\frac{1}{2}} = \frac{2(\overline{\phi}_{i+1} - \overline{\phi}_i)}{\Delta x_i + \Delta x_{i+1}}$. The $minmod$ and $maxmod$ functions are given respectively as

$$minmod(a_1, a_2) = \begin{cases} s \cdot min(|a_1|, |a_2|), & \text{if } s = sign(a_1) = sign(a_2), \\ 0, & \text{otherwise,} \end{cases} \tag{3.20}$$

and

$$maxmod(a_1, a_2) = \begin{cases} s \cdot max(|a_1|, |a_2|), & \text{if } s = sign(a_1) = sign(a_2), \\ 0, & \text{otherwise.} \end{cases} \tag{3.21}$$

Once the CIP-CSL3 interpolation is constructed over $\Omega_i$, we can obtain the spatial derivatives of the physical variables at two boundary points $x_{i\pm\frac{1}{2}}$ which are then used to update the PVs at cell boundaries.

### 3.2.2 Solution for VIA moment

We first formulate the semi-discrete finite volume scheme of (3.1) for updating VIA moments as

$$\frac{d}{dt}\begin{pmatrix} \overline{\rho}_i \Delta x_i \\ \overline{M}_i \Delta x_i \\ \overline{(\rho E)}_i \Delta x_i \end{pmatrix} = -\begin{pmatrix} \hat{f}^D_{i+\frac{1}{2}} - \hat{f}^D_{i-\frac{1}{2}} \\ \hat{f}^M_{i+\frac{1}{2}} - \hat{f}^M_{i-\frac{1}{2}} \\ \hat{f}^E_{i+\frac{1}{2}} - \hat{f}^E_{i-\frac{1}{2}} \end{pmatrix}, \tag{3.22}$$

with the fluxes approximated from PVs as

$$\begin{cases} \hat{f}^D_{i+\frac{1}{2}} = (u_{i+\frac{1}{2}} - u_{g,i+\frac{1}{2}})\rho_{i+\frac{1}{2}}, \\ \hat{f}^M_{i+\frac{1}{2}} = (u_{i+\frac{1}{2}} - u_{g,i+\frac{1}{2}})M_{i+\frac{1}{2}} + p_{i+\frac{1}{2}}, \\ \hat{f}^E_{i+\frac{1}{2}} = (u_{i+\frac{1}{2}} - u_{g,i+\frac{1}{2}})(\rho E)_{i+\frac{1}{2}} + p_{i+\frac{1}{2}}u_{i+\frac{1}{2}}. \end{cases} \tag{3.23}$$

It is noted that by solving the physical variables as computational variables at cell boundaries we don't need to solve any Riemann problem to find the numerical fluxes at cell boundaries which are required in the finite volume formulations for the VIAs of the conservative variables. Thus, the present method is computationally efficient.

### 3.2.3 Solution for PV moment

The governing equations (3.6) of PVs in one-dimensional case are formulated by

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} = 0, \tag{3.24}$$

with $\mathbf{W} = [\rho, u, p]$ being the primitive variables. The Jacobian matrix can be factorized into $\mathbf{A} = \mathbf{R_A}\mathbf{\Lambda_A}\mathbf{L_A}$, where $\mathbf{\Lambda_A}$ is the diagonal matrix of eigenvalues, and $\boldsymbol{L}_\mathbf{A}$ and $\boldsymbol{R}_\mathbf{A}$ are the corresponding left and right eigen matrices respectively. Details of eigenstructure are documented in Appendix A.1.

As introduced in [57], Eq.(3.24) is solved by Roe's Riemann solver point-wisely as

$$\frac{\partial \mathbf{W}_{i+\frac{1}{2}}}{\partial t} = -\frac{1}{2}\left(\tilde{\mathbf{A}}\left(\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x} + \frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}\right) + \tilde{\mathbf{R}}_\mathbf{A}|\tilde{\mathbf{\Lambda}}_\mathbf{A}|\tilde{\mathbf{L}}_\mathbf{A}\left(\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x} - \frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}\right)\right), \tag{3.25}$$

where $\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x}$ and $\frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}$ denote the derivatives of variables $\mathbf{W}_{i+\frac{1}{2}}$ on the left and right sides of point $x_{i+\frac{1}{2}}$ and the Jacobian matrix is calculated from the Roe-averaging. Each Roe-averaging value is computed from the VIA values in the surrounding cells by

$$\tilde{\phi}_{i+\frac{1}{2}} = \frac{\sqrt{\overline{\rho}_i}\cdot\overline{\phi}_i + \sqrt{\overline{\rho}_{i+1}}\cdot\overline{\phi}_{i+1}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}}, \tag{3.26}$$

where $\tilde{\phi}_{i+\frac{1}{2}}$ stands for $\tilde{u}_{i+\frac{1}{2}}$ and $\tilde{H}_{i+\frac{1}{2}}$ and $\overline{H}_i = \left(\overline{(\rho E)}_i + \overline{p}_i\right)/\overline{\rho}_i$ is the total enthalpy. We compute the sound speed by

$$\tilde{a}_{i+\frac{1}{2}} = \left((\gamma-1)(\tilde{H}_{i+\frac{1}{2}} - \frac{1}{2}\tilde{u}^2_{i+\frac{1}{2}})\right)^{\frac{1}{2}}. \tag{3.27}$$

As shown in 3.2.1, the interpolation function is reconstructed for the conservative variables $\rho$, $M$ and $\rho E$, thus the spatial derivatives of conservative variables must be transferred to the spatial derivatives of primitive variables $\mathbf{W} = [\rho, u, p]$. We use the following formulations for this purpose in this work

$$\frac{\partial u^{\pm}_{i+\frac{1}{2}}}{\partial x} = \frac{1}{\tilde{\rho}_{i+\frac{1}{2}}}\frac{\partial M^{\pm}_{i+\frac{1}{2}}}{\partial x} - \frac{\tilde{M}_{i+\frac{1}{2}}}{\tilde{\rho}^2_{i+\frac{1}{2}}}\frac{\partial \rho^{\pm}_{i+\frac{1}{2}}}{\partial x}, \tag{3.28}$$

$$\frac{\partial p^{\pm}_{i+\frac{1}{2}}}{\partial x} = (\gamma-1)\left(\frac{\partial (\rho E)^{\pm}_{i+\frac{1}{2}}}{\partial x} - \tilde{u}_{i+\frac{1}{2}}\frac{\partial M^{\pm}_{i+\frac{1}{2}}}{\partial x} + \frac{1}{2}\tilde{u}^2_{i+\frac{1}{2}}\frac{\partial \rho^{\pm}_{i+\frac{1}{2}}}{\partial x}\right). \tag{3.29}$$

The semi-discrete time evolution equations (3.25) are then solved to update the PVs at cell boundaries.

### 3.2.4 Time integration

In order to achieve third order accuracy in time, we implement the third order Runge-Kutta scheme [76] to update the semi-discrete equations (3.22) and (3.25) in time. Since the mesh changes with the time in ALE scheme, not only VIAs and PVs but also the position of each vertex and the size of each cell should be updated in each Runge-Kutta step.

We summarize the solution procedure using the third order Runge-Kutta scheme to update the numerical solutions of Euler equations from time level $n$ ($t = t^n$) to $n+1$

($t = t^n + \Delta t$) as follows,

Step 1:

$$
\begin{aligned}
&x_{i+\frac{1}{2}}^{(1)} = x_{i+\frac{1}{2}}^{n} + u_{i+\frac{1}{2}}^{n}\Delta t, \quad \Delta x_i^{(1)} = x_{i+\frac{1}{2}}^{(1)} - x_{i-\frac{1}{2}}^{(1)},\\
&\overline{\mathbf{U}}_i^{(1)}\Delta x_i^{(1)} = \overline{\mathbf{U}}_i^{n}\Delta x_i^{n} + \Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{n}, \mathbf{W}_{i+\frac{1}{2}}^{n}),\\
&\mathbf{W}_{i+\frac{1}{2}}^{(1)} = \mathbf{W}_{i+\frac{1}{2}}^{n} + \Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{n}, \mathbf{W}_{i+\frac{1}{2}}^{n});
\end{aligned}
$$

Step 2:

$$
\begin{aligned}
&x_{i+\frac{1}{2}}^{(2)} = \frac{3}{4}x_{i+\frac{1}{2}}^{n} + \frac{1}{4}\left[x_{i+\frac{1}{2}}^{(1)} + u_{i+\frac{1}{2}}^{(1)}\Delta t\right], \quad \Delta x_i^{(2)} = x_{i+\frac{1}{2}}^{(2)} - x_{i-\frac{1}{2}}^{(2)},\\
&\overline{\mathbf{U}}_i^{(2)}\Delta x_i^{(2)} = \frac{3}{4}\overline{\mathbf{U}}_i^{n}\Delta x_i^{n} + \frac{1}{4}\left[\overline{\mathbf{U}}_i^{(1)}\Delta x_i^{1} + \Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{(1)}, \mathbf{W}_{i+\frac{1}{2}}^{(1)})\right],\\
&\mathbf{W}_{i+\frac{1}{2}}^{(2)} = \frac{3}{4}\mathbf{W}_{i+\frac{1}{2}}^{n} + \frac{1}{4}\left[\mathbf{W}_{i+\frac{1}{2}}^{(1)} + \Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{(1)}, \mathbf{W}_{i+\frac{1}{2}}^{(1)})\right];
\end{aligned}
$$

Step 3:

$$
\begin{aligned}
&x_{i+\frac{1}{2}}^{n+1} = \frac{1}{3}x_{i+\frac{1}{2}}^{n} + \frac{2}{3}\left[x_{i+\frac{1}{2}}^{(2)} + u_{i+\frac{1}{2}}^{(2)}\Delta t\right], \quad \Delta x_i^{n+1} = x_{i+\frac{1}{2}}^{n+1} - x_{i-\frac{1}{2}}^{n+1},\\
&\overline{\mathbf{U}}_i^{n+1}\Delta x_i^{n+1} = \frac{1}{3}\overline{\mathbf{U}}_i^{n}\Delta x_i^{n} + \frac{2}{3}\left[\overline{\mathbf{U}}_i^{(2)}\Delta x_i^{(2)} + \Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{(2)}, \mathbf{W}_{i+\frac{1}{2}}^{(2)})\right],\\
&\mathbf{W}_{i+\frac{1}{2}}^{n+1} = \frac{1}{3}\mathbf{W}_{i+\frac{1}{2}}^{n} + \frac{2}{3}\left[\mathbf{W}_{i+\frac{1}{2}}^{(2)} + \Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{(2)}, \mathbf{W}_{i+\frac{1}{2}}^{(2)})\right],
\end{aligned}
$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ stand for the numerical operators of the right sides of (3.22) and (3.25) respectively. The time step is simply determined by

$$
\Delta t = C_E \min_{1\leqslant i\leqslant I}\left(\frac{\Delta x_i^n}{|u_i^n| + c_i^n}\right), \tag{3.30}
$$

where we set $C_E = 0.4$ in our computations except for the blast wave problem.

## 3.3 Benchmark tests in 1D

In this section, one-dimensional ALE multi-moment finite volume scheme for Euler equations is tested by some benchmark tests. To verify that the mesh can move arbitrarily for the proposed ALE scheme, we set the grid velocity as a fraction of the local fluid velocity in this chapter. With $u_g = 0$ for Eulerian framework and $u_g = u$ for Lagrangian framework, we include the case of $u_g = 0.5u$ to demonstrate the present ALE framework with a specified speed to move the mesh. It is noted that numerical formulation of the present ALE works well for any specified mesh speed. The ideal gas with $\gamma = 1.4$ is used unless otherwise stated and also for later 2D tests.

### 3.3.1 Accuracy test

The accuracy of the proposed scheme is evaluated by a benchmark test with smooth solutions [35]. The initial conditions are

$$\rho(x,0) = 2 + \sin(2\pi x), \quad u(x,0) = 1 + 0.1\sin(2\pi x), \quad p(x,0) = 1, \quad x \in [0,1], \tag{3.31}$$

and the periodic boundary condition is used. We tested the present ALE scheme in the Eulerian and Lagrangian frameworks as two special cases and also another moving-mesh case with the grid velocity specified as half of the fluid velocity ($u_g = 0.5u$). We summarize in Table 3.1 the errors and numerical convergence rates for the density field at $t = 1$ with the reference solution calculated from the third order multi-moment constraint finite volume method (MCV) [77] in the Eulerian framework with $8000$ grids. To show the accuracy improvements of the present multi-moment ALE scheme in comparison with the conventional finite volume ALE method, we also calculated the same benchmark test by a conventional third-order finite volume ALE scheme using the quadratic polynomials constructed over a three-cell stencil. Numerical results of third-order finite volume ALE scheme are shown in Table 3.2 for comparison, and we compare the computation time of both schemes in Table 3.3.

TABLE 3.1: Errors and convergence rates of density with initially uniform cells by using the present ALE scheme

| cells | Eulerian ($u_g = 0$) | | | | ALE ($u_g = 0.5u$) | | | | Lagrangian ($u_g = u$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate |
| 50 | 1.96e-4 | - | 7.70e-4 | - | 1.06e-4 | - | 5.25e-4 | - | 6.88e-5 | - | 3.12e-4 | - |
| 100 | 2.52e-5 | 2.96 | 1.04e-4 | 2.89 | 1.39e-5 | 2.93 | 7.00e-5 | 2.91 | 8.88e-6 | 2.95 | 4.17e-5 | 2.90 |
| 200 | 3.16e-6 | 3.00 | 1.32e-5 | 2.98 | 1.74e-6 | 3.00 | 8.97e-6 | 2.96 | 1.11e-6 | 3.00 | 5.36e-6 | 2.96 |
| 400 | 3.96e-7 | 3.00 | 1.66e-6 | 2.99 | 2.19e-7 | 2.99 | 1.13e-6 | 2.99 | 1.40e-7 | 2.99 | 6.76e-7 | 2.99 |
| 800 | 4.94e-8 | 3.00 | 2.08e-7 | 3.00 | 2.73e-8 | 3.00 | 1.41e-7 | 3.00 | 1.74e-8 | 3.01 | 8.46e-8 | 3.00 |

TABLE 3.2: Errors and convergence rates of density with initially uniform cells by using 3rd-order finite volume ALE scheme

| cells | Eulerian ($u_g = 0$) | | | | ALE ($u_g = 0.5u$) | | | | Lagrangian ($u_g = u$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate |
| 50 | 1.12e-3 | - | 4.34e-3 | - | 6.33e-4 | - | 3.11e-3 | - | 3.82e-4 | - | 1.68e-3 | - |
| 100 | 1.46e-4 | 2.93 | 5.95e-4 | 2.87 | 8.34e-5 | 2.93 | 4.17e-4 | 2.90 | 4.93e-5 | 2.95 | 2.59e-4 | 2.70 |
| 200 | 1.84e-5 | 2.99 | 7.54e-5 | 2.98 | 1.05e-5 | 2.99 | 5.27e-5 | 2.98 | 6.23e-6 | 2.98 | 3.36e-5 | 2.95 |
| 400 | 2.28e-6 | 3.01 | 9.41e-6 | 3.00 | 1.32e-6 | 3.00 | 6.59e-6 | 3.00 | 7.82e-7 | 2.99 | 4.25e-6 | 2.98 |
| 800 | 2.54e-7 | 3.17 | 1.18e-6 | 3.00 | 1.67e-7 | 2.98 | 8.56e-7 | 2.95 | 1.02e-7 | 2.94 | 5.63e-7 | 2.91 |

It is clearly that the present ALE scheme can achieve third order accuracy. By the comparison, we can find that with same mesh resolution, the numerical error of present

TABLE 3.3: Computational time comparison of the present ALE scheme and 3rd-order finite volume ALE scheme. The elapse time of computations was measured on a PC with an Intel Core i7-4790 CPU @ 3.60GHz. $500, 1000, 2000, 4000$ and $8000$ time-steps were used respectively for the cases of $50, 100, 200, 400$ and $800$ cells.

| | The present scheme | | | 3rd-order FVM | | |
|---|---|---|---|---|---|---|
| cells | Eulerian ($u_g = 0$) | ALE ($u_g = 0.5u$) | Lagrangian ($u_g = u$) | Eulerian ($u_g = 0$) | ALE ($u_g = 0.5u$) | Lagrangian ($u_g = u$) |
| 50 | 0.05s | 0.04s | 0.04s | 0.04s | 0.04s | 0.03s |
| 100 | 0.15s | 0.15s | 0.14s | 0.11s | 0.12s | 0.12s |
| 200 | 0.55s | 0.57s | 0.55s | 0.45s | 0.46s | 0.45s |
| 400 | 2.15s | 2.19s | 2.17s | 1.79s | 1.78s | 1.77s |
| 800 | 8.52s | 8.53s | 8.57s | 6.90s | 7.01s | 7.08s |

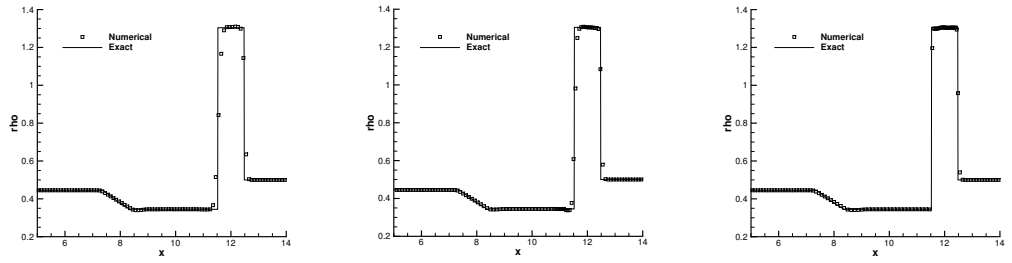
scheme is only $15 \sim 20\%$ of the conventional 3rd-order finite volume ALE scheme no matter mesh moves or not. Since two kinds of moment are used, the computational cost increases $20 \sim 25\%$. However it is not substantial compared with the reduction of the numerical error. Besides, it is also observed that the Lagrangian framework of both methods shows the smallest numerical errors compared to the other two frameworks.

### 3.3.2 Lax problem

Known as Lax's Riemann problem for Euler equations, this benchmark test starts from the following initial conditions,

$$(\rho_0, u_0, p_0) = \begin{cases} (0.445, 0.698, 3.528), & \text{for } 4 \leqslant x \leqslant 10, \\ (0.5, 0, 0.571), & \text{for } 10 < x \leqslant 14. \end{cases} \tag{3.32}$$

Figure 3.1 shows the results of the Eulerian framework (a), the ALE framework (b) and the Lagrangian framework (c) with $100$ initially uniform cells at time $t = 1$. The present scheme can resolve the discontinuous solution well with the superbee limiter. We can see from Figure 3.1 that the Lagrangian framework can capture sharper contact discontinuity than the Eulerian framework. The ALE scheme with $u_g = 0.5u$ gives numerical results lying between Eulerian and Lagrangian frameworks.




(a) (b) (c)

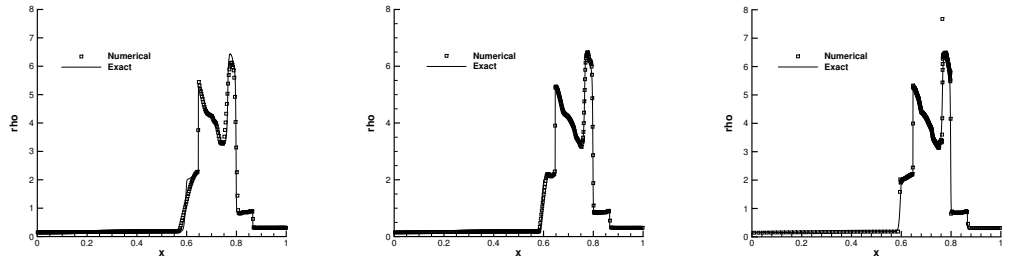
FIGURE 3.1: The density results for Lax problem with the present ALE scheme: (a) Eulerian ($u_g = 0$); (b) ALE ($u_g = 0.5u$); (c) Lagrangian ($u_g = u$).

### 3.3.3 Two interacting blast waves

The benchmark test of two interacting blast waves [78] was computed. The initial conditions were specified as

$$\rho_0 = 1, \quad u_0 = 1, \quad p_0 = \begin{cases} 1000, & \text{for } 0 \leqslant x \leqslant 0.1, \\ 0.01, & \text{for } 0.1 < x \leqslant 0.9, \\ 100, & \text{for } 0.9 < x \leqslant 1. \end{cases} \tag{3.33}$$

The reflective boundary conditions were applied at the two ends of computation domain, $x = 0$ and $x = 1$. We solved this problem with 400 initially uniform cells to time $t = 0.38$, and the CFL number is set as $0.1$. The density results of the Eulerian, ALE and Lagrangian framework are plotted in Figure 3.2. The left-most contact discontinu-




(a) (b) (c)

FIGURE 3.2: The density results for blast waves with the present ALE scheme: (a) Eulerian ($u_g = 0$); (b) ALE ($u_g = 0.5u$); (c) Lagrangian ($u_g = u$).

ity in the Eulerian framework is diffused as observed in the numerical solutions of other high-resolution schemes, while the numerical solutions of the Lagrangian framework demonstrate much superior capability in resolving both discontinuities and smooth solutions. The Lagrangian framework can resolve the contact discontinuity within two

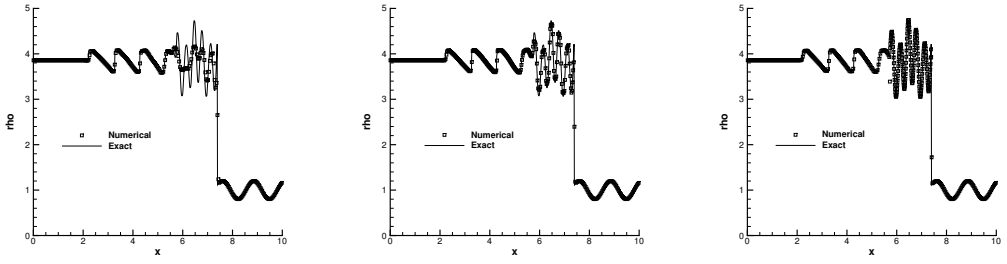
cells, which is very challenging to Eulerian sort approach even with high-order reconstruction. Meanwhile, we also observe some overshoots in the Lagrangian framework quite similar to that reported in [35]. Again, the ALE result with $u_g = 0.5u$ shows a result lying between Eulerian and Lagrangian frameworks.

### 3.3.4 Shock entropy wave interaction

This problem was proposed in [79]. The initial conditions are given as

$$(\rho_0, u_0, p_0) = \begin{cases} (3.857148, 2.629369, 10.333333), & \text{for } -5 \leqslant x \leqslant 1, \\ (1 + 0.2\sin(5x - 5), 0, 1), & \text{for } 1 < x \leqslant 10. \end{cases} \tag{3.34}$$

The numerical results of density at time $t = 1.8$ with 400 initially uniform mesh are displayed in Figure 3.3. The Lagrangian framework resolves finer structure in the numerical solutions which are much superior to the Eulerian framework. The ALE result with $u_g = 0.5u$ shows good results for both shocks and density perturbations, and performs as a balanced solver between Eulerian and Lagrangian frameworks as expected.




(a) (b) (c)

FIGURE 3.3: The density results for shock-turbulence interaction test with the present ALE scheme: (a) Eulerian ($u_g = 0$); (b) ALE ($u_g = 0.5u$); (c) Lagrangian ($u_g = u$).

In this section, a direct ALE multi-moment finite volume scheme is developed for one-dimensional Euler equations. It can achieve third order accuracy for smooth solutions and can successfully solve the 1D compressible Euler equations include discontinuous solutions on moving meshes. The Lagrangian framework captures the discontinuities more sharply than the Eulerian framework.

## 3.4 Multi-moment finite volume ALE scheme for multi-dimensional Euler equation

The 1D ALE formulation presented in the previous section can be extended straightforwardly to multi-dimensional case. In the present multi-moment finite volume scheme, we solve the PV at each cell vertex as the computational variable as well in addition to the VIA moment. Since the vertex is moved at a specified velocity, the basic idea developed in the previous section can be applied to update the two kinds of moments in multi-dimensional case.

### 3.4.1 Grid configuration and notations

The original Volume-average/point-value formulation was proposed by Xie [54, 80, 81] for incompressible Navier-Stokes equations on static unstructured grids. Following Xie's work, we give the definition of computational grids. The computational domain is divided into elements $\Omega_i$ $(i = 1, 2, \cdots, \mathbb{I})$ of different types, triangular and quadrilateral elements for 2D and quadrilateral element for 3D (will be used in Chapter 5) as shown in Figure 3.4.

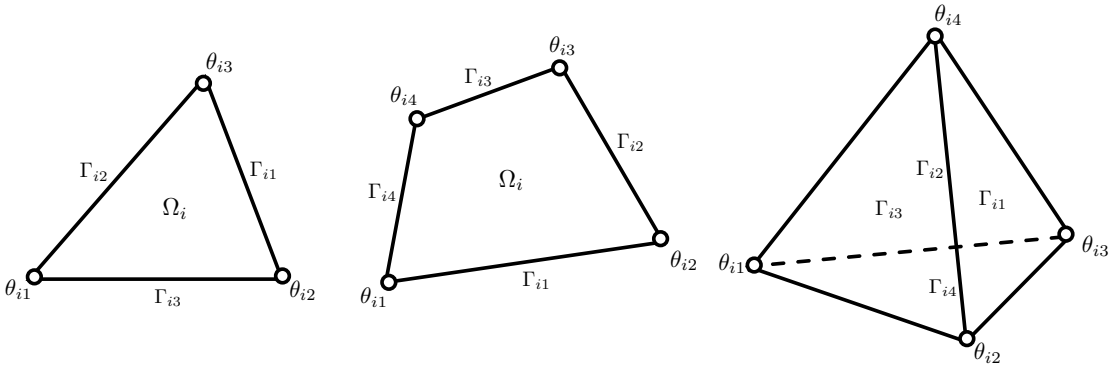



FIGURE 3.4: 2D: triangular (left), quadrilateral (middle) element and 3D: tetrahedral element (right).

The global indices of cell vertices are denoted by $\theta_{\mathfrak{p}}$, $(\mathfrak{p} = 1, 2, \cdots, \mathbb{P})$. To reconstruct the high order polynomials on local cells, the vertices of cell $\Omega_i$ are denoted by $\theta_{ik}$ $(k = 1, 2, \cdots, \mathbb{K})$ with the coordinate $\boldsymbol{x}_{ik} = (x_{ik}, y_{ik}, z_{ik})$ (i.e. $\mathbb{K} = 3$ for triangle, $\mathbb{K} = 4$ for quadrilateral and 3D tetrahedron). The boundary surface segments of element $\Omega_i$ are denoted by $\Gamma_{ij}$ $(j = 1, 2, \cdots, \mathbb{J})$ with subscript $ij$ denoting the index of $j$th cell boundary (i.e. $\mathbb{J} = 3$ for triangle, $\mathbb{J} = 4$ for quadrilateral and tetrahedron). The boundary segments $\Gamma_{ij}$ is actually an edge for 2D and a surface for 3D as shown in

Figure 3.4. For later use, we also denote the area and the outward unit normal vector of boundary segments by $|\Gamma_{ij}|$ and $\mathbf{n}_{ij} = (n_{xij}, n_{yij}, n_{zij})$, and the cell volume and mass center by $V_i = |\Omega_i|$ and $\theta_{ic}(x_{ic}, y_{ic}, z_{ic})$, respectively. The topological relation between the global index of vertices "$\mathfrak{p}$" and its local index "$ik$" in respect of the target cell $\Omega_i$ is realized through a connection table.

In VPM method, two kinds of computational variables, i.e. the volume integrated averages (VIA) over $\Omega_i$ and the point values (PV) at the vertices are defined with respect to a physical variable $\phi(x, y, z, t)$

$$\overline{\phi}_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} \phi(x, y, z, t) d\Omega, \ i = 1, 2, \cdots, \mathbb{I}; \tag{3.35}$$

$$\phi_{ik} = \phi(x_{ik}, y_{ik}, z_{ik}, t), \ k = 1, 2, \cdots, \mathbb{K}, \tag{3.36}$$

where $\phi$ stands for all state variables same as that in 1D.

### 3.4.2 Local reconstruction

For algorithm simplicity, following [54, 55, 80], element $\Omega_i$ is firstly transformed from global coordinate system $x-y-z$ to a canonical referential coordinate system $\xi-\eta-\zeta$ by $\mathbf{x} = \sum_{k=1}^{\mathbb{K}} \mathbf{x}_{ik} \mathcal{N}_{ik}$ as shown in Figure 3.5, where the third dimension is discarded by default for 2D elements. Given an arbitrary quantity $\phi_{ik}$ at a vortex $ik$ of the physical

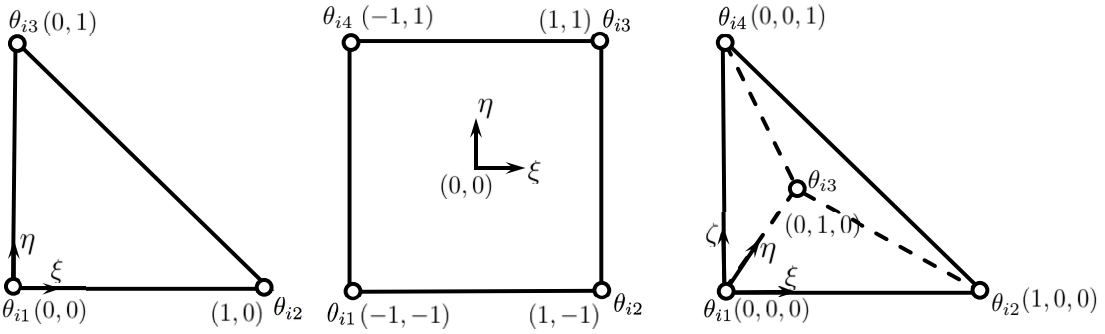



FIGURE 3.5: 2D: triangular (left), quadrilateral (middle) element and 3D: tetrahedral element (right) in local referential coordinate system.

coordinates $(x_{ik}, y_{ik}, z_{ik})$, the corresponding quantity in referential coordinate can be

connected by basis function $\mathcal{N}_{ik}(\boldsymbol{\xi})$ as

$$\begin{bmatrix} 1 \\ x \\ y \\ z \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ x_{i1} & x_{i2} & \cdots & x_{i\mathbb{K}-1} & x_{i\mathbb{K}} \\ y_{i1} & y_{i2} & \cdots & y_{i\mathbb{K}-1} & y_{i\mathbb{K}} \\ z_{i1} & z_{i2} & \cdots & z_{i\mathbb{K}-1} & z_{i\mathbb{K}} \\ \phi_{i1} & \phi_{i2} & \cdots & \phi_{i\mathbb{K}-1} & \phi_{i\mathbb{K}} \end{bmatrix} \begin{bmatrix} \mathcal{N}_{i1}(\boldsymbol{\xi}) \\ \mathcal{N}_{i2}(\boldsymbol{\xi}) \\ \cdots \\ \mathcal{N}_{i\mathbb{K}-1}(\boldsymbol{\xi}) \\ \mathcal{N}_{i\mathbb{K}}(\boldsymbol{\xi}) \end{bmatrix} \tag{3.37}$$

The mapping functions for above three types of elements are listed in Table 3.4. Another

TABLE 3.4: Mapping functions

| | Quadrilateral element | Triangular element | Tetrahedral element |
|---|---|---|---|
| Basis functions | $\mathcal{N}_{i1} = \frac{1}{4}(1-\xi-\eta+\xi\eta)$ | $\mathcal{N}_{i1} = 1-\xi-\eta$ | $\mathcal{N}_{i1} = 1-\xi-\eta-\zeta$ |
| | $\mathcal{N}_{i2} = \frac{1}{4}(1+\xi-\eta-\xi\eta)$ | $\mathcal{N}_{i2} = \xi$ | $\mathcal{N}_{i2} = \xi$ |
| | $\mathcal{N}_{i3} = \frac{1}{4}(1+\xi+\eta+\xi\eta)$ | $\mathcal{N}_{i3} = \eta$ | $\mathcal{N}_{i3} = \eta$ |
| | $\mathcal{N}_{i4} = \frac{1}{4}(1-\xi+\eta-\xi\eta)$ | | $\mathcal{N}_{i4} = \zeta$ |

important issue is regarding to the transformation of first order derivative between two coordinate systems, which is given as

$$\begin{bmatrix} \phi_{\xi i} \\ \phi_{\eta i} \\ \phi_{\zeta i} \end{bmatrix} = \begin{bmatrix} x_{\xi i} & y_{\xi i} & z_{\xi i} \\ x_{\eta i} & y_{\eta i} & z_{\eta i} \\ x_{\zeta i} & y_{\zeta i} & z_{\zeta i} \end{bmatrix} \begin{bmatrix} \phi_{xi} \\ \phi_{yi} \\ \phi_{zi} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \phi_{xi} \\ \phi_{yi} \\ \phi_{zi} \end{bmatrix} = \begin{bmatrix} \xi_{xi} & \eta_{xi} & \zeta_{xi} \\ \xi_{yi} & \eta_{yi} & \zeta_{yi} \\ \xi_{zi} & \eta_{zi} & \zeta_{zi} \end{bmatrix} \begin{bmatrix} \phi_{\xi i} \\ \phi_{\eta i} \\ \phi_{\zeta i} \end{bmatrix} \tag{3.38}$$

As detailed in [54, 55, 80], the first-order and second-order derivatives of the physical field are firstly computed in the physical coordinates and then transformed to the local coordinates. With the supplementary of the first and second order derivatives at cell center $(\phi_{\xi ic}, \phi_{\eta ic}, \phi_{\zeta ic})$ and $(\phi_{\xi^2 ic}, \phi_{\eta^2 ic}, \phi_{\zeta^2 ic})$, the reconstruction polynomial $\Phi$ for conservative variable $\mathbf{U} = [\rho, \mathbf{M}, \rho E]$ can be formulated on the local coordinates as

$$\Phi_i(\xi,\eta,\zeta) = c_0 + c_1\xi + c_2\eta + c_3\xi\eta + c_4\xi^2 + c_5\eta^2 + c_6\zeta + c_7\xi\zeta + c_8\eta\zeta + c_9\zeta^2 + c_{10}\xi\eta\zeta \tag{3.39}$$

where the formulation for 2D elements can be described by discarding the third component of coordinate in $\eta$ direction of Eq.(3.39). The coefficients can be obtained from the constrained conditions in term of multi-moments, i.e. VIA $(\overline{\phi}_i)$, PVs $(\phi_{ik})$ and derivatives $(\phi_{\xi ic}, \phi_{\eta ic}, \phi_{\zeta ic})$ at cell center, which are documented in the Appendix B.

Following the limiting procedure in [56, 57], the high order reconstruction is degraded to a linear polynomial by MLP [82] scheme to eliminate spurious oscillations. The

first-order derivatives at vertices are computed from the reconstruction functions in local coordinates and then transformed back to the physical coordinates for updating PV moments.

Given the spatial derivatives of physical field $\mathbf{U}$ at point $\mathfrak{p}$, $\left(\frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial x}, \frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial y}, \frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial z}\right)$, which are calculated from the surrounding cells $\Omega_{\mathfrak{p}m}$ ($m = 1, 2, 3, \cdots, \mathbb{M}$), we approximate the upwind-biased derivatives with respect to $x$, $y$ and $z$ at vertex point $\mathfrak{p}$ by

$$\frac{\partial \mathbf{U}_{\mathfrak{p}}^{\pm}}{\partial x} = \sum_{m=1}^{4} \omega_{\mathfrak{p}m}^{x\pm} \frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial x}, \frac{\partial \mathbf{U}_{\mathfrak{p}}^{\pm}}{\partial y} = \sum_{m=1}^{4} \omega_{\mathfrak{p}m}^{y\pm} \frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial y} \text{ and } \frac{\partial \mathbf{U}_{\mathfrak{p}}^{\pm}}{\partial z} = \sum_{m=1}^{4} \omega_{\mathfrak{p}m}^{z\pm} \frac{\partial \mathbf{U}_{\mathfrak{p}m}}{\partial z} \tag{3.40}$$

respectively. The weights for derivatives on left and right sides of point $\mathfrak{p}$ in $x$ and $y$ directions are calculated by

$$\omega_{\mathfrak{p}m}^{\tau\pm} = \frac{\max(0, \mathbf{n}^{\tau\pm} \cdot \overrightarrow{\theta_{\mathfrak{p}mc}\theta_{\mathfrak{p}}})}{\sum_{m=1}^{\mathbb{M}} \max(0, \mathbf{n}^{\tau\pm} \cdot \overrightarrow{\theta_{\mathfrak{p}mc}\theta_{\mathfrak{p}}})}, \tag{3.41}$$

where $\overrightarrow{\theta_{\mathfrak{p}mc}\theta_{\mathfrak{p}}}$ denotes the vector from the mass center $\theta_{\mathfrak{p}mc}$ to vertex $\theta_{\mathfrak{p}}$, and the unit normal vector $\mathbf{n}^{\tau\pm}$ represents $\mathbf{n}^{x\pm}(\pm 1, 0, 0)$, $\mathbf{n}^{y\pm}(0, \pm 1, 0)$ and $\mathbf{n}^{z\pm}(0, 0, \pm 1)$ respectively.

### 3.4.3 Solution for VIA moment

We formulate the semi-discrete finite volume scheme (3.1) in 2D with a Q-point Gaussian quadrature for numerical fluxes as

$$\frac{d}{dt}\left(\overline{\mathbf{U}}_i V_i\right) = -\sum_{j=1}^{\mathbb{J}} \sum_{q=1}^{Q} \omega_q \tilde{\mathbf{F}}_{ij}(G_q, t) |\Gamma_{ij}|, \tag{3.42}$$

where $G_q$ is the Gaussian quadrature point on cell edge $\Gamma_{ij}$ which separates cells $\Omega_i$ and $\Omega_{ij}$. The four-point Gauss-Lobatto integral formula is used in this work, which reads $G_1 = P_1, G_2 = \frac{1}{2}(P_1 + P_2) - \frac{\sqrt{5}}{10}(P_2 - P_1), G_3 = \frac{1}{2}(P_1 + P_2) + \frac{\sqrt{5}}{10}(P_2 - P_1), G_4 = P_2$, and $\omega_1 = \omega_4 = \frac{1}{12}, \omega_2 = \omega_3 = \frac{5}{12}$ for the edge with endpoints $P_1$ and $P_2$. The HLLC scheme on moving grids [83] is implemented in this study to calculate flux $\tilde{\mathbf{F}}_{ij}$ on edge $\Gamma_{ij}$, as shown in Appendix C.3.

### 3.4.4 Solution for PV moment

We solve Eq.(3.6) point-wisely by Roe's Riemann solver as

$$\begin{aligned}\frac{\partial \mathbf{W}_{\mathfrak{p}}}{\partial t} &= -\frac{1}{2}\left(\tilde{\boldsymbol{A}}\left(\frac{\partial \mathbf{W}_{\mathfrak{p}}^{-}}{\partial x}+\frac{\partial \mathbf{W}_{\mathfrak{p}}^{+}}{\partial x}\right)+\tilde{\boldsymbol{R}}_{\boldsymbol{A}}|\tilde{\boldsymbol{\Lambda}}_{\boldsymbol{A}}|\tilde{\boldsymbol{L}}_{\boldsymbol{A}}\left(\frac{\partial \mathbf{W}_{\mathfrak{p}}^{-}}{\partial x}-\frac{\partial \mathbf{W}_{\mathfrak{p}}^{+}}{\partial x}\right)\right)\\ &\quad -\frac{1}{2}\left(\tilde{\boldsymbol{B}}\left(\frac{\partial \mathbf{W}_{\mathfrak{p}}^{-}}{\partial y}+\frac{\partial \mathbf{W}_{\mathfrak{p}}^{+}}{\partial y}\right)+\tilde{\boldsymbol{R}}_{\boldsymbol{B}}|\tilde{\boldsymbol{\Lambda}}_{\boldsymbol{B}}|\tilde{\boldsymbol{L}}_{\boldsymbol{B}}\left(\frac{\partial \mathbf{W}_{\mathfrak{p}}^{-}}{\partial y}-\frac{\partial \mathbf{W}_{\mathfrak{p}}^{+}}{\partial y}\right)\right)\end{aligned} \tag{3.43}$$

where the matrices are approximated from the Roe-averaging values of the surrounding VIAs by

$$\tilde{\phi}_{\mathfrak{p}}=\frac{\sum_{m=1}^{\mathbb{M}}\sqrt{\overline{\rho}_{\mathfrak{p}m}}\cdot\overline{\phi}_{\mathfrak{p}m}}{\sum_{m=1}^{\mathbb{M}}\sqrt{\overline{\rho}_{\mathfrak{p}m}}} \tag{3.44}$$

and the sound speed is calculated as

$$\tilde{a}_{\mathfrak{p}}=\left((\gamma-1)(\tilde{H}_{\mathfrak{p}}-\frac{1}{2}(\tilde{u}_{\mathfrak{p}}^{2}+\tilde{v}_{\mathfrak{p}}^{2}))\right)^{\frac{1}{2}}. \tag{3.45}$$

Eigenstructures are detailed in Appendix A.2. The spatial derivatives of primitive variables $\left(\frac{\partial \mathbf{W}_{\mathfrak{p}}^{\pm}}{\partial x},\frac{\partial \mathbf{W}_{\mathfrak{p}}^{\pm}}{\partial y}\right)$ can be obtained from the spatial derivatives of the conservative variables $\left(\frac{\partial \mathbf{U}_{\mathfrak{p}}^{\pm}}{\partial x},\frac{\partial \mathbf{U}_{\mathfrak{p}}^{\pm}}{\partial y}\right)$ in analogy to (3.28) and (3.29).

The overshoots/undershoots of PV can be simply removed by

$$\phi_{\mathfrak{p}}=\min\left(\phi_{\mathfrak{p}},\max(\overline{\phi}_{\mathfrak{p}m})\right),\quad \phi_{\mathfrak{p}}=\max\left(\phi_{\mathfrak{p}},\min(\overline{\phi}_{\mathfrak{p}m})\right), \tag{3.46}$$

where $\max(\overline{\phi}_{\mathfrak{p}m})$ and $\min(\overline{\phi}_{\mathfrak{p}m})$ are the maximum and minimum VIAs on the surrounding cells sharing node $\theta_{\mathfrak{p}}$.

### 3.4.5 Geometrical conservation law

For a uniform flow, the spatial gradients of all flow variables are zero, thus Equation (3.42) becomes

$$\frac{dV_i}{dt}=\sum_{j=1}^{\mathbb{J}}\sum_{q=1}^{Q}\omega_q\boldsymbol{u}_g(G_q,t)\cdot\mathbf{n}|\Gamma_{ij}|, \tag{3.47}$$

which is the semi-discretization of geometrical conservation law (GCL). Here, $\boldsymbol{u}_g(G_q,t)$ is the grid velocity at Gaussian point $G_p$ on edge $\Gamma_{ij}$. For an arbitrarily varying mesh, we can suppose that the grid velocity has a linear distribution over cell edges. Given the grid velocities at two ends of boundary segments $\Gamma_{ij}$ denoted as $\boldsymbol{u}_{gij0}$ and $\boldsymbol{u}_{gij1}$,

Equation (3.47) can be simplified as

$$\frac{dV_i}{dt} = \sum_{j=1}^{\mathbb{J}} \overline{\boldsymbol{u}}_{gij} \cdot \mathbf{n} |\Gamma_{ij}|, \tag{3.48}$$

where the edge-averaged velocity is given as

$$\overline{\boldsymbol{u}}_{gij} = \frac{1}{2}(\boldsymbol{u}_{gij0} + \boldsymbol{u}_{gij1}). \tag{3.49}$$

It is noted that, by the assumption that the cell edge moves as a solid body, Equation (3.48) might not truly realize the pure Lagrangian framework for deformational flows because the cell boundaries do not move with the local fluid in the same accuracy order of the third order spatial discretizations. As shown in [37], curvilinear meshes are necessary for purely Lagrangian scheme higher than second order. Nevertheless, by assuming cell boundaries always move as a linear edge and then computing the numerical fluxes over this linear edge, it is still possible to achieve third order accuracy for the ALE framework. We call the framework that only moves cell vertices with the local fluid as quasi-Lagrangian framework in this research to distinguish it from the purely Lagrangian framework.

### 3.4.6 Time step limitation

Time marching for all the semi-discrete time evolution equations (4.5), (4.7) and (3.48) in this section is implemented by third order Runge-Kutta scheme similar in 3.2.4. Following [32, 84], a classical CFL stability condition and a geometrical time step limitation are both considered for a suitable time step.

At time $t^n$, the classical CFL time step is given as

$$\Delta t_E = C_E \min_{1 \leqslant i \leqslant \mathbb{I}} \frac{\lambda_i}{\sqrt{u_i^2 + v_i^2} + a_i}, \tag{3.50}$$

where $\lambda_i$ is the in-circle diameter of cell $i$ and $a_i$ is the sound speed in the cell. We set the CFL number $C_E = 0.3$ for later simulation.

The second criterion limits the variation rate of the element volume in one time step, which is given as

$$\Delta t_V = C_V \min_{1 \leqslant i \leqslant \mathbb{I}} \frac{|S_i^n|}{|S_i'|}, \tag{3.51}$$

where

$$S_i' = \sum_{j=1}^{\mathbb{J}} \overline{\boldsymbol{u}}_{gij} \cdot \mathbf{n} |\Gamma_{ij}|, \tag{3.52}$$

and $C_V$ is set as $0.1$.

Finally, the interval for time stepping is

$$\Delta t = \min\left(\Delta t_E, \Delta t_V, C_M \Delta t^n\right), \tag{3.53}$$

where $C_M$ is set as $1.1$ in this paper.

## 3.5 Benchmark tests in 2D

In this section, the multi-moment ALE method is verified with some 2D benchmark tests. For the Lagrangian framework, the grid velocity is simply calculated from the Roe average of the surrounding VIAs as shown in (4.8) with $\phi$ representing the fluid velocity $\boldsymbol{u}$. As shown in the 1D cases, the Lagrangian and Eulerian frameworks have been evaluated as two special cases of the general ALE formulations. We also assessed the performance of the ALE scheme on moving meshes with specified nodal velocities.

### 3.5.1 Isentropic vortex

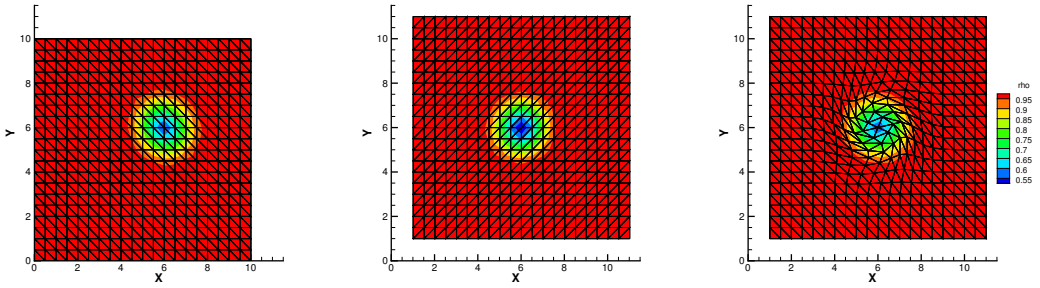
Advection of an isentropic vortex [85] is computed to examine the convergence rate of the present ALE scheme. we computed this problem on a computational domain of $[0, 10] \times [0, 10]$ with the initial conditions specified as

$$(\rho_0, u_0, v_0, p_0) = (1 + \delta\rho, 1 + \delta u, 1 + \delta v, 1 + \delta p), \tag{3.54}$$

where $\delta\rho, \delta u, \delta v, \delta p$ are the initial perturbations to the primitive variables, which are centered in the computational domain and given by

$$\begin{aligned}
\delta\rho &= (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \\
\delta u &= -\tfrac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} (y - 5), \\
\delta v &= \tfrac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} (x - 5), \\
\delta p &= (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1,
\end{aligned} \tag{3.55}$$

where the vortex strength $\epsilon = 5$, the temperature perturbation $\delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}e^{1-r^2}$ and $r^2 = (x-5)^2 + (y-5)^2$. The periodic boundary conditions were applied to the boundaries of the computational domain for Eulerian and Lagrangian framework and zero gradient boundary condition is set for the following specific ALE framework.




(a) (b) (c)

FIGURE 3.6: The density results for 2D isentropic vortex solved by the present ALE scheme in triangular mesh. (a) Eulerian ($u_g = 0, v_g = 0$); (b) ALE ($u_g = 1 - \pi\tilde{y}/2, v_g = 1 + \pi\tilde{x}/2$); (c) Quasi-Lagrangian ($u_g = u, v_g = v$). It is noted that the coordinate axes are fixed, while the meshes in (b) and (c) move.

We computed this problem to time $t = 1$ using the Eulerian ($u_g = 0, v_g = 0$), Lagrangian ($u_g = u, v_g = v$) and ALE ($u_g = 1 - \pi\tilde{y}/2, v_g = 1 + \pi\tilde{x}/2$) frameworks respectively. In the ALE framework, where $\tilde{x} = (x - (5 + t))$ and $\tilde{y} = (y - (5 + t))$, the mesh moves at a velocity field that combines a translational motion $(1, 1)$ and a rigid-rotational motion around the vortex center with an angular speed of $\pi/2$. Up to $t = 1$, the mesh is moved over a distance of 1 in both $x$ and $y$ directions, and rotated by $90^o(\pi/2)$ as shown in Figure 3.6(b).

The density fields of numerical solutions using different mesh-moving strategies are shown in Figure 3.6. The exact solution of the Euler equations from the particular initial conditions is the pure translation of the vortex in the diagonal direction of the computational domain. Our numerical experiments started with Cartesian grids of gradually refined resolutions. The Eulerian framework remains the initial Cartesian grids fixed, while the grids nodes were moved at specified velocities in ALE and Quasi-Lagrangian cases.

To quantify the spatial discretization error, we define $L_1$ and $L_\infty$ error as follows,

$$L_1 = \frac{1}{N}\sum_{i=1}^{N}|\phi_{ni} - \phi_{ei}|, \quad L_\infty = \max_{1\leqslant i\leqslant N}|\phi_{ni} - \phi_{ei}|, \tag{3.56}$$

TABLE 3.5: Errors and convergence rates of density for 2D isentropic vortex in triangular mesh

| cells | Eulerian | | | | ALE | | | | Quasi-Lagrangian | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate |
| 800 | 8.28e-4 | - | 1.79e-2 | - | 6.11e-4 | - | 8.43e-3 | - | 5.91e-4 | - | 9.69e-3 | - |
| 3200 | 1.24e-4 | 2.73 | 2.99e-3 | 2.58 | 8.66e-5 | 2.82 | 1.89e-3 | 2.15 | 7.63e-5 | 2.95 | 1.26e-3 | 2.95 |
| 12800 | 1.59e-5 | 2.97 | 3.81e-4 | 2.97 | 1.13e-5 | 2.94 | 2.79e-4 | 2.76 | 9.04e-6 | 3.08 | 1.38e-4 | 3.19 |
| 51200 | 2.00e-6 | 2.99 | 4.66e-5 | 3.03 | 1.68e-6 | 2.75 | 3.97e-5 | 2.81 | 1.09e-6 | 3.05 | 1.56e-5 | 3.15 |

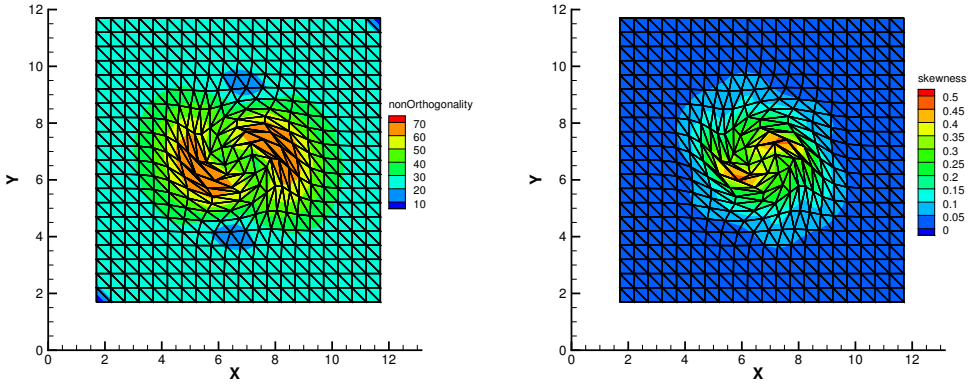
where $\phi_{ni}$ and $\phi_{ei}$ denote the numerical and exact solutions in cell $i$ respectively. For a converging numerical formulation, $L_1$ and $L_\infty$ error reduce with the grid refinement as $N$ is increased. The rate of error reduction, i.e. the convergence rate, is related to the leading term of the truncation error for a specific spatial discretization algorithm. In practice, we evaluate the convergence rate as

$$\mathcal{O}_L = \frac{\ln(L^{(1)}/L^{(2)})}{\ln\sqrt{N^{(2)}/N^{(1)}}}, \tag{3.57}$$

where $L^{(1)}$ and $L^{(2)}$ represent $L_1$ or $L_\infty$ errors for two successive refined grids with different resolutions quantified by cell numbers $N^{(1)}$ and $N^{(2)}$ respectively.

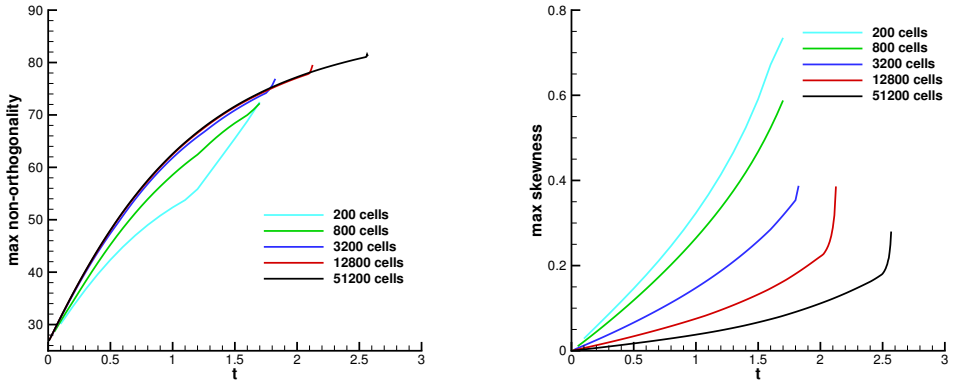
We measured the $L_1$ and $L_\infty$ errors of the numerical experiments, which are shown with the convergence rates in Table 3.5. We observe that a uniformly third order accuracy is achieved for all frameworks. It also demonstrates that higher than second order accuracy can be achieved for linear edge element by ALE scheme.

To explore the worst mesh quality the present scheme can work on, we further run the quasi-Lagrangian case until the program stops. The mesh quality is evaluated by non-orthogonality and skewness, which are defined in Appendix D. As presented in Figure 3.7, the degrees of non-orthogonality and skewness in the vortex region reach to high values before the program ceases. We plot the time histories of mesh non-orthogonality and skewness for different mesh resolutions in Figure 3.8. With time evolves, the mesh quality becomes worse. It is observed that the the program can run a longest time for the finest mesh where the maximum degree of non-orthogonality is $81^o$. The robustness of present scheme mostly depends on the mesh non-orthogonality while the mesh skewness does not have much restriction. Figure 3.9 shows the time history of $L1$ and $L_\infty$ errors in the case with 800 cells, which indicates that the numerical error increases with the deterioration of mesh quality and the error sharply rises right before the program stops. The mesh quality at the corner point (e.g. t=1.5 in Figure 3.9) is chosen as the worst mesh quality the present scheme can endure. We collect the values of maximum non-orthogonality and skewness for the five mesh resolutions at the




(a) (b)

FIGURE 3.7: Mesh non-orthogonality and skewness for quasi-Lagrangian case at time t=1.7 in the case with 800 cells.




(a) (b)

FIGURE 3.8: Time history of mesh non-orthogonality (a) and skewness (b) with different mesh resolutions.

limit situation, and plot the values on the plane 3.10. We can roughly conclude that the present scheme can endure the mesh deformation to the maximum non-orthogonality of around $70^o$.

### 3.5.2 Free stream preservation

Free stream preservation is a common test to check the geometrical conservation law for moving mesh schemes. The computational domain and mesh are set as the same as in section 3.5.1. The initial conditions are uniform for all physical fields with $(\rho_0, u_0, v_0, p_0) = (1, 1, 1, 1)$.

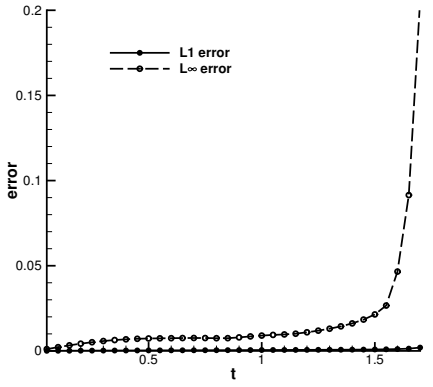



FIGURE 3.9: Time history of $L1$ and $L_\infty$ errors in the case with 800 cells.

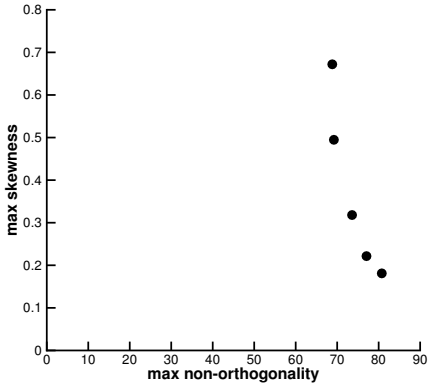



FIGURE 3.10: Maximum mesh non-orthogonality and skewness plane.

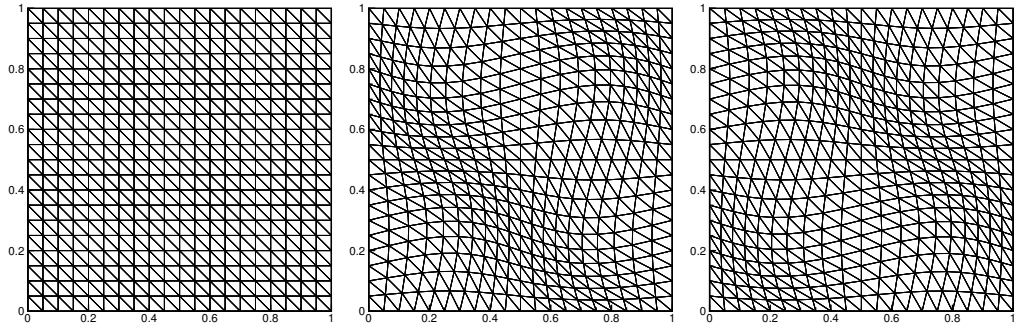
We perform the calculation up to time $t = 1$ in moving mesh where the grid velocity $\mathbf{u}_g = (u_g, v_g)$ is prescribed by

$$u_g(\mathbf{x}, t) = X_0 \cdot 2\pi n_t / t_0 \cos(2\pi n_t t / t_0) \sin(2\pi n_x x / L_x) \sin(2\pi n_y y / L_y), \tag{3.58a}$$

$$v_g(\mathbf{x}, t) = Y_0 \cdot 2\pi n_t / t_0 \cos(2\pi n_t t / t_0) \sin(2\pi n_x x / L_x) \sin(2\pi n_y y / L_y), \tag{3.58b}$$

with parameters given as $X_0 = 1, Y_0 = 1, n_t = 4, n_x = 1, n_y = 1, t_0 = 2, L_x = 10, L_y = 10$. The mesh configurations are plotted in Fig. 3.11 to show the large mesh motions at different time instants.

We plot the $L_1$ and $L_\infty$ numerical errors of density for meshes with different element sizes as shown in Figure 3.12. The numerical errors for all mesh resolutions are of machine precision, which justifies the exact geometrical conservation of the proposed method.



(a) t=0 (b) t=0.15 (c) t=0.35

FIGURE 3.11: The mesh configurations at different time instants.

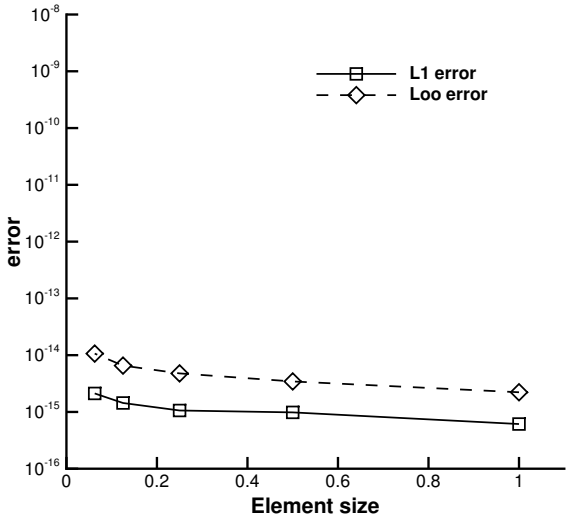



FIGURE 3.12: $L_1$ and $L_\infty$ errors of density for meshes with different element sizes.

### 3.5.3 Sod problem

We solved the well-known Sod shock tube problem on 2D meshes within the computational domain $[0, 1] \times [0, 0.1]$. The initial conditions are

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (1, 0, 0, 1), & \text{for } 0 \leqslant x \leqslant 0.5, \\ (0.125, 0, 0, 0.1), & \text{for } 0.5 < x \leqslant 1, \end{cases} \tag{3.59}$$

which will generate rarefaction wave, contact discontinuity and shock wave.

The computational domain is partitioned into $100 \times 10$ initially uniform quadrilateral elements. We computed the problem to time $t = 0.25$ for the Eulerian, Lagrangian and ALE framework. The grid for Lagrangian frame at time $t = 0.25$ is shown in Figure 3.13. Figure 3.14 shows the density results computed by the present scheme on

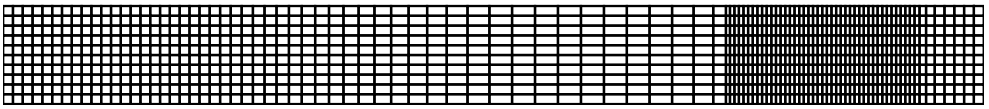
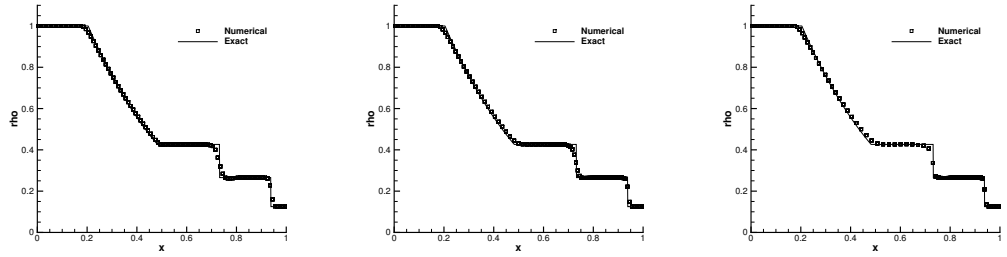


FIGURE 3.13: 2D Sod problem grid at time $t = 0.25$ for Lagrangian framework.

different frameworks. We can find that the Lagrangian framework gets more accurate results around contact discontinuity than the Eulerian framework.




(a) (b) (c)

FIGURE 3.14: The density profiles of the results for Sod problem with different mesh configurations: (a) Eulerian ($u_g = 0$); (b) ALE ($u_g = 0.5u$); (c) Lagrangian ($u_g = u$).

### 3.5.4 Sedov problem

We tested the present scheme with the Sedov problem, which is a circular blast wave generated from a point source at the origin of the Cartesian grid. The exact solution of this problem is available based on the self-similarity arguments [86]. We computed this problem on a quarter of the physical domain, $[0, 1.2] \times [0, 1.2]$, with $30 \times 30$ initially uniform quadrilateral elements. The initial conditions are given by $(\rho_0, u_0, v_0, p_0) = (1, 0, 0, 10^{-6})$ except in the first zone, where an initial delta-function energy source was set in terms of pressure by

$$p_{\text{or}} = (\gamma - 1)\rho_0 \frac{\mathcal{E}_0}{V_{\text{or}}}, \tag{3.60}$$

where $V_{\text{or}}$ denotes the volume of the cell containing the origin and $\mathcal{E}_0$ is the total amount of released energy which was set as $\mathcal{E}_0 = 0.244816$ in our numerical tests. Reflective boundary conditions were imposed on the left and lower boundaries of the computational domain. Numerical tests were carried out up to time $t = 1$. As discussed in [86], the solution consists of an axis-symmetrically diverging shock whose front is located at radius $R = 1$ at time $t = 1$ with the peak density reaching the value of 6.

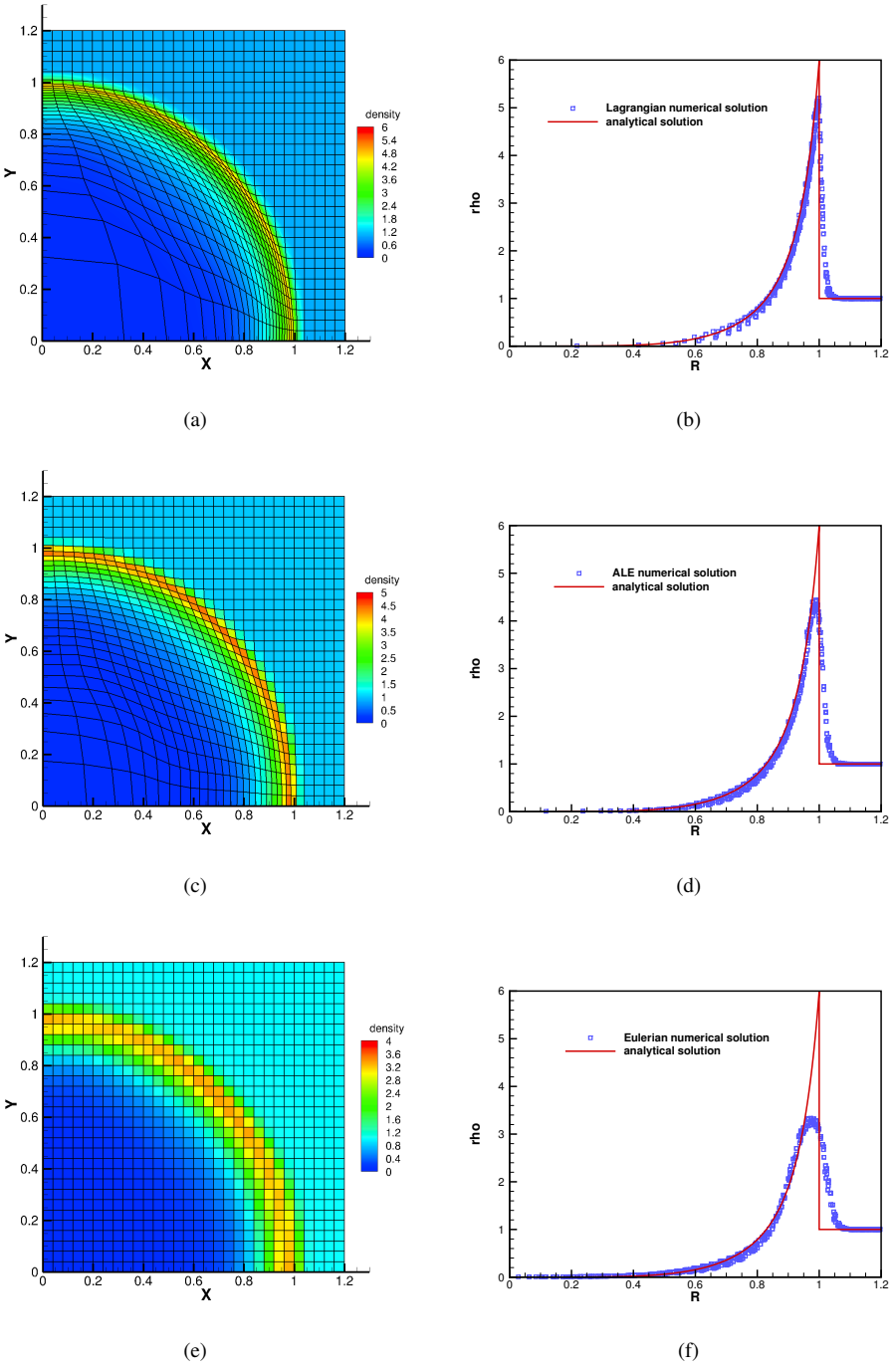



FIGURE 3.15: Mesh and density results of Sedov problem with present scheme: (a)(b) Lagrangian ($\boldsymbol{u}_g = \boldsymbol{u}$); (c)(d) ALE ($\boldsymbol{u}_g = 0.7\boldsymbol{u}$); (e)(f) Eulerian ($\boldsymbol{u}_g = 0$).

We computed this test case in Lagrangian, Eulerian and ALE frameworks respectively, with the grid velocity being $0.7$ of the fluid velocity for ALE framework. Figure 3.15 presents the density results for different frameworks, where panels (a), (c) and (e) show the density distribution over the computational domain as well as the computational meshes, and panels (b), (d) and (f) compare the profiles of density along the radial directions with the exact solution. The Lagrangian solution is the most accurate among the three frameworks, which well captures the discontinuity and the peak value.

## 3.6 Short summary

In this chapter, we presented a multi-moment finite volume ALE scheme for Euler equations. In the multi-moment finite volume method, both the volume integrated averages (VIA) and the point values (PV) at the vertices of mesh cells, which are treated as prognostic variables and updated simultaneously, are used for high-order reconstructions. The governing equations are written with respect to the conservative variables in the integral form to update the VIA, while non-conservative equations of the primitive variables in the differential form are used to update the PV moment. The PVs are computed point-wisely with Roe Riemann solver using the spatial derivatives of primitive variables, while the VIAs are updated by the finite volume formulation of flux form where the numerical fluxes are calculated directly from the cell boundary PVs in 1D case and from the high-order multi-moment reconstructions in 2D case. Third order Runge-Kutta scheme is implemented for time marching. The mesh is moved by giving specified velocities to the mesh vertices, which results in a simple and efficient straight-line formulation for ALE computations.

The proposed direct ALE multi-moment finite volume formulation provides a high-order and efficient ALE computational model for inviscid compressible flows. The numerical verifications show that the present model can achieve 3rd-order convergence rate in 1D and 2D accuracy tests though the cell edges of moving mesh remain as straight lines in 2D. From the accuracy test in 1D, we can find that, with same mesh resolution, numerical errors of present scheme reduce to $15 \sim 20\%$ of a conventional 3rd-order finite volume scheme no matter mesh moves or not. On the other hand, computational cost only increases $20 \sim 25\%$, which is not substantial compared with the reduction of numerical errors. In multi-dimensional moving mesh computation, the numerical resolution and robustness can be affected by mesh quality. The proposed scheme can endure the mesh deformation to the extend of maximum non-orthogonality

of around $70^o$. Moreover, the geometrical conservation law is well satisfied for a uniform flow on arbitrarily moving mesh, which may play the role for improving the numerical accuracy and stability. Besides, numerical tests also show that the present scheme can resolve the inviscid compressible flows with discontinuities well, where the Lagrangian framework, as a particular case of the ALE model, has superior capability to track contact discontinuities.

# Chapter 4

# Viscous compressible flow involved with forced moving body

The application of CFD often encounters fluid flows involving geometrically varying boundaries, such as fluid-structure interaction (FSI). In some cases, the moving boundaries might be driven by imposed external effects, such as flapping wings. Thus, the problem can be simplified by just concerning a one-way coupling, where the fluid state is affected by a given deforming boundary. In this chapter, we will present the numerical model for viscous compressible flows involved with forced moving body. Multi-moment finite volume scheme for Euler equations presented in Chapter 3 is directly extended to the viscous flows, while the PV moments are calculated from the differential equations in terms of conservative variables. For practical applications, the whole computational mesh of fluid domain should be moved with the moving boundaries. To this end, a simple moving mesh strategies, radial basis function (RBF) interpolation, is implemented to transfer the movement of boundary to internal mesh points. At last, numerical tests are calculated to verify the capability of the present solver.

## 4.1 Governing equations

### 4.1.1 Integral form

As introduced in Chapter 2, We reformulate the ALE integral form of viscous compressible Navier-Stoke equations as

$$\frac{\partial}{\partial t}\int_{\Omega_{\boldsymbol{\chi}}(t)} \mathbf{U} d\Omega + \int_{\Gamma(t)} \mathbf{F}(\mathbf{U}) d\Gamma - \int_{\Gamma(t)} \mathbf{G}(\mathbf{U}, \nabla \mathbf{U}) d\Gamma = 0, \tag{4.1}$$

with the vectors of convective flux and viscous flux respectively. In this formulation, the vector of conservative variables and fluxes taking the form as follows

$$\mathbf{U} = \begin{pmatrix} \rho \\ \mathbf{M} \\ \rho E \end{pmatrix}, \mathbf{F}(\mathbf{U}) = \begin{pmatrix} (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\rho \\ (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\mathbf{M} + p \cdot \mathbf{n} \\ (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \mathbf{n}\rho E + p\boldsymbol{u} \cdot \mathbf{n} \end{pmatrix}, \mathbf{G}(\mathbf{U}, \nabla\mathbf{U}) = \begin{pmatrix} 0 \\ \bar{\bar{\tau}} \cdot \mathbf{n} \\ \bar{\bar{\tau}} \cdot \mathbf{n}\boldsymbol{u} - \mathbf{Q} \cdot \mathbf{n} \end{pmatrix}. \tag{4.2}$$

The geometrical conservation law (GCL) is given as

$$\frac{dV}{dt} - \int_{\Gamma(t)} \mathbf{u}_g \cdot \mathbf{n} d\Gamma = 0. \tag{4.3}$$

All variables, closure equations and notations here are same as Chapter 2.

### 4.1.2 Differential form

Since the differential equation with respect to primitive variables is indirect for solving viscous conservative Navier-Stokes equations, we cast Eq.(2.25) into the standard quasi-linear form in 2D based on ***Form* 2** as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\chi} + \boldsymbol{A}\frac{\partial \mathbf{U}}{\partial x} + \boldsymbol{B}\frac{\partial \mathbf{U}}{\partial y} - \nabla \cdot (\mathcal{G}(\mathbf{U}, \nabla\mathbf{U})) = 0, \tag{4.4}$$

which is just the extension of Eq.(3.9) with the diffusive term. The coefficient matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined in Eq.(3.10) and Eq.(3.11) with eigenstructures given in Appendix A.3.

Since the grid velocity $\boldsymbol{u}_g$ appears in both integral and differential formulations, the system can be discretized on arbitrarily moving grids.

## 4.2 Numerical model for viscous compressible Navier Stokes equations

### 4.2.1 Solution for VIA moment

We formulate the semi-discrete form for Equation (4.1) with a Q-point Gaussian quadrature for surface convective fluxes and a relatively simpler integration for viscous fluxes

as

$$\frac{\partial}{\partial t}\left(\overline{\mathbf{U}}_i V_i\right) = -\sum_{j=1}^{\mathbb{J}}\sum_{q=1}^{Q}\omega_q \tilde{\mathbf{F}}_{ij}(G_q,t)|\Gamma_{ij}| + \sum_{j=1}^{\mathbb{J}}\tilde{\mathbf{G}}_{ij}(t)|\Gamma_{ij}|, \tag{4.5}$$

where $\overline{\mathbf{U}}_i$ denotes the volume integrated average (VIA) of conservative variables on cell $\Omega_i$ with volume $V_i$, $G_q$ is the Gaussian quadrature point on cell edge $\Gamma_{ij}$ which separates cells $\Omega_i$ and $\Omega_{ij}$. Two-points Gaussian quadrature formula is used in this work, which reads $G_1 = (\frac{1}{2}+\frac{\sqrt{3}}{6})P_1+(\frac{1}{2}-\frac{\sqrt{3}}{6})P_2, G_2 = (\frac{1}{2}-\frac{\sqrt{3}}{6})P_1+(\frac{1}{2}+\frac{\sqrt{3}}{6})P_2$ and $\omega_1 = \omega_2 = \frac{1}{2}$ for the edge with endpoints $P_1$ and $P_2$.

Various Riemann solvers for calculating the inviscid flux $\tilde{\mathbf{F}}_{ij}$ on edge $\Gamma_{ij}$ are implemented in this research, such as the robust Lax-Friedrich scheme, HLL scheme [31] and the less diffusive HLLC scheme [83]. HLLC scheme is mainly adopted in this chapter. Lax-Friedrich and HLL schemes are employed to problems involving strong shock waves to stabilize the computation. These Riemann solvers on moving mesh are documented in Appendix C.

The viscous numerical fluxes are evaluated based on the values $\mathbf{U}_{ij}$ and $(\nabla\mathbf{U})_{ij}$ of conservative variables on cell edge segment $\Gamma_{ij}$. Thus, we formulate the viscous flux $\tilde{\mathbf{G}}_{ij}$ on the edge $\Gamma_{ij}$ as

$$\tilde{\mathbf{G}}_{ij} = \mathbf{G}_{ij}\left(\frac{1}{2}(\overline{U}^{-}_{\Gamma_{ij}} + \overline{U}^{+}_{\Gamma_{ij}}), (\nabla\mathbf{U})_{\Gamma_{ij}}\right), \tag{4.6}$$

where $\overline{U}^{-}_{\Gamma_{ij}}$ and $\overline{U}^{+}_{\Gamma_{ij}}$ are reconstructed line-averaged values of two neighbouring cells respectively, $(\nabla\mathbf{U})_{\Gamma_{ij}}$ is linearly interpolated from the spatial derivatives in the cell center of the two neighbouring cells.

### 4.2.2 Solution for PV moment

We solve Equation (4.4) point-wisely by Roe's Riemann solver [30] for convective term and $\mathcal{TEC}$ (Time-evolution Converting) formulation [87] for viscous term as

$$\begin{aligned}\frac{\partial \mathbf{U}_{\mathfrak{p}}}{\partial t} &= -\frac{1}{2}\left(\tilde{\mathbf{A}}\left(\frac{\partial \mathbf{U}^{-}_{\mathfrak{p}}}{\partial x}+\frac{\partial \mathbf{U}^{+}_{\mathfrak{p}}}{\partial x}\right)+\tilde{\mathbf{R}}_{\mathbf{A}}|\tilde{\mathbf{\Lambda}}_{\mathbf{A}}|\tilde{\mathbf{L}}_{\mathbf{A}}\left(\frac{\partial \mathbf{U}^{-}_{\mathfrak{p}}}{\partial x}-\frac{\partial \mathbf{U}^{+}_{\mathfrak{p}}}{\partial x}\right)\right)\\ &\quad -\frac{1}{2}\left(\tilde{\mathbf{B}}\left(\frac{\partial \mathbf{U}^{-}_{\mathfrak{p}}}{\partial y}+\frac{\partial \mathbf{U}^{+}_{\mathfrak{p}}}{\partial y}\right)+\tilde{\mathbf{R}}_{\mathbf{B}}|\tilde{\mathbf{\Lambda}}_{\mathbf{B}}|\tilde{\mathbf{L}}_{\mathbf{B}}\left(\frac{\partial \mathbf{U}^{-}_{\mathfrak{p}}}{\partial y}-\frac{\partial \mathbf{U}^{+}_{\mathfrak{p}}}{\partial y}\right)\right)\\ &\quad +\mathcal{TEC}(\nabla\cdot\mathbf{G}(\mathbf{U})_{\mathfrak{p}m}),\end{aligned} \tag{4.7}$$

where the matrices with tilde are calculated from the Roe-averaging values of the surrounding VIAs as

$$\tilde{\phi}_{\mathfrak{p}} = \frac{\sum_{m=1}^{M} \sqrt{\overline{\rho}_{\mathfrak{p}m}} \cdot \overline{\phi}_{\mathfrak{p}m}}{\sum_{m=1}^{M} \sqrt{\overline{\rho}_{\mathfrak{p}m}}} \tag{4.8}$$

and the sound speed is approximated by

$$\tilde{c}_{\mathfrak{p}} = \left( (\gamma - 1)(\tilde{H}_{\mathfrak{p}} - \frac{1}{2}(\tilde{u}_{\mathfrak{p}}^2 + \tilde{v}_{\mathfrak{p}}^2)) \right)^{\frac{1}{2}}. \tag{4.9}$$

## 4.3 Moving mesh strategies

### 4.3.1 Review of moving mesh techniques

One challenge for solving fluid flows involving moving body by moving mesh method is to maintain effective mesh quality. Over past few decades, there have been several techniques to obtain mesh motion for unstructured grids in CFD. The most popular method may be the spring analogy method [88], where the connection of neighbouring two grid points is represented by a linear spring. However, it was observed lacking robustness on arbitrarily unstructured grids [89]. Other moving mesh techniques relates to solving a PDE to determine the movements of internal mesh displacement given the boundary points movements. Examples are a Laplace equation with constant or variable distance-based diffusion coefficient [90], solid body rotation stress (SBR stress) equation [91] and so on. These methods are straightforward to implement since the existing CFD codes are available. However, it fails in maintaining high mesh quality when the boundary points move with high rotation angles.

In recent years, the radial basis function (RBF) method has drawn much attention as an approach to interpolate scattered data. It is widely used in the practical applications, such as computer graphics, error estimations and fluid-structure interactions. Boer de et al. [64] initially used the RBF interpolation to transfer the given boundary-point displacements to internal-point displacements, which can produce high-quality mesh for even largely deformed configurations. Successive works have been carried out to improve the efficiency of mesh motion in [65, 66]. The RBF interpolation is implemented in this research to determine the displacements of internal mesh points for moving boundary problems.

### 4.3.2 Radial basis function interpolation

The moving velocity $\mathbf{u}_g = \dot{\mathbf{x}}$ specified at the internal points of fluid domain is approximated from the motion of solid boundary by using the radial basis function (RBF) interpolation [64, 65, 92]. The interpolation function $s(\mathbf{x})$ is evaluated at structural node position given by a sum of basis functions

$$s(\mathbf{x}) = \sum_{j=1}^{n_b} \boldsymbol{\gamma}_j \varphi(||\mathbf{x} - \mathbf{x}_{bj}||) + h(\mathbf{x}), \tag{4.10}$$

where $\varphi$ is the radial basis function, $\mathbf{x}_{bj}$ the location of control points, $n_b$ its total number, $\boldsymbol{\gamma}_j$ the coefficients of basis function and $||\cdot||$ denotes the Euclidean norm. Since the fluid-structure interface does not change, we use a linear polynomial to recover the translation and rotation of rigid body motions. Thus the $h(\mathbf{x})$ in (4.10) is constructed from the vector of monomials $\mathbf{q} = [1, x, y, z]$ as

$$h(\mathbf{x}) = \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 x + \boldsymbol{\beta}_3 y + \boldsymbol{\beta}_4 z \tag{4.11}$$

where $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3$ and $\boldsymbol{\beta}_4$ are to-be-determined coefficients.

Given the displacements of control points $\Delta\mathbf{x}_b$, we can impose the constraint conditions

$$s(\mathbf{x}_{bj}) = \Delta\mathbf{x}_{bj} \tag{4.12}$$

and

$$\sum_{j=1}^{n_b} \boldsymbol{\gamma}_j \mathbf{q}_i(\mathbf{x}_{bj}) = \mathbf{0}, \tag{4.13}$$

which leads to the following linear system of equations

$$\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{P}_b \\ \mathbf{P}_b^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{x}_b \\ \mathbf{0} \end{bmatrix}. \tag{4.14}$$

where $\mathbf{M}_{bb}$ is an $n_b \times n_b$ matrix with $\mathbf{M}_{ij} = \varphi(||\mathbf{x}_{bi} - \mathbf{x}_{bj}||)$, $\mathbf{P}_b$ is an $n_b \times 4$ matrix with the row $i$ given by $[1, x_{bi}, y_{bi}, z_{bi}]$, $\mathbf{P}_b^T$ is the transpose matrix of $\mathbf{P}_b$, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ contain all coefficients $\boldsymbol{\gamma}_j$ and $\boldsymbol{\beta}_j$ respectively. The displacements of internal fluid points can then be evaluated by

$$\Delta\mathbf{x}_{int} = s(\mathbf{x}_{int}), \tag{4.15}$$

where $\mathbf{x}_{int}$ denotes the position of internal point.

Among various radial basis functions available for data interpolation, we use thin plate spline (TPS) with compact support which has the form as

$$\varphi(x) = x^2 \log(x). \tag{4.16}$$

For more details, please consult the related papers [64, 65].

In practice, the control points are selected from the boundary points by a coarsening technique to reduce the computational cost. In addition, the outer boundary points far from the boundary are fixed to further improve the efficiency. To make the RBF interpolation vanishing at the outer boundary, a smoothing function is defined varying between $0$ and $1$ as

$$\psi(\tilde{x}) = \begin{cases} 1, & \tilde{x} \leqslant 0, \\ 1 - \tilde{x}^2(3 - 2\tilde{x}), & 0 \leqslant \tilde{x} \leqslant 1, \\ 0, & \tilde{x} \geqslant 1, \end{cases} \tag{4.17}$$

where $\tilde{x}$ is given by

$$\tilde{x} = \frac{||\mathbf{x}_i - \mathbf{x}_0|| - R_{\text{inner}}}{R_{\text{outer}} - R_{\text{inner}}}. \tag{4.18}$$

Here, $\mathbf{x}_i$ is the grid node coordinate, $\mathbf{x}_0$ is the focal point, $R_{\text{inner}}$ and $R_{\text{outer}}$ are two radii of the transitional region. Therefore, the corrected location for each inner grid point is given by

$$\Delta\tilde{\mathbf{x}}_{int} = \psi(\tilde{x})\Delta\mathbf{x}_{int}. \tag{4.19}$$

For fluid flows with moving boundary, we firstly approximate the internal mesh movement by constructing RBF interpolation function which is determined from fluid boundary movement. Navier-Stokes equations in ALE formulation are then solved based on the mesh movement and with the corresponding boundary conditions.

## 4.4 Numerical tests

### 4.4.1 2D isentropic vortex

Since the PV moment is solved different from Chapter 3, we still calculate the isentropic vortex [85] benchmark test to examine the convergence rate of inviscid part of the present scheme. The initial computational domain is $\Omega(0) = [0, 10] \times [0, 10]$ and

initial conditions are given in terms of primitive variables as

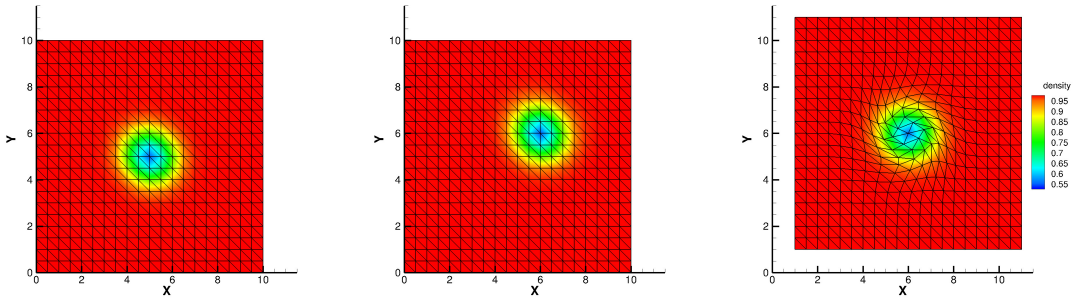
$$(\rho_0, u_0, v_0, p_0) = (1 + \delta\rho, 1 + \delta u, 1 + \delta v, 1 + \delta p), \tag{4.20}$$

where $\delta\rho, \delta u, \delta v, \delta p$ are the initial vortex perturbations. We set this perturbation at the center of computation domain $(5, 5)$ with the vortex strength $\epsilon = 5$. Then the initial perturbations are given as following,

$$\begin{aligned} \delta\rho &= (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \\ \delta u &= -\frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} (y - 5), \\ \delta v &= \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} (x - 5), \\ \delta p &= (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1, \end{aligned} \tag{4.21}$$

where $\delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}$ denotes the temperature perturbation, and $r^2 = (x - 5)^2 + (y - 5)^2$. The initial profile of density is presented in Figure 4.1 (a).

We compute this problem to time $t = 1$ in the Eulerian and quasi-Lagrangian framework respectively with gradually increased number of triangular elements. The periodic boundary condition is applied. The final vortices of density calculated on 800 triangular

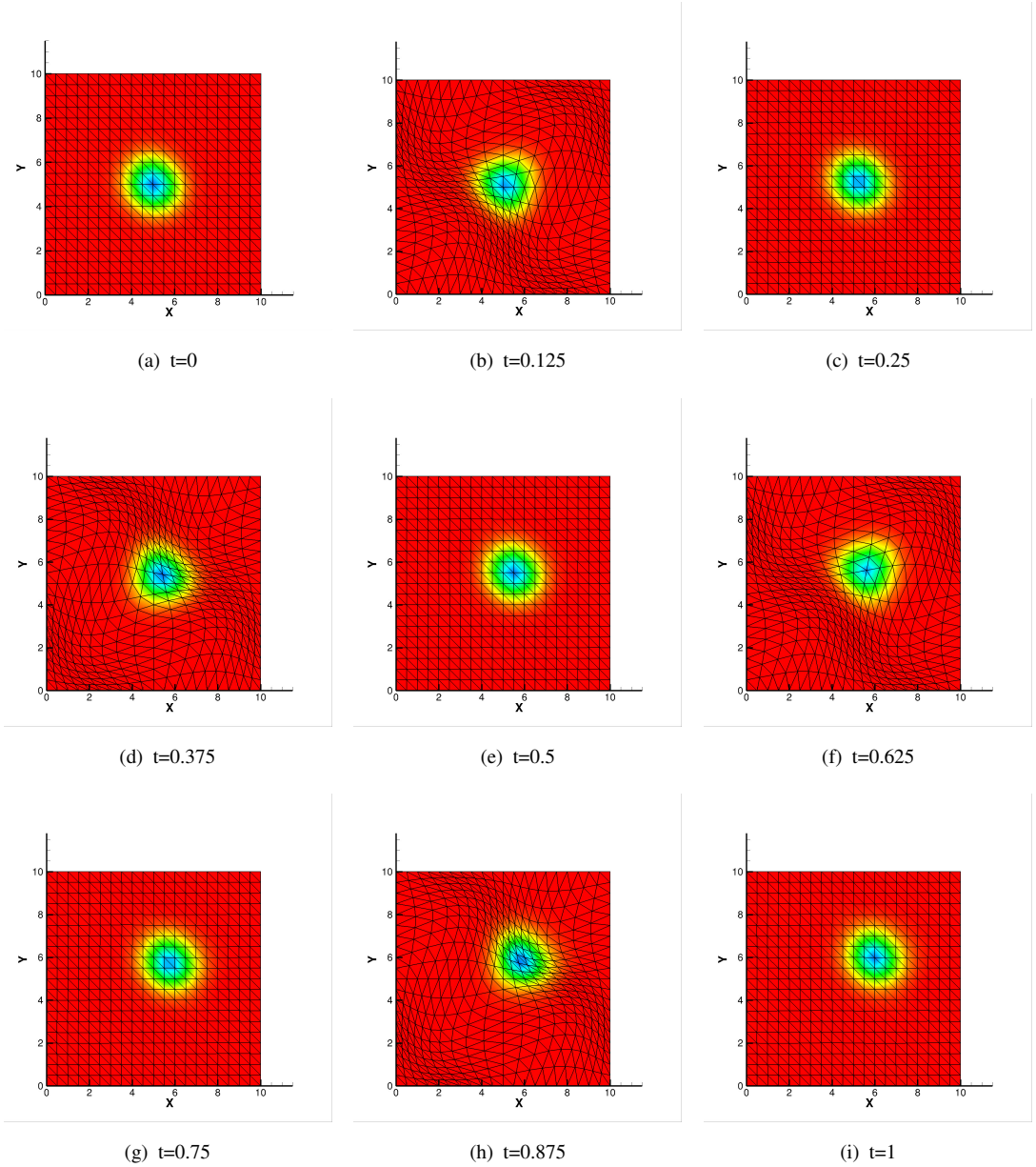



(a) Initial condition (b) Eulerian (c) Quasi-Lagrangian

FIGURE 4.1: (a) Initial profile of 2D isentropic vortex; (b) and (c) show the density results with the present scheme in Eulerian framework and quasi-Lagrangian framework at time $t = 1$.

elements show in Figure 4.1. To verify that the mesh can move arbitrarily for the proposed scheme, we also calculate the same case on a specific moving mesh framework where the mesh moves independent of fluid motion. The grid velocities vary with time as

$$u_g(\mathbf{x}, t) = X_0 \cdot 2\pi n_t / t_0 \cos(2\pi n_t t / t_0) \sin(2\pi n_x x / L_x) \sin(2\pi n_y y / L_y), \tag{4.22a}$$



(a) t=0 (b) t=0.125 (c) t=0.25

(d) t=0.375 (e) t=0.5 (f) t=0.625

(g) t=0.75 (h) t=0.875 (i) t=1

FIGURE 4.2: Density and grid variation for vortex advection with the grid velocity given in Equation (4.22).

$$v_g(\mathbf{x}, t) = Y_0 \cdot 2\pi n_t / t_0 \cos(2\pi n_t t / t_0) \sin(2\pi n_x x / L_x) \sin(2\pi n_y y / L_y), \tag{4.22b}$$

where $X_0 = 1, Y_0 = 1, n_t = 4, n_x = 1, n_y = 1, t_0 = 2, L_x = 10, L_y = 10$. Figure 4.2 shows the density and mesh variation with time. Though the mesh moves independent of fluid motion, the vortex propagates in the domain and accurately reproduced during whole computation process.

For this advection test of isentropic vortex, the analytical solution is available, which is just the time-shifted initial profile for the distance of 1 in both $x$ and $y$ directions. Thus

TABLE 4.1: Errors and convergence rates of density for 2D isentropic vortex

| cells | Eulerian | | | | Quasi-Lagrangian | | | | Specific-speed ALE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate | $L_1$ | rate | $L_\infty$ | rate |
| 200 | 4.16e-3 | - | 7.10e-2 | - | 3.12e-3 | - | 5.15e-2 | - | 5.78e-3 | - | 1.06e-1 | - |
| 800 | 8.01e-4 | 2.38 | 1.76e-2 | 2.01 | 6.06e-4 | 2.36 | 9.70e-3 | 2.41 | 1.46e-3 | 1.99 | 3.04e-2 | 1.81 |
| 3200 | 1.16e-4 | 2.78 | 3.07e-3 | 2.52 | 7.94e-5 | 2.93 | 1.35e-3 | 2.85 | 2.42e-4 | 2.59 | 5.12e-3 | 2.57 |
| 12800 | 1.47e-5 | 2.99 | 3.87e-4 | 2.99 | 9.50e-6 | 3.06 | 1.52e-4 | 3.15 | 3.18e-5 | 2.92 | 6.59e-4 | 2.96 |
| 51200 | 1.85e-6 | 2.99 | 4.81e-5 | 3.01 | 1.15e-6 | 3.04 | 1.75e-5 | 3.12 | 4.03e-6 | 2.98 | 8.24e-5 | 3.00 |

we measure the $L_1$ and $L_\infty$ errors of the Eulerian, quasi-Lagrangian and the specific-speed moving frameworks, and show the results in Table 4.1. As expected, third order numerical accuracy is achieved for all of these three frameworks. It also demonstrates that higher than second order accuracy can be achieved for linear edge element by ALE scheme.

### 4.4.2 Saltzman problem

Saltzman problem is a well-known challenging benchmark test to assess the robustness of numerical algorithms involving mesh movement, which consists of a strong piston-driven shock wave as shown in [33, 93–95]. It is firstly proposed by Dukowicz et al. in [96] for a two-dimensional uniformly skewed Cartesian grid in a computational domain of $[0, 1] \times [0, 0.1]$ which is discretized by $100 \times 10$ quadrilateral elements. The skewed coordinate of grid $\mathbf{x}^{'} = (x^{'}, y^{'})$ is transformed from the uniform Cartesian grid $\mathbf{x} = (x, y)$ by

$$\begin{cases} x^{'} = x + (0.1 - y)\sin(\pi x), \\ y^{'} = y. \end{cases} \tag{4.23}$$

In this research, each quadrilateral element is split into two triangular elements as shown in Figure 4.3. The initial condition is given by an ideal gas ($\gamma = \frac{5}{3}$) with

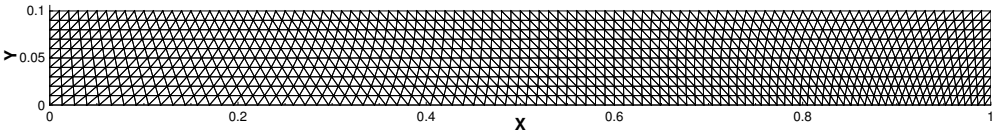


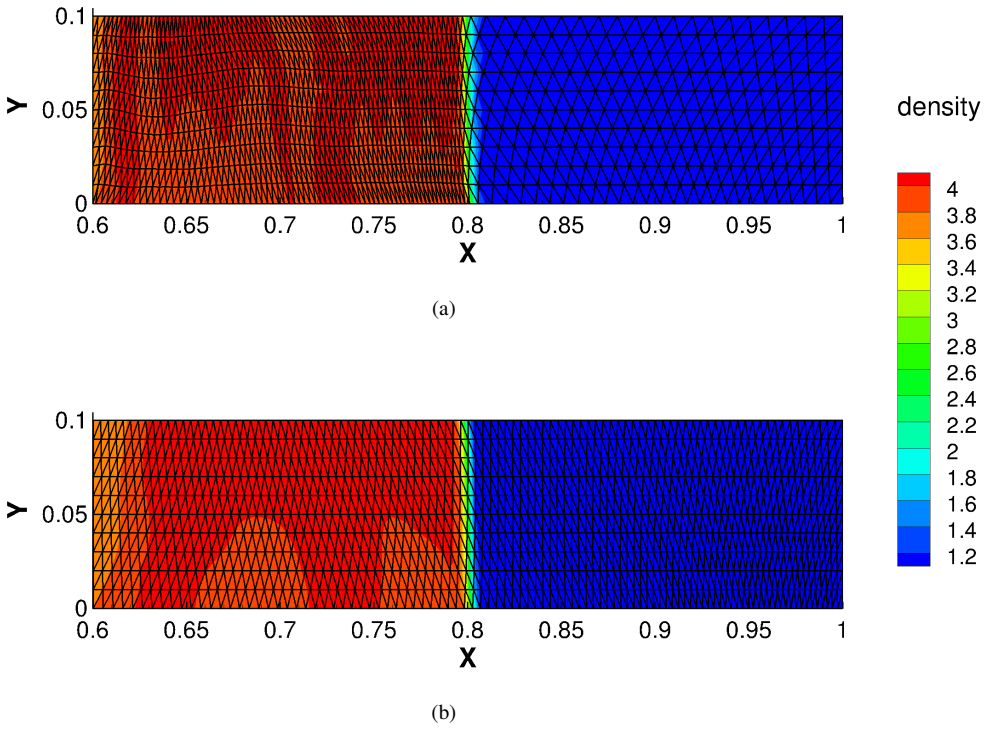

FIGURE 4.3: Initial mesh configuration of Saltzman problem.

$(\rho_0, u_0, v_0, p_0) = (1, 0, 0, \frac{2}{3} \cdot 10^{-4})$. The gas is pushed by a moving piston from left to right with velocity $(1, 0)$. A moving slip wall boundary condition is set for the piston on the left and reflective boundary condition is set for other boundaries. The analytical solution is a one-dimensional infinite strength shock wave that moves toward right direction at speed $\frac{4}{3}$ with a post-shock density plateau of 4. To meet the motion of the left

piston, we performed the simulation to time $t = 0.6$ in a quasi-Lagrangian framework and also in a specific-speed ALE framework where the mesh movement is given as

$$\mathbf{u}_g(\mathbf{x}, t) = (\frac{1-x}{1-t}, 0). \tag{4.24}$$

We use Rusanov scheme [97] for convection flux and multi-dimensional limiting process (MLP) scheme [56, 98] is employed for suppressing numerical oscillations. The mesh and density map at final time are given in Figure 4.4 for both frameworks and density values for the whole field are plotted in Figure 4.5. Since the ideal gas is filled in the closed container, we also measure the total mass of the gas. The mass variation with time is of machine error, which verifies the conservation of the present scheme.



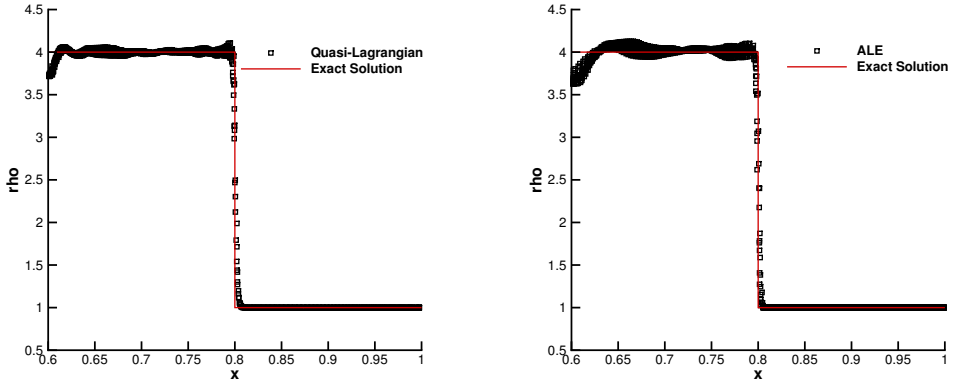

(a)

(b)

FIGURE 4.4: Mesh and density map for Saltzman problems at time t=0.6. (a) the quasi-Lagrangian framework (b) an ALE framework with a specific mesh moving strategy as given in Equation (4.24).

### 4.4.3 Viscous double Mach reflection

Double Mach reflection is originally proposed by Woodward and Colella in [99] for compressible Euler equations, which involves a Mach 10 shock that hits a $30^o$ ramp. We run a viscous version of double Mach reflection [100] in this section to verify the




(a) (b)

FIGURE 4.5: Density values in all cells for Saltzman problem at time t=0.6. (a) the quasi-Lagrangian framework (b) an ALE framework with a specific mesh moving strategy as given in Equation (4.24).

capability of the present solver for simulating viscous flows. The computational domain is set as $[0, 3.2] \times [0, 1]$. The initial conditions, which can be obtained by Rankine-Hugoniot conditions, are given here as

$$(\rho_0, u_0, v_0, p_0) = \begin{cases} (\frac{8.0}{\gamma}, 8.25\cos(\frac{\pi}{6}), -8.25\sin(\frac{\pi}{6}), \frac{116.5}{\gamma}), & \text{if} \quad y > \sqrt{3}(x - \frac{1}{6}), \\ (1.0, 0, 0, \frac{1.0}{\gamma}), & \text{if} \quad y < \sqrt{3}(x - \frac{1}{6}). \end{cases} \tag{4.25}$$

Reflecting wall lies along the bottom of the domain where $x > \frac{1}{6}$. Analytical solution of a shock Mach number $M_s = 10$ is imposed on the upper wall and bottom wall ($x < \frac{1}{6}$). Inflow and outflow boundary conditions are set for left and right sides respectively. We partitioned the domain into totally $296,418$ triangular elements with 640 and 200 uniform segments in the corresponding boundaries. Constant dynamic viscosity is used in this calculation. The other parameters are: $\gamma = 1.4$, $c_v = 2.5$, $\kappa = \frac{\gamma c_v \mu}{Pr}$ with $Pr = \frac{3}{4}$. We first performed the inviscid case up to time $t = 0.2$ with HLL Riemann solver [31] for inviscid finite volume flux, and also MLP scheme [56, 98] is used for limiting process. The well resolved vortex structures as shown in Figure 4.6 demonstrate the high fidelity of present method. For viscous shock waves, the Reynolds number is defined based on the shock speed and an unitary reference Length (L=1) as $Re_s = \frac{\rho_0 c_0 M_s L}{\mu}$ with $c_0 = \sqrt{\frac{\gamma p_0}{\rho_0}}$. We then calculate the cases with $Re_s = 1000$ and $Re_s = 100$ and the comparisons with inviscid case are shown in Figure 4.7. We observe that, with a rather larger physical viscosity, the small-scale vortex structures are suppressed, which is comparable with [100].

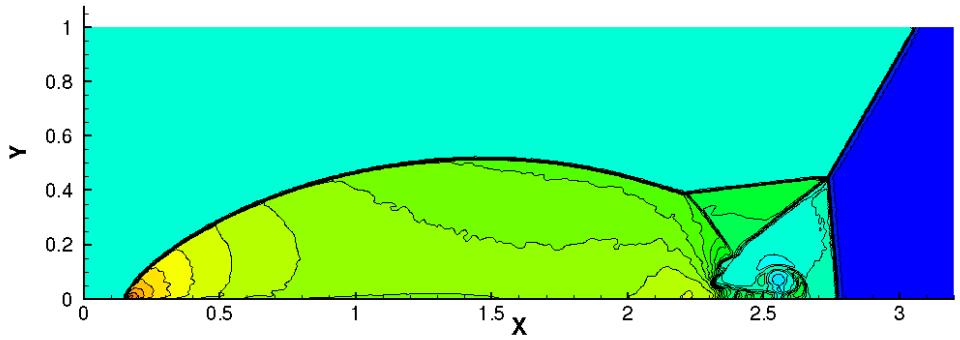
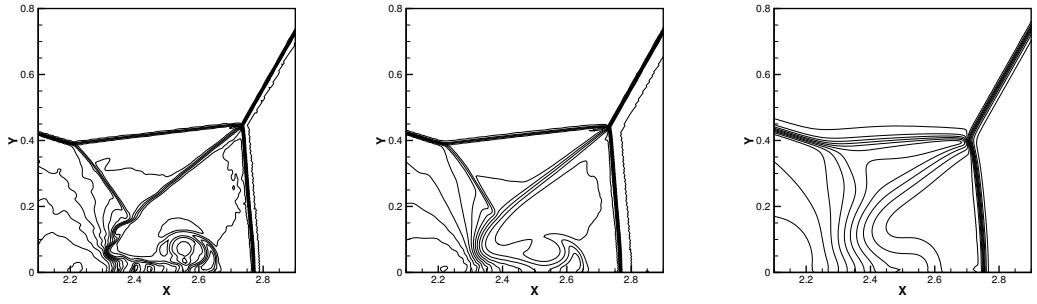



FIGURE 4.6: Numerical results of inviscid double Mach at $t = 0.2$ with $296,418$ triangular elements. 31 density isolines in the the interval [1,17].




(a) inviscid (b) $Re_s = 1000$ (c) $Re_s = 100$

FIGURE 4.7: Comparison of (a) inviscid case (b) $Re_s = 1000$ and (c) $Re_s = 100$ at $t = 0.2$ with $296,418$ triangular elements. 31 density isolines in the the interval [1,17].

### 4.4.4 Flow over cylinder

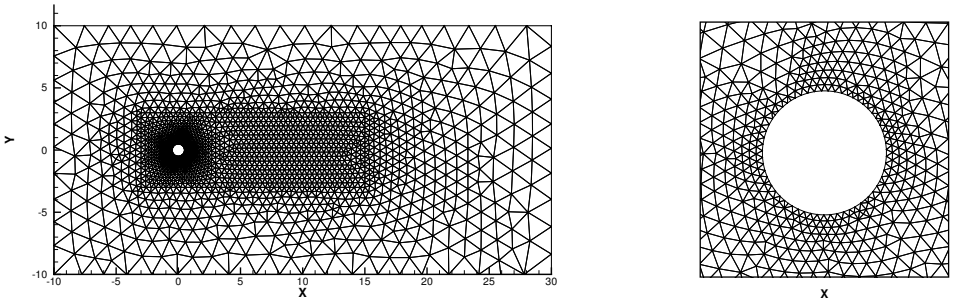
Flow around a fixed or oscillating cylinder is a widely used benchmark test to evaluate numerical models for fluid-structure interactions. We firstly compute the flow past a fixed cylinder to test the performance of the present solver for stationary grids, and then simulate the flow past a transversely oscillating cylinder to verify its capability for moving boundary problems.

#### 4.4.4.1 Fixed cylinder

A fixed cylinder of diameter $d = 1$ is located at the origin and the computational domain is set as a rectangular of $[-10 : 30] \times [-10 : 10]$. The Reynolds number based on the cylinder diameter is defined as $Re = \frac{u_\infty d}{\mu}$ and Mach number based on freestream flow is given as $Ma_\infty = \frac{u_\infty}{c_\infty}$, where $u_\infty$ and $c_\infty$ are the x-component of inlet velocity and inlet sound speed respectively. To obtain the desired Reynolds number $Re = 150$ and Mach

number $Ma_\infty = 0.2$, we set initial conditions $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 0.2, 0, 1/\gamma)$ and the Prantdtl number $Pr = 1$. The linear viscosity law is chosen in the calculation with $\beta = 2$ and $s = 0$. To avoid the reflection of shock wave from boundaries, the characteristic boundary condition [101] is used for inlet and outlet boundary. The slip boundary condition is set for top and bottom walls.

The grid dependency test was firstly conducted to choose a reliable mesh for later simulations. The computational domain is partitioned by a mesh of triangular elements, where the area near the cylinder has relatively smaller cell sizes in order to adequately represent the geometric configuration of the solid body as well as the induced wake and vortex structures. Three kinds of grids, Grid A, Grid B and Grid C, are generated where distances between neighbouring points on the cylinder are around $\frac{1}{20}d$, $\frac{1}{40}d$ and $\frac{1}{80}d$ respectively. Accordingly, the total element numbers for Grid A, Grid B and Grid C are 4358, 14060 and 48163 respectively. Figure 4.17(a) shows the image of Grid A, and the enlarged view of the grid cells near the cylinder is plotted in Figure 4.17(b).




(a) (b)

FIGURE 4.8: The mesh image for grid A (a), and the enlarged part near the cylinder surface (b).

The simulations were performed until final time $t = 1000$ on Grid A, Grid B and Grid C respectively. We show the numerical results for the case of Re=150 on Grid B in Figure 4.9 for Mach number and Figure 4.10 for vorticity. The Kármán vortex street was clearly reproduced after the symmetrical flow structure in the earlier stage was broken.

To quantitatively analyze the numerical accuracy, we measure the drag coefficient $C_d$ and the lift coefficient $C_l$ which are calculated as

$$C_d = \frac{F_D}{\frac{1}{2}\rho u_\infty^2 d} \quad \text{and} \quad \mathrm{C_l} = \frac{\mathrm{F_L}}{\frac{1}{2}\rho \mathrm{u}_\infty^2 \mathrm{d}}, \tag{4.26}$$

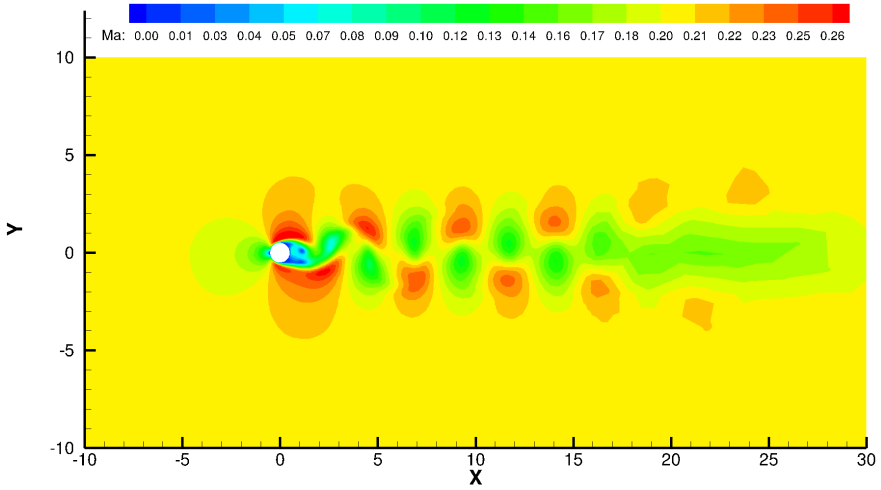



FIGURE 4.9: Mach number distribution for flow over a fixed cylinder at time t=1000 (Re=150).

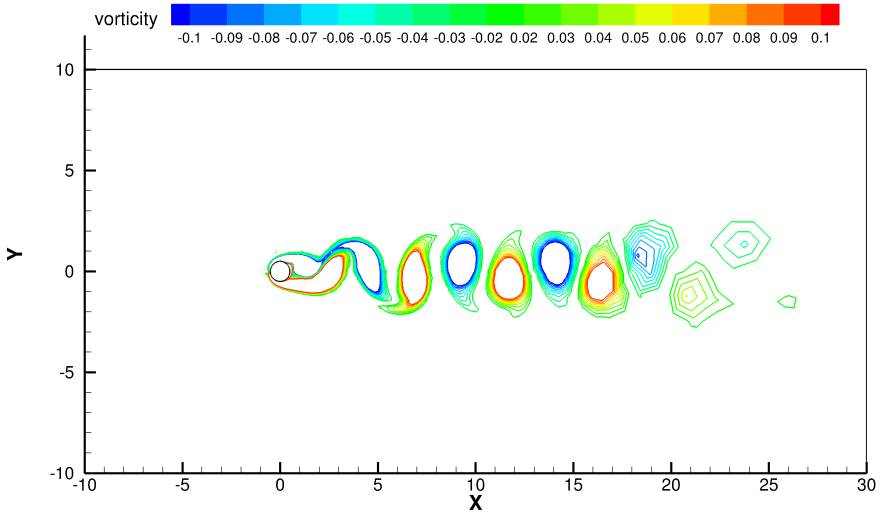



FIGURE 4.10: Contours of vorticity for flow over a fixed cylinder at time t=1000 (Re=150).

where $F_D$ and $F_L$ are the drag force and the lift force exerted on the cylinder. We plot the drag and lift coefficients for the case calculated on Grid C in Figure 4.11. In

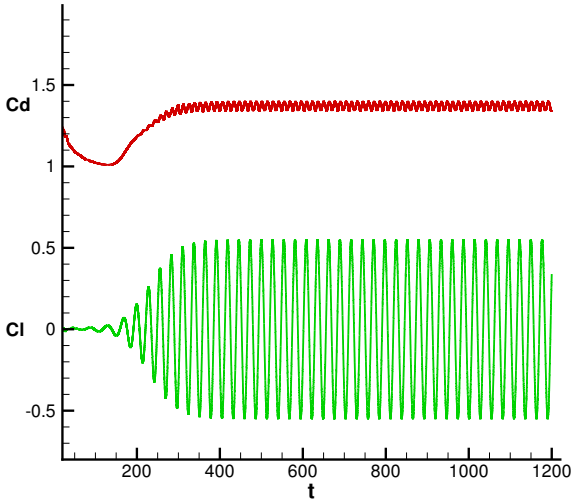



FIGURE 4.11: Temporal variations of drag and lift coefficients for flow over a fixed cylinder at Reynolds number Re=150.

the initial stage of computation, the wake around the cylinder keeps symmetry, thus the lift coefficient $C_l$ approaches zero. With the development of instability, the vortex sheds with gradually increasing shedding amplitude in the downstream of the cylinder and finally the periodic regime established. The profile of lift coefficient $C_l$ looks very close to the representative results in the literature. Due to the periodic vortex shedding, the drag coefficient $C_d$ correspondingly fluctuates as shown in Figure 4.11. We then calculated the mean drag coefficient

$$\overline{C}_d = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} C_d(t) dt \tag{4.27}$$

with $[t_1 : t_2]$ covering several periods of time span sampled during the periodic regime to reduce the statistical error. The Strouhal frequency $f_s$ is also an important parameter for the Kármán streets, which was calculated from the period of lift coefficient in this research. And then the Strouhal number was calculated by

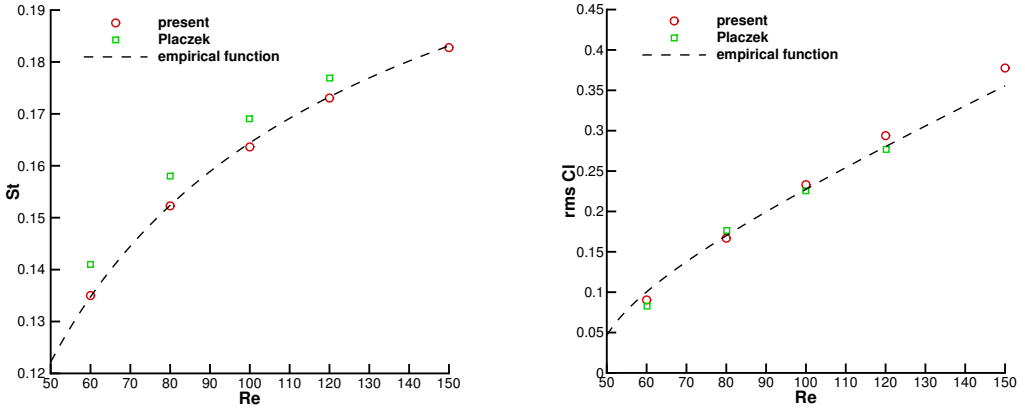
$$\text{St} = f_s \frac{d}{u_\infty}. \tag{4.28}$$

TABLE 4.2: Time-averaged drag coefficient and the Strouhal number on gradually refined meshes.

| | Time-averaged $\overline{C}_d$ | Strouhal number St |
|---|---|---|
| Grid A | 1.364 | 0.172 |
| Grid B | 1.371 | 0.183 |
| Grid C | 1.371 | 0.184 |
| DG/FV (4th order) [102] | 1.348 | 0.184 |
| Müller [103] | 1.34 | 0.183 |

We measured the time-averaged drag coefficient $\overline{C}_d$ and Strouhal number St for gradually refined meshes, Grid A, Grid B and Grid C as shown in Table 4.2. The time-averaged drag coefficients approaches 1.371 and the Strouhal number also converges with the mesh finer than Grid B. Therefore, we assume that Grid B can give adequate accuracy for later test cases. Compared with the existing results, the Strouhal number has a good agreement with results calculated by DG/FV method [102] and Müller's simulation [103].

We also calculated the cases with $Re = 60, 80, 100$ and 120 to evaluate the capability of the numerical model to simulate flows with different Reynolds number. Figure 4.12 compares the numerical results of Strouhal numbers and r.m.s. (root-mean-square) lift coefficients with the empirical function [104] and Placzek's numerical results [105]. The good agreement of Strouhal number and r.m.s. lift coefficients demonstrates the accuracy of the present solver for stationary mesh, which provides a good base for implementation to moving mesh applications.




(a) Strouhal number

(b) r.m.s. Lift coefficient

FIGURE 4.12: Strouhal number and r.m.s. lift coefficient for different Reynolds numbers.

#### 4.4.4.2 Oscillating cylinder

Viscous flow over oscillating cylinder provides a test bed for numerical solvers for fluid-structure interactions. One of the most interesting phenomena for flow past forced oscillating cylinder is the lock-in phenomenon, which has been reported in the literature to verify numerical models. The characteristics of vortices and wakes around the cylinder is significantly different from that of a fixed cylinder. We investigate the flow over a forced oscillating cylinder in this part with a Reynolds number Re=100.

The cylinder is forced to oscillate in the transversal direction of the main uniform flow with a sinusoidal motion in time defined by

$$y(t) = y_{\max} \sin(2\pi f_0 t) \tag{4.29}$$

where $y_{\max}$ is the maximal displacement which can be characterized by the non-dimensional amplitude $A = \frac{y_{\max}}{d}$. The oscillating frequency, $f_0$, which associates with the Strouhal frequency $f_s$ for fixed cylinder, plays an essential role in lock-in phenomenon, which can be categorized by the frequency ratio $f = \frac{f_0}{f_s}$. Figure 4.13 represents the lock-in zone with respect to the amplitude $A$ and the frequency ratio $f$ where the data is collected from Koopmann and Anagnostopoulos's studies [1, 2].

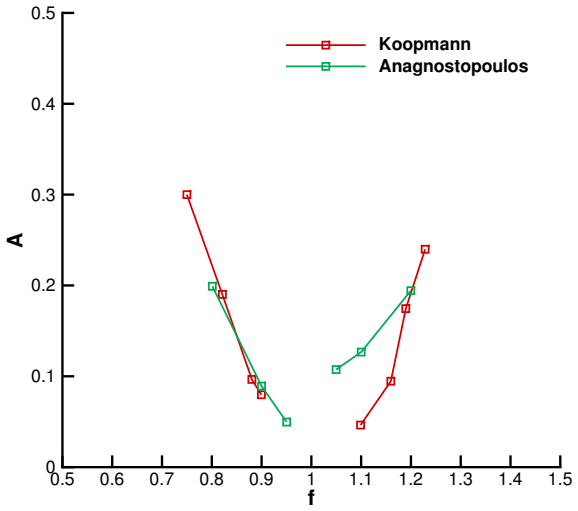
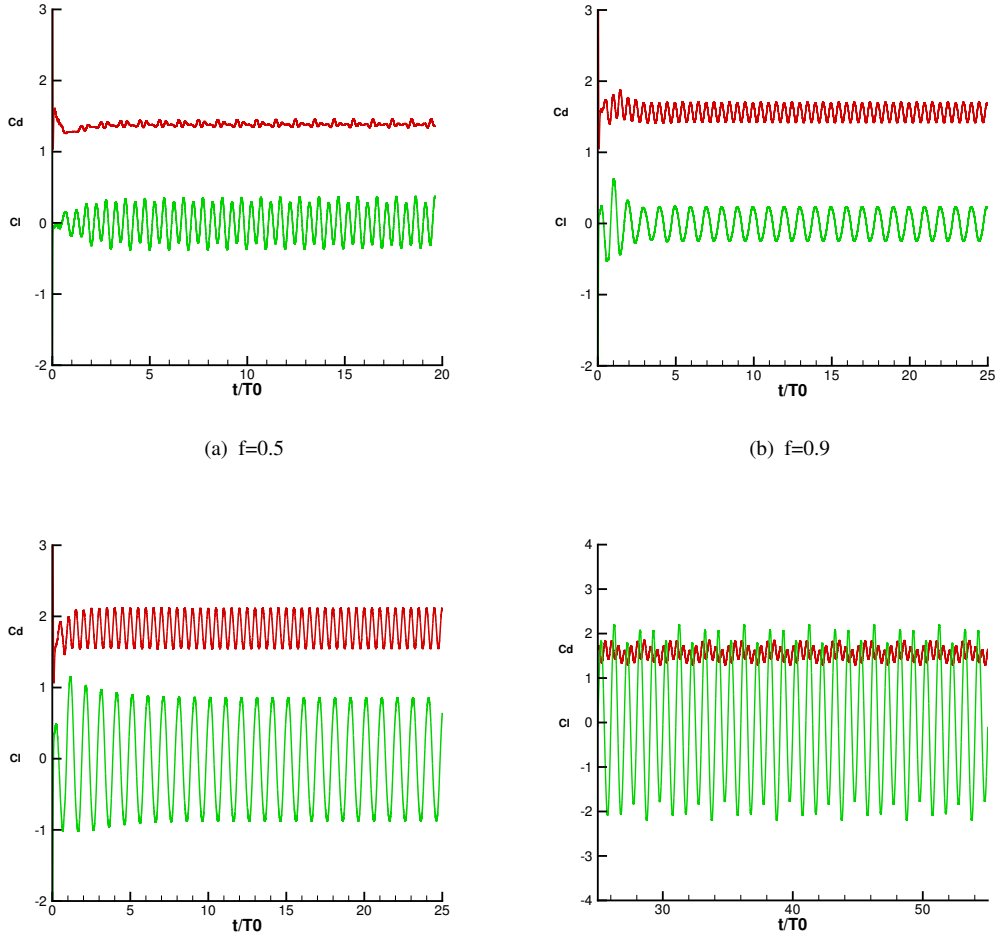



FIGURE 4.13: Lock-in regime for flow past an oscillating cylinder in ($f$,$A$) plane at $Re = 100$ with the data collected from Koopmann and Anagnostopoulos's studies [1, 2]. The region above the data line is the lock-in region and otherwise unlocked region.

According to Figure 4.13, we keep the amplitude constant as $A = 0.3$ to devise the locked and unlocked configurations with $f = 0.5, 0.9, 1.1$ and $1.5$. We plot the drag coefficient and lift coefficient over time for four cases in Figure 4.14 with time characterized by the oscillating period $T_0$. As expected in Figure 4.13, the case with $f = 0.9$ and $f = 1.1$ are locked where the vortex shedding frequency are synchronized with the cylinder oscillating frequency as shown in Figure 4.14 (c) and (d). For locked con-




(a) f=0.5

(b) f=0.9

(c) f=1.1

(d) f=1.5

FIGURE 4.14: Time variation of drag and lift coefficients for certain frequencies (a)$f = 0.5$, (b)$f = 0.9$,(c)$f = 1.1$,(d)$f = 1.5$ with $A = 0.3$ at $Re = 100$.

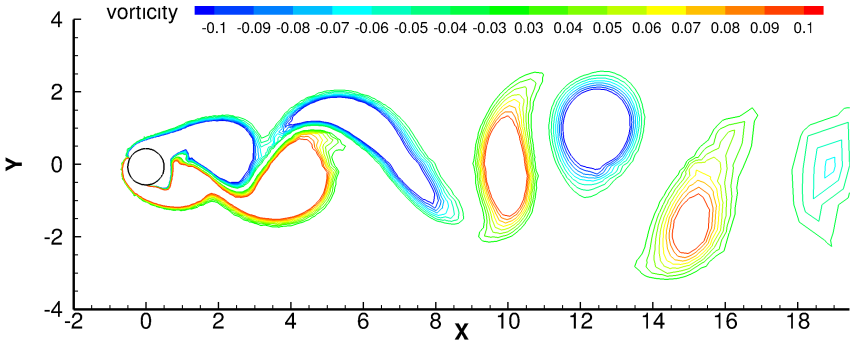
figurations, the mean drag coefficient has a significant increase which reaches $1.56$ for $f = 0.9$ and $1.83$ for $f = 1.1$, while it is $1.38$ for the fixed cylinder. The maximum lift coefficient reduces to $0.24$ for $f = 0.9$ but increases to $0.86$ for $f = 1.1$, in comparison with its value of $0.33$ for the fixed cylinder. Unlocked wakes are formed with the frequency ratio $f = 0.5$ and $f = 1.5$ which represent more complicated phenomena than locked cases. The variation of lift coefficients does not follow the sinusoidal profile any

more. The period of signal is over several cylinder oscillation cycles which is called as the beating period $T_b$. The simpler case is $f = 0.5$ that the beating periodic $T_b$ is equal to the forced oscillation period $T_0$. In this case, it is twice of the Strouhal period $T_s$. More complex case is $f = 1.5$ where the beating period consists several oscillating periods as shown in Figure 4.14 (d). The snapshots of vorticity contour lines within one oscillating period are plotted in Figure 4.15. The cylinder moves with a sinusoidal function in y direction: (a) $t = t_0$, it is located at the origin of the domain; (b) $t = t_0 + \frac{1}{4}T_0$, it moves to the maximum displacement in y direction; (c) $t = t_0 + \frac{1}{2}T_0$, it returns to the origin; (d) $t = t_0 + \frac{3}{4}T_0$, it moves to the maximum displacement in the reverse direction of y axis.

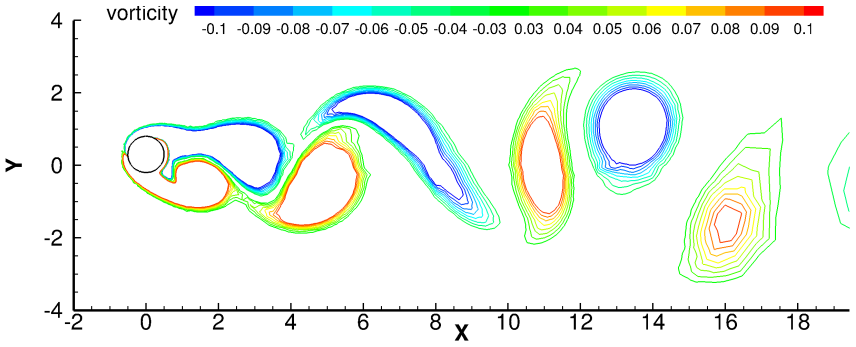
Cases with a larger amplitude $A = 2$ are also calculated to show the ability of our proposed method for large mesh distortions. The oscillating period ratio is set as $f = 0.5$. We show the snapshots of vorticity contour lines within one cylinder oscillating period in Figure 4.16. Figure 4.17 (a) gives the mesh image when the cylinder moves to the maximum displacement in y direction, and we magnify the area near the cylinder surface in Figure 4.17 (b). By using RBF interpolation function, though large boundary movements are specified, the displacements of the boundary points are smoothly interpolated to the internal grid points based on the distance of the internal nodes to the boundary which is the essence of RBF function to ensure whole mesh maintains a good quality. In the simulation of forced oscillation problems, RBF interpolation is outside the Runge Kutta loop, which makes the grid velocity $u_g$ constant during the whole time step (including all sub-steps in the 3-stage Runge-Kutta method). It is in fact equivalent to the case that each grid point moves at a constant speed in the duration of each time step. In this way, the computational cost of RBF interpolation can be substantially reduced without losing numerical accuracy. Meanwhile, a coarsening technique is employed as presented in [65]. We select every other 5 points in the cylinder boundary as control points for saving computational cost. We estimate the computational cost by running the case up to time t=10 on a PC with an Intel Core i7-4790 CPU @ 3.60GHz, the computational time for RBF interpolation is 48.63s while the whole running time is 1433.35s.

## 4.5 Short summary

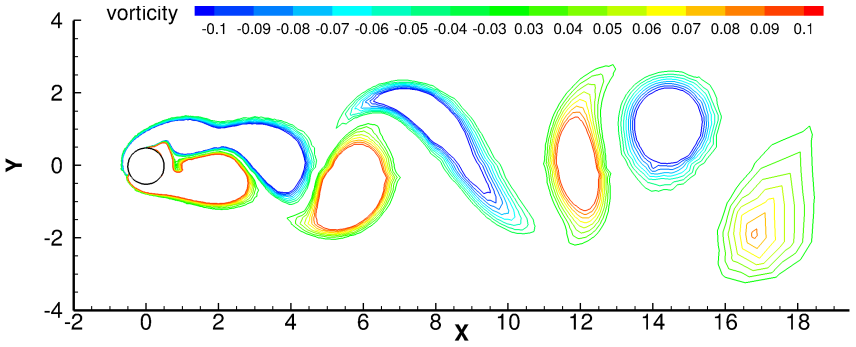
In this chapter, we extend the multi-moment finite volume scheme to viscous compressible flows. The VIA is computed by a finite volume method, where conventional



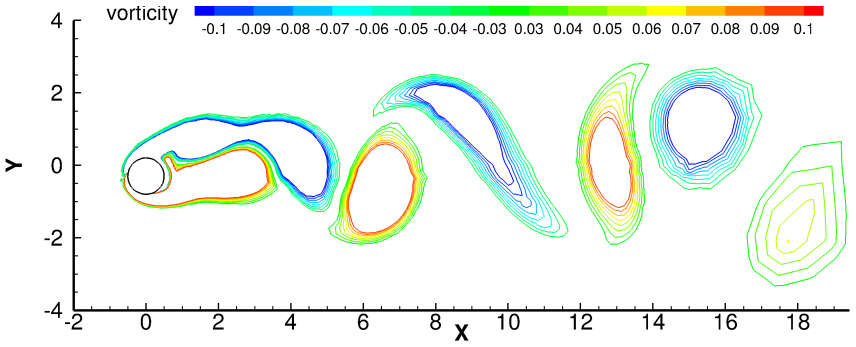

(a) t=$t_0$




(b) t=$t_0 + \frac{1}{4}T_0$

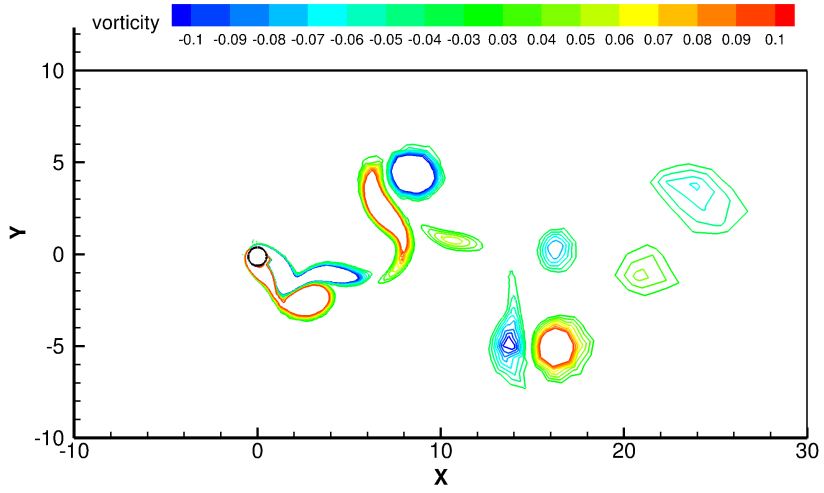



(c) t=$t_0 + \frac{1}{2}T_0$

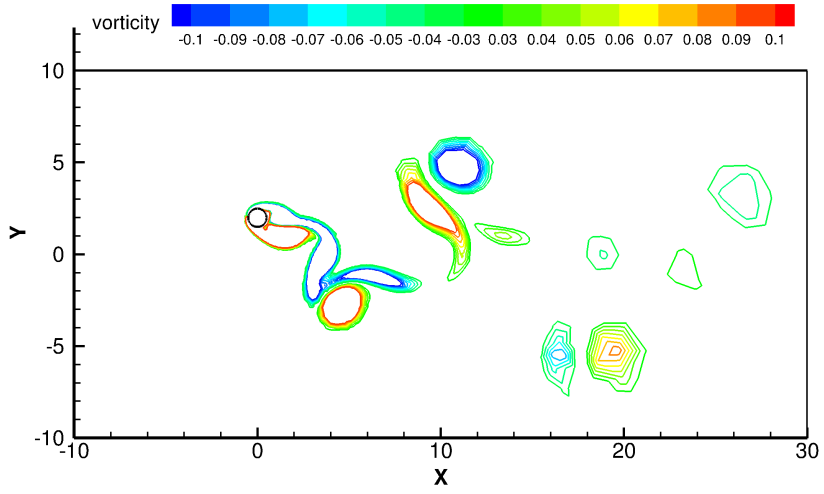



(d) t=$t_0 + \frac{3}{4}T_0$

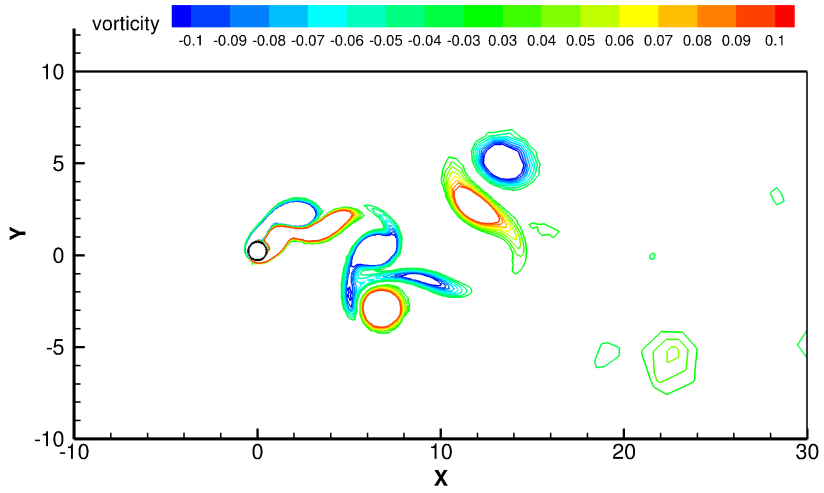
FIGURE 4.15: The snapshots of vorticity contour lines within one oscillating period (A=0.3, F=1.5).



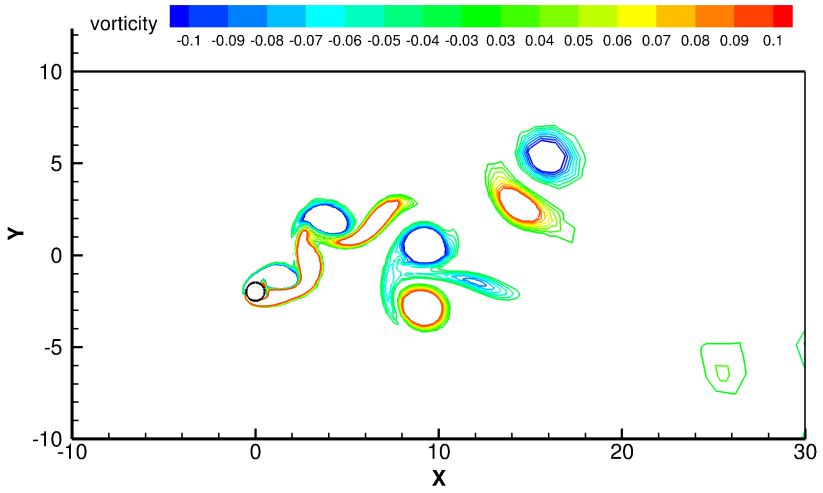

(a) t=$t_0$




(b) t=$t_0 + \frac{1}{4}T_0$

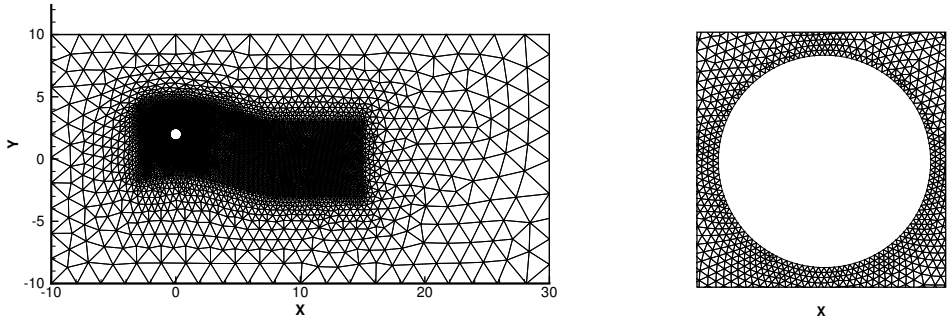



(c) t=$t_0 + \frac{1}{2}T_0$




(g) t=$t_0 + \frac{3}{4}T_0$

FIGURE 4.16: Same as Figure 4.15, but for (A=2, F=0.5).




(a) (b)

FIGURE 4.17: The mesh image for Grid B when the cylinder moves to the maximum displacement in y direction (A=2, F=0.5) (a), and the enlarged part near the cylinder surface (b).

Riemann solvers can be used for calculating the convective fluxes and variable gradients on cell edge required in approximating the viscous fluxes are simply calculated from the interpolation of neighboring cell centers. A minor difference from Chapter 3 is that the PV moment is calculated by differential equation with respect to conservative variables, which is more direct for adding viscous term. For updating PV moment, numerical formulations of inviscid fluxes are still by using Roe's Riemann solver, and viscous term is simply calculated by the $\mathcal{TEC}$ (Time-evolution Converting) formula. Numerical tests for compressible flows with high Mach numbers, such as Saltzman problem, viscous double Mach reflection are firstly calculated to demonstrate the performance of present solver. It is pointed out that defining and using the physical variables at the vertices of

mesh cells as new computational variables leads to an accurate and efficient numerical formulation for ALE implementation.

To simulate fluid flows with moving boundaries, we integrate the fluid model with the radial basis function (RBF) interpolation which transfers the boundary movement to internal mesh points. Finally, flow past an oscillating cylinder is computed to show the feasibility for fluid-structure interaction applications. With different oscillating amplitude and frequency, the vortex structure around the cylinder is different. In our simulation, lock-in phenomena are observed by setting cylinder oscillating frequency with $f = 0.9$ and $f = 1.1$, and oscillating amplitude with $A = 0.3$, which agrees well with published studies. To evaluate the efficiency of RBF interpolation, we estimate the computational time in single-core calculation, where the RBF code takes only $3.4\%$ of the total computational time by using a coarsening technique. Numerical tests verify the capability of present solver for calculating viscous compressible flows involved with forced moving body.

# Chapter 5

# Incompressible flows on moving domain and fluid-solid body interaction

## 5.1 Brief introduction

Multiphysics phenomena, which involve the interaction of at least two different physical fields, have attracted much attention in recent years [106, 107]. Fluid structure interaction (FSI), as one classical problem of this sort, is of great interest in broad-range engineering applications, such as aerospace engineering, hemodynamics and oceanic engineering, just a few among many others. The application of computational fluid dynamics (CFD) in FSI is rather challenging due to the fact that the spatial domain occupied by fluids always varies with time and its boundary is a subsequent result of fluid-structure interaction. There have been some significant contributions in FSI field [108–110], where flexible bodies are fully coupled with fluid flows. Meanwhile, for some applications, the deformation of structure shape is so slight or negligible for the phenomenon of interest that the structure can be idealized as a rigid body. In such a case, the FSI can be simplified into fluid-rigid body interaction, which does not require numerical formulation of solid deformation and largely reduces the computational complexity. In this paper, we focus on a novel numerical formation for moving mesh and FSI, and limit our interest to fluid-rigid body interaction.

The illustration of mathematical model for FSI is given in Figure 5.1, where the fluid model and solid model are calculated simultaneously. In this system, the solid model is updated with the force $\boldsymbol{f}$ or torque $\boldsymbol{T}$ evaluated from the physical variables of fluid,

and the fluid model is calculated with the displacement and velocity on the moving interface which is provided by solid body. Thus, the major parts of this system are: fluid model, solid model and fluid-solid coupling algorithm, which will present in the following sections.

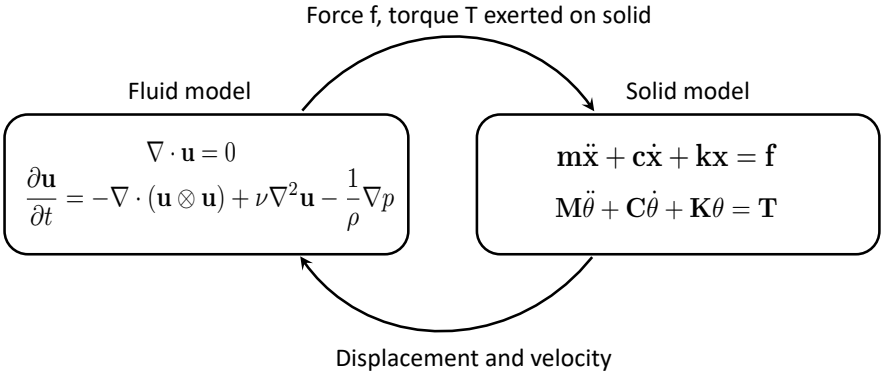



FIGURE 5.1: FSI numerical model.

## 5.2 Numerical model for incompressible flows on moving domain

### 5.2.1 Incompressible Navier-Stokes equations

For fluid flows of low Mach number ($Ma \ll 1$), the compressibility of fluid can be neglected and the fluid density is generally treated as constant in fluid mechanics. Meanwhile, the energy equation is no longer required for closure of the system. Thus, the incompressible flow can be described by simplifying Eq.(2.15) as

$$\nabla \cdot \mathbf{u} = 0, \tag{5.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p, \tag{5.2}$$

with $\mathbf{u}$, $\rho$, $p$ and $\nu$ being velocity, density, pressure and viscosity of fluid respectively. To describe fluid flows with arbitrary boundary deformation, the momentum equation is cast into Arbitrary Lagrangian Eulerian (ALE) formulation in integral and differential form [68] respectively as

Integral form:

$$\frac{\partial}{\partial t} \int_{\Omega_{\boldsymbol{\chi}}(t)} \mathbf{u} d\Omega = -\int_{\Gamma(t)} (\mathbf{u} - \mathbf{u}_g) \otimes \mathbf{u} \cdot \mathbf{n} d\Gamma + \int_{\Gamma(t)} \nu \nabla \mathbf{u} \cdot \mathbf{n} d\Gamma - \int_{\Gamma(t)} \frac{1}{\rho} p \mathbf{n} d\Gamma, \tag{5.3}$$

Differential form:

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\chi} = -(\mathbf{u}-\mathbf{u}_g)\otimes\nabla\mathbf{u} + \nu\nabla^2\mathbf{u} - \frac{1}{\rho}\nabla p, \tag{5.4}$$

where $\Omega_{\boldsymbol{\chi}}(t)$ is a moving control volume enclosed by its boundary $\Gamma(t)$, $\left.\frac{\partial}{\partial t}\right|_{\chi}$ represents the operator of time derivative with respect to the referential moving frame, $\mathbf{u}_g = \dot{\mathbf{x}}$ is the grid velocity of fluid domain, and $\mathbf{n}$ stands for the surface normal vector of the boundary $\Gamma(t)$. For a uniform flow, (5.3) reduces to a geometrical conservation law (GCL) [25]

$$\frac{dV}{dt} - \int_{\Gamma(t)} \mathbf{u}_g\mathbf{n}d\Gamma = 0, \tag{5.5}$$

where $V = \int_{\Omega_{\boldsymbol{\chi}}(t)} d\Omega$ represents the moving volume.

For fluid-rigid body interaction, fluid-solid interface is treated as the boundary of both fluid and solid. Thus, the force $\mathbf{f}$ or torque $\mathbf{T}$ exerted on the rigid body can be evaluated by integrating fluid pressure and viscous stress over the body surface as

$$\mathbf{f} = \rho\int_{\partial\Omega_I} (p\mathbf{n} - \nu\nabla\mathbf{u}\cdot\mathbf{n})\, d\Gamma, \tag{5.6}$$

$$\mathbf{T} = \rho\int_{\partial\Omega_I} (\mathbf{x}-\mathbf{x}_c)\times(p\mathbf{n} - \nu\nabla\mathbf{u}\cdot\mathbf{n})\, d\Gamma, \tag{5.7}$$

where $\partial\Omega_I$ represents the fluid-solid interface, $\mathbf{x}_c$ the mass center of rigid body.

### 5.2.2 Solution procedure of pressure projection

The pressure projection method, a fractional-step method [15, 61, 62], which has been popularly used in the simulation of incompressible flows, is adopted in the present simulation. We summarize the numerical procedure for updating fluid variables from time $t^n$ to $t^{n+1}$ as follows:

1. Given the velocity $\mathbf{u}^n$ at step $n$, compute the convection and diffusion terms of momentum equation (5.3) and (5.4) with GCL condition (5.5) to put forward the intermediate velocity $\mathbf{u}^*$ by ALE integral and differential formulations as,

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \mathbf{u}d\Omega = -\int_{\Gamma(t)} (\mathbf{u}-\mathbf{u}_g)\otimes\mathbf{u}^n\cdot\mathbf{n}d\Gamma + \int_{\Gamma(t)} \nu\nabla\mathbf{u}^n\cdot\mathbf{n}d\Gamma, \tag{5.8}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}-\mathbf{u}_g)\cdot\nabla\mathbf{u}^n + \nu\nabla^2\mathbf{u}^n. \tag{5.9}$$

2. Generally, the intermediate velocity $\mathbf{u}^*$ does not satisfy the continuity equation. Thus, we calculate the pressure by solving Poisson equation to enforce the divergence-free constraint condition

$$\nabla \cdot \left( \frac{1}{\rho} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \tag{5.10}$$

3. Correct the velocity by the correction step equivalently by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla p^{n+1}. \tag{5.11}$$

It is obvious that by this procedure the updated velocity in the next time step satisfies the continuity property.

### 5.2.3 Semi-discrete formulations

As shown in Figure 5.2. We define both VIA ($\overline{\mathbf{u}}_i$) and PV ($\mathbf{u}_{ik}$) for velocity while only VIA ($\overline{p}_i$) for pressure for simplicity. Unlike the conventional FVM, both VIA

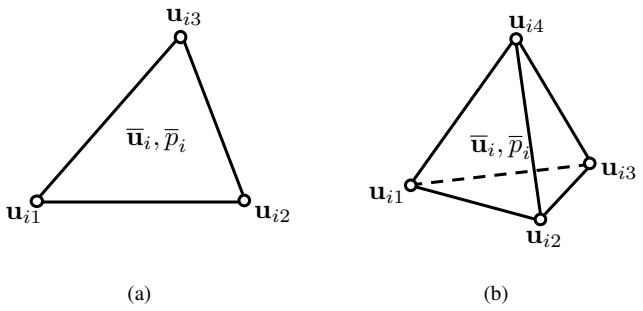



FIGURE 5.2: The arrangement of velocity and pressure defined in multi-moment finite volume method on triangular (a) and tetrahedral (b) elements.

and PV of velocity field are treated as prognostic variables and updated separately at each time step through the pressure-projection solution procedure. We compute the evolution equations of VIA by finite volume scheme and those of PVs efficiently in finite difference formulation. As elaborated in our previous work [54, 55, 111], VPM method effectively improves the solution accuracy and numerical robustness without loss of algorithmic simplicity. Next, we discuss the spatial discrete formulations given in section 5.2.2 at each sub-step in details.

In the first step, ALE convection and diffusion terms of momentum equation in integral and differential forms are solved for the evolution of VIA and PV moments respectively. Employing the concept of finite volume method, we can obtain the semi-discrete formulations of the evolution equation (5.8) on the moving grids as

$$\frac{\partial}{\partial t}(\overline{\mathbf{u}}_i V_i) \approx -\sum_{j=1}^{\mathbb{J}} \left(\overline{\mathbf{u}}_{\Gamma_{ij}} - \overline{\mathbf{u}}_{g,ij}\right) \overline{\mathbf{u}}_{ij} \cdot \mathbf{n}_{ij} |\Gamma_{ij}| + \sum_{j=1}^{\mathbb{J}} \nu \overline{\nabla \mathbf{u}}_{ij} \cdot \mathbf{n}_{ij} |\Gamma_{ij}|, \tag{5.12}$$

where $\overline{\mathbf{u}}_{ij}$ denotes the surface-averaged value calculated from multi-moment reconstruction (5.2), $\overline{\mathbf{u}}_{g,ij}$ the moving velocity approximated at the boundary surface and $\overline{\nabla \mathbf{u}}_{ij}$ the surface gradient computed by the least-square method. Given grid velocities on mesh nodes, we suppose that the grid velocity has a linear distribution over cell surface. Thus, the surface-averaged grid velocity $\overline{\mathbf{u}}_{g,ij}$ is approximated from the algebraic average of surface node velocities.

To remain the global conservation in ALE formulation, we solve (5.12) along with the geometrical conservation law (5.5) numerically by

$$\frac{dV_i}{dt} = \sum_{j=1}^{\mathbb{J}} \overline{\mathbf{u}}_{g,ij} \cdot \mathbf{n}_{ij} |\Gamma_{ij}|, \tag{5.13}$$

which preserves a uniform flow on a grid that moves arbitrarily in space regardless of numerical discretization scheme used.

The semi-discrete formulation of (5.9) is derived for PVs based on the relation between the time and referential derivatives

$$\frac{\partial \mathbf{u}_{\mathfrak{p}}}{\partial t} = -(\mathbf{u}_{\mathfrak{p}} - \mathbf{u}_{g,\mathfrak{p}}) \cdot (\nabla \mathbf{u})_{\mathfrak{p}} + \nu \cdot \overset{\text{VIA}\to\text{PV}}{\mathcal{TEC}} (\nabla \mathbf{u}_{ij}), \tag{5.14}$$

where the gradient of convection term is evaluated from a weighted upwind scheme [54] and the diffusion term is simply calculated by the $\mathcal{TEC}$ (Time-evolution Converting) scheme [87, 111]. After the computation of convection and diffusion equation, the VIA and PV of velocity are put forward to the intermediate values as $\overline{\mathbf{u}}_i^*$ and $\mathbf{u}_{\mathfrak{p}}^*$ respectively.

To enforce the divergence-free condition, the pressure Poisson equation is recast in an integral form as

$$\sum_{j=1}^{\mathbb{J}} \left(\nabla \overline{p}_{ij}^{n+1} \cdot \mathbf{n}_{ij} |\Gamma_{ij}|\right) = \frac{1}{\Delta t} \sum_{j=1}^{\mathbb{J}} \left(\overline{\mathbf{u}}_{ij}^* \cdot \mathbf{n}_{ij} |\Gamma_{ij}|\right), \tag{5.15}$$

where $\overline{\mathbf{u}}^*_{ij}$ is obtained from the multi-moment reconstruction in terms of the intermediate velocity values, i.e. $\overline{\mathbf{u}}^*$ and $\mathbf{u}^*_{\mathfrak{p}}$. The calculations of the pressure gradient $\nabla\overline{p}_{ij}$ in (5.15) are split to the orthogonal part and the non-orthogonal part, where the orthogonal part is calculated implicitly with central difference and the non-orthogonal part is computed explicitly for deferred correction [54, 112].

Finally, the pressure obtained above is then used to correct the velocity at boundary surface by (5.11), while VIA and PV of velocity are corrected through $\mathcal{TEC}$ formulation

$$\begin{aligned}
\overline{\mathbf{u}}^{n+1}_{ij} &= \overline{\mathbf{u}}^*_{ij} - \frac{\triangle t}{\rho}\nabla\overline{p}^{n+1}_{ij}, \\
\overline{\mathbf{u}}^{n+1}_{i} &= \overline{\mathbf{u}}^*_{i} - \overset{\text{SIA}\rightarrow\text{VIA}}{\mathcal{TEC}}\left((\overline{\mathbf{u}}^{n+1}_{ij} - \overline{\mathbf{u}}^*_{ij})\cdot\mathbf{n}_{ij}\right), \\
\mathbf{u}^{n+1}_{\mathfrak{p}} &= \mathbf{u}^*_{\mathfrak{p}} - \overset{\text{VIA}\rightarrow\text{PV}}{\mathcal{TEC}}\left((\overline{\mathbf{u}}^{n+1}_{ij} - \overline{\mathbf{u}}^*_{ij})\cdot\mathbf{n}_{ij}\right).
\end{aligned}$$

In these formulation, we first correct the velocity at cell surface $\overline{\mathbf{u}}_{ij}$, so called surface integrate average (SIA). SIA is then used to correct VIA and finally PV by $\mathcal{TEC}$ formulation. The interested reader can refer to [80, 111] for more details.

For FSI problems, the boundary conditions of moving interface need to be treated carefully, since they substantially affect the interaction phenomenon. In this study, PVs on the interface of multi-materials, which are the interactive results of fluid and solid motions (see section 5.4 for details), are directly used as part of the boundary conditions for fluid motion. As PVs are updated at every time step and always available at the fluid-solid interface in the present numerical formulation, the algorithm for moving mesh and FSI can be very simple and accurate.

For fluid flows with moving boundary, we firstly approximate the internal mesh movement by constructing RBF interpolation function which is determined from fluid boundary movement. Navier-Stokes equations in ALE formulation are then solved based on the mesh movement and with the corresponding boundary conditions. Finally, the computed fluid state can be used for evaluating the force or torque for fluid-rigid body interaction phenomena by (5.6) and (5.7). From the perspective of the rigid body, the fluid model can be treated as a function $\mathcal{F}$, where the computational variable is the fluid-solid interface $\mathbf{x}_I$ and the output is the force or torque on the body. Thus, we denote it as

$$\sigma = \mathcal{F}(\mathbf{x}_I) \tag{5.16}$$

where $\sigma$ represents the force $\mathbf{f}$ or torque $\mathbf{T}$.

## 5.3 Motion of an elastically mounted rigid body

For the dynamics of a rigid body in a fluid-structure coupled system, we first consider the motion of solid translation which can be described by a second-order ODE (ordinary differential equation)

$$\mathbf{m}\ddot{\mathbf{x}} + \mathbf{c}\dot{\mathbf{x}} + \mathbf{k}\mathbf{x} = \mathbf{f}. \tag{5.17}$$

Here, $\mathbf{x} = (x, y, z)$ stands for the translational displacement, $\mathbf{m}$, $\mathbf{c}$ and $\mathbf{k}$ are the diagonal matrix of mass, damping and stiff coefficients respectively.

Specifically, we solve (5.17) component-wisely in each direction which is taken as an instance to illustrate the numerical schemes used in the computations. The $x$-component of (5.17) is written as

$$m_x\ddot{x} + c_x\dot{x} + k_x x = f_x, \tag{5.18}$$

where $m_x$, $c_x$, $k_x$ and $f_x$ denote the coefficient and force regarding to $x$-component direction respectively. For the sake of simplicity, we decomposed the second-order ODE into a set of first-order ODEs with $v = \dot{x}$

$$\dot{v} = \frac{f_x - c_x v - k_x x}{m_x}, \tag{5.19}$$

$$\dot{x} = v, \tag{5.20}$$

in which the explicit Runge-Kutta scheme is used for time stepping of weak coupling.

The weak coupling scheme may suffer numerical instability especially when the density ratio of solid and fluid ($\rho_s/\rho_f$) is small. To achieve the sufficient robustness, Newmark scheme is used to treat the light solid bodies which is derived from Taylor expansion as following

$$\dot{x}^{n+1} = \dot{x}^n + (1-\gamma)\Delta t\ddot{x}^n + \gamma\Delta t\ddot{x}^{n+1}, \tag{5.21}$$

$$x^{n+1} = x^n + \Delta t\dot{x}^n + (\frac{1}{2} - \beta)\Delta t^2\ddot{x}^n + \beta\Delta t^2\ddot{x}^{n+1}, \tag{5.22}$$

where $\Delta t$ is the time step. The coefficients here are set as $\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$ since it is unconditional stable. The solution of (5.18) then can be formulated as

$$x^{n+1} = \frac{f_x^{n+1} + a_1 x^n + a_2\dot{x}^n + a_3\ddot{x}^n}{\hat{k}} \tag{5.23}$$

with coefficients given by

$$\begin{cases} a_1 = \frac{m_x}{\beta\Delta t^2} + \frac{c_x\gamma}{\beta\Delta t}, \\ a_2 = \frac{m_x}{\beta\Delta t} + c_x(\frac{\gamma}{\beta} - 1), \\ a_3 = (\frac{1}{2\beta} - 1)m_x + c_x\Delta t(\frac{\gamma}{2\beta} - 1), \\ \hat{k} = \frac{m_x}{\beta\Delta t^2} + \frac{c_x\gamma}{\beta\Delta t} + k_x. \end{cases} \tag{5.24}$$

And the translational velocity is calculated as the first order derivative of its displacement by

$$\dot{x}^{n+1} = \frac{\gamma}{\beta\Delta t}(x^{n+1} - x^n) + (1 - \frac{\gamma}{\beta})\dot{x}^n + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{x}^n. \tag{5.25}$$

Similarity, we can construct an alternative equation regarding to the rotational displacement $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ of the solid motion

$$\mathbf{M}\ddot{\boldsymbol{\theta}} + \mathbf{C}\dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta} = \mathbf{T}, \tag{5.26}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are coefficient matrices respectively.

After the transitional velocity $\mathbf{u} = \dot{\mathbf{x}}$ and the rotational velocity $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$ are obtained by solving (5.17) and (5.26), we end up with the displacement and velocity of the moving rigid body by considering both transitional and rotational motions as

$$\mathbf{x}_s = \mathbf{x} + \boldsymbol{\theta} \times (\mathbf{x} - \mathbf{x}_c), \tag{5.27}$$

$$\mathbf{u}_s = \mathbf{u} + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_c), \tag{5.28}$$

which satisfies the rigidity condition and will never cause any distortion on the solid body. Since the rigid body is assumed in this study, the displacement and velocity of the body $\mathbf{x}_s$ and $\mathbf{u}_s$ can be easily transformed to that of fluid-solid interface $\mathbf{x}_I$ and $\mathbf{u}_I$, which can be treated as boundary condition for fluid model.

Finally, we formulate the relation of the interface displacement $\mathbf{x}_I$ and the effect of fluid flow $\sigma$ as

$$\mathbf{x}_I = \mathcal{S}(\sigma), \tag{5.29}$$

where $\mathcal{S}$ is the numerical model of solid motion. It represents that the interface position is associated with the force or torque exerted on the rigid body.

## 5.4 Fluid Solid Coupling

As aforementioned, partitioned approach is preferred to solve the fluid-structure interaction in this study due to its modularity and the possibility of re-using existing solvers which reduce the code development time by taking advantage of the available code or numerical algorithms that have been validated and used for many fluid and structural problems. In the partitioned approach, fluid and the structure equations are solved separately with respective numerical algorithms and discretization method, and interface coupling is enforced to communicate information between the fluid and structure solutions. To achieve accurate and efficient fluid-structure interaction, we present the both weakly coupled and semi-implicit coupled schemes for fluid-rigid body applications. To distinguish different domain, we denote $\mathbf{x}_f$ for the fluid mesh and $\mathbf{x}_s$ for the solid displacement. At time $t^n$, we define the computational variables $\mathbf{u}^n, p^n, \mathbf{x}_f^n$ for the fluid and $\mathbf{x}_s^n, \mathbf{v}_s^n$ for the solid, and then update them by integrating the fluid and structural solvers described above via weak or semi-implicit coupling scheme which will be detailed in the rest of this section.

### 5.4.1 Weak coupling

The basic and well-known weakly coupled scheme is the conventional serial staggered (CSS) scheme [113]. The simplicity and high efficiency makes it popular and be widely used in practical applications. The solution procedure of this scheme is summarized as following:

1. Given computational variables at $t^n$: $\mathbf{u}^n, p^n, \mathbf{x}_f^n, \mathbf{x}_s^n, \mathbf{v}_s^n$.
2. Evaluate the force $\mathbf{f}^n$ and torque $\mathbf{T}^n$ by (5.6) and (5.7) from fluid variables.
3. Update the solid displacement and velocity to next step $\mathbf{x}_s^{n+1}, \mathbf{v}_s^{n+1}$ by (5.27) and (5.28).
4. Update fluid mesh position $\mathbf{x}_f^{n+1}$ by RBF interpolation (4.19).
5. Update grid velocity of fluid domain: $\mathbf{u}_g^{n+1} = \left(\mathbf{x}_f^{n+1} - \mathbf{x}_f^n\right)/\Delta t$.
6. Solve Navier-Stokes equation: $\mathcal{NS}(\mathbf{u}^{n+1}, \mathbf{u}^n, p^{n+1}, \mathbf{x}_f^{n+1}, \mathbf{x}_f^n, \mathbf{u}_g^{n+1}) = 0$ with GCL condition, and then obtain $\mathbf{u}^{n+1}, p^{n+1}$.
7. Up to now, we have all the computational variables at $t^{n+1}$, then go to next step calculation.

In this procedure, Navier-Stokes equation $\mathcal{NS}(\mathbf{u}^{n+1}, \mathbf{u}^n, p^{n+1}, \mathbf{x}_f^{n+1}, \mathbf{x}_f^n, \mathbf{u}_g^{n+1}) = 0$ represents whole moving-mesh numerical model for fluids where $\mathbf{u}^n, \mathbf{x}_f^n, \mathbf{x}_f^{n+1}$ and $\mathbf{u}_g^{n+1}$

are input values and $\mathbf{u}^{n+1}$ and $p^{n+1}$ are variables obtained in the new time step. In this study, most validations are computed based on the weak coupling scheme due to its relatively low computational cost since the fluid and solid equations are coupled once in the calculation of each time step. However, when the coupling intensity is strong due to the small solid/fluid ratios, the instability was observed in many published works [60, 114] which needs a semi-implicit scheme to stabilize and accelerate the computations.

### 5.4.2 Semi-implicit coupling

To remedy the deficiency of weak coupling schemes, implicit coupling schemes are always used in FSI simulations to ensure the fluid and solid be updated at same time. Based on implicit scheme, iterative steps are usually performed until both fluid and solid variables converge in the next time step in the framework of partitioned approach. The key work of iterations is always settled in finding the ultimate equilibrium interface position $\mathbf{x}_I$ which is initialized by the computed solid position. Based on the interface position at next time step $\mathbf{x}_I^{n+1}$, the force or the torque on the body can be evaluated by (5.16). Then, solid equation is solved to update the solid displacement, namely, the fluid-solid interface, by (5.29).

Therefore, one closed loop for FSI is reduced to

$$\tilde{\mathbf{x}}_I^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_I^{n+1}). \tag{5.30}$$

We can find that the fully implicit schemes for both fluid and solid equations are required for obtaining a strong FSI scheme.

In this study, we follow a semi-implicit scheme where only the pressure term of fluid is implicitly coupled with the structure and the remaining terms of fluid equations are computed explicitly. Thus, the coupling process can be straightforwardly constructed based on the projection method for fluid equations as presented in section 5.2.2 and Newmark scheme for solid equation in section 5.3. It is observed that the pressure term is responsible for the added-mass effect and numerical instability is more attributed to the explicit coupling of pressure term [63, 115]. We construct the semi-implicit coupling scheme based on fixed-point methods [116] in this study. A block Gauss-Seidel iteration can be formulated as

$$\tilde{\mathbf{x}}_I^{n+1,l+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_I^{n+1,l+1}), \tag{5.31}$$

where $l$ stands for the sub-iterative step. The Aitken's dynamics relaxation method [117] is used to stabilize and accelerate the computation by

$$\mathbf{x}_I^{n+1,l+1} = \mathbf{x}_I^{n+1,l} + \lambda^l \mathbf{r}^{l+1}, \tag{5.32}$$

with $\mathbf{r}^{l+1} = \tilde{\mathbf{x}}_I^{n+1,l+1} - \mathbf{x}_I^{n+1,l}$ and $\lambda^l = -\lambda^{l-1}\left(\mathbf{r}^{l+1}(\mathbf{r}^{l+1} - \mathbf{r}^l)\right) / |\mathbf{r}^{l+1} - \mathbf{r}^l|^2$ . In this study, the interface velocity is then calculated by

$$\dot{\mathbf{x}}_I^{n+1,l+1} = \frac{\gamma}{\beta\Delta t}(\mathbf{x}_I^{n+1,l+1} - \mathbf{x}_I^n) + (1 - \frac{\gamma}{\beta})\dot{\mathbf{x}}_I^n + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\mathbf{x}}_I^n \tag{5.33}$$

for the body with translational degrees of freedom, which is used as the boundary condition of fluid flows. For complex solid motion involving both translation and rotation, the interface velocity may be simply approximated by

$$\dot{\mathbf{x}}_I^{n+1,l+1} = \frac{\mathbf{x}_I^{n+1,l+1} - \mathbf{x}_I^n}{\Delta t}. \tag{5.34}$$

The numerical procedure of semi-implicit coupling scheme devised in this study is presented in Figure 5.3. The convergence criterion is set as

$$\mathbf{x}_I^{n+1,l+1} - \mathbf{x}_I^{n+1,l} < \epsilon \quad \& \quad \dot{\mathbf{x}}_I^{n+1,l+1} - \dot{\mathbf{x}}_I^{n+1,l} < \epsilon, \tag{5.35}$$

where $\epsilon$ is the tolerance error.

It is noted that since the point values at vertices of mesh cells are updated as the computational variables and ready to use at every time step, the coupling between different materials can be directly computed without extra interpolation procedure. Hence, the present ALE formulation based on multi-moment finite volume method provides an accurate and efficient numerical framework for fluid-solid interaction.

## 5.5 Numerical tests

### 5.5.1 Accuracy test

We evaluate the convergence rate of the present scheme by computing the advection transport of a sine function whose initial profile is given as

$$\phi_0(x, y) = \sin 2\pi(x + y) \tag{5.36}$$

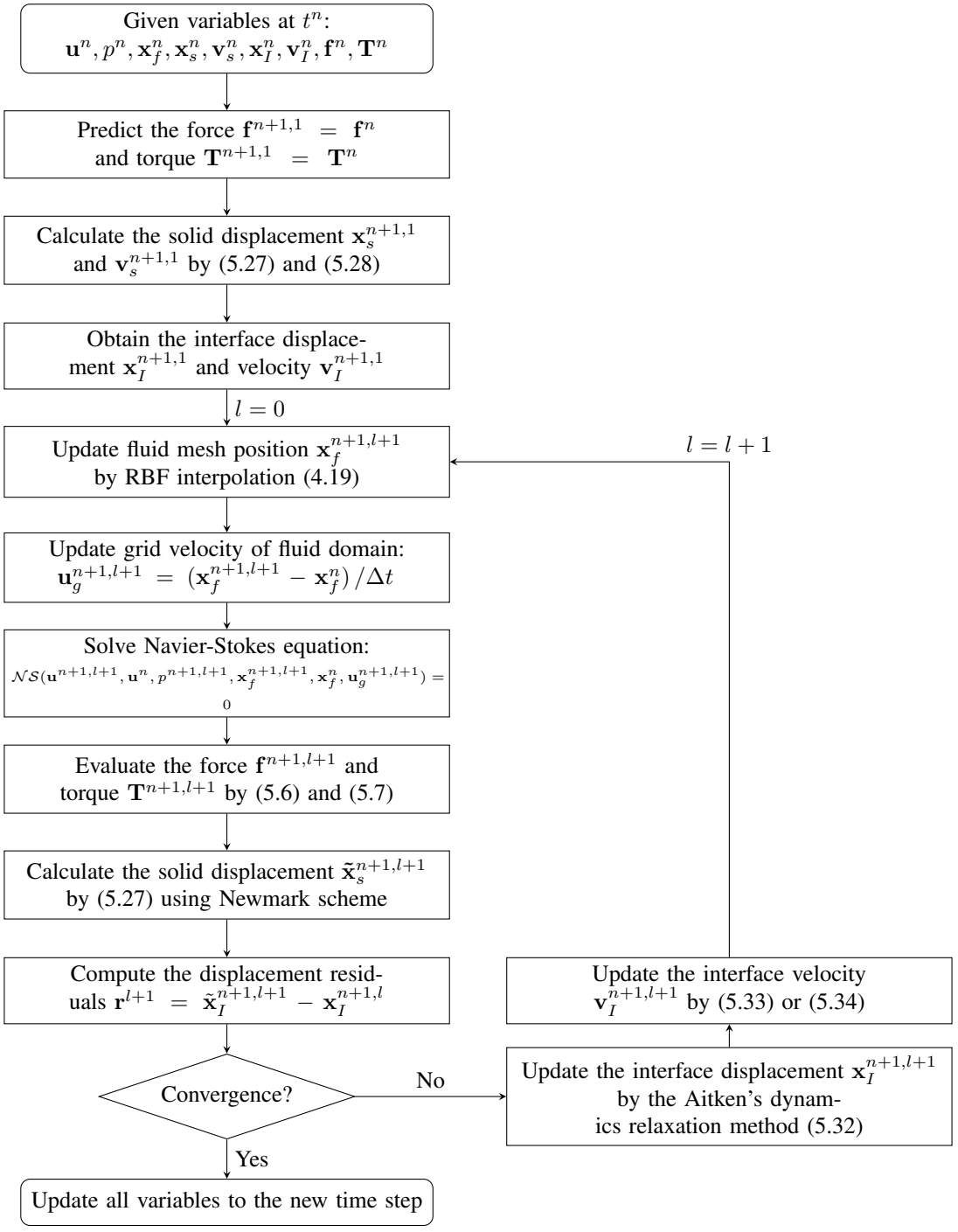



FIGURE 5.3: Flowchart of semi-implicit coupling scheme for fluid-rigid body interaction.

over a computational domain of $[0,1] \times [0,1]$. We use gradually refined triangular elements which are obtained by uniformly dividing each square element of Cartesian grid into two triangles. The constant advection velocity is specified as $\mathbf{u} = (1,1)$. The 3rd-order Runge-Kutta scheme is used for time integration and the time step is given as $0.08 \Big/ \sqrt{N/2}$ where $N$ denotes the cell numbers in each grid. The periodic condition is prescribed for all boundaries of computational domain. We perform the calculation up to time $t = 1$ in ALE form with the grid velocity $\mathbf{u}_g = (u_g, v_g)$ which is prescribed by zero for the case of static mesh by

$$u_g(\mathbf{x},t) = X_0 \cdot 2\pi n_t/t_0 \cos(2\pi n_t t/t_0)\sin(2\pi n_x x/L_x)\sin(2\pi n_y y/L_y), \quad (5.37)$$

$$v_g(\mathbf{x},t) = Y_0 \cdot 2\pi n_t/t_0 \cos(2\pi n_t t/t_0)\sin(2\pi n_x x/L_x)\sin(2\pi n_y y/L_y), \quad (5.38)$$

for the moving mesh case with parameters given as $X_0 = 1, Y_0 = 1, n_t = 4, n_x = 1, n_y = 1, t_0 = 2, L_x = 10, L_y = 10$.

We summarize $L_1$ and $L_\infty$ errors along with their convergence rates by using present method on static and moving meshes as shown in Table 5.1. The elapsed time of computation is measured on a PC with an Intel Core i7-4790 CPU @ 3.60GHz. It is observed that uniformly 3rd-order accuracy is obtained with the same level of numerical errors between two cases which demonstrate the validity and efficiency of proposed ALE scheme. For the sake of comparison, we also compute with the same conditions by a conventional 3rd order finite volume method (FVM) [118] where the quadratic polynomial for spatial reconstruction is performed by using least-square approximations on a merged stencil. The 3rd order accuracy is also produced as shown in Table 5.2 however at expense of larger computational cost.

TABLE 5.1: Errors and convergence rates for 2D sine wave computed by VPM method in ALE form respectively.

| Cells | Static mesh with $\mathbf{u}_g = \mathbf{0}$ | | | | | Moving mesh with $\mathbf{u}_g$ by (5.37) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ error | Rate | $L_\infty$ error | Rate | Time(s) | $L_1$ error | rate | $L_\infty$ error | rate | Time(s) |
| 200 | $2.80 \times 10^{-2}$ | – | $4.47 \times 10^{-2}$ | – | 0.08 | $3.65 \times 10^{-2}$ | – | $7.44 \times 10^{-2}$ | – | 0.43 |
| 800 | $3.66 \times 10^{-3}$ | 2.94 | $5.76 \times 10^{-3}$ | 2.96 | 0.55 | $5.16 \times 10^{-3}$ | 2.82 | $1.22 \times 10^{-2}$ | 2.60 | 3.07 |
| 3200 | $4.66 \times 10^{-4}$ | 2.97 | $7.36 \times 10^{-4}$ | 2.97 | 4.31 | $6.73 \times 10^{-4}$ | 2.94 | $1.68 \times 10^{-3}$ | 2.86 | 23.32 |
| 12800 | $5.87 \times 10^{-5}$ | 2.99 | $9.25 \times 10^{-5}$ | 2.99 | 40.32 | $8.58 \times 10^{-5}$ | 2.97 | $2.25 \times 10^{-4}$ | 2.90 | 192.58 |
| 51200 | $7.36 \times 10^{-6}$ | 3.00 | $1.16 \times 10^{-5}$ | 3.00 | 329.21 | $1.07 \times 10^{-5}$ | 3.00 | $2.66 \times 10^{-5}$ | 3.09 | 1440.77 |

From the numerical results of Table 5.1 and Table 5.2, we can find that the VPM method can achieve 3rd order accuracy with smaller numerical errors than the conventional FVM regardless of moving mesh or not. On the other hand, since two kinds of moments are employed as prognostic variables, the VPM method needs less computational cost and algorithmic complexities than that of FVM which may be associated with the

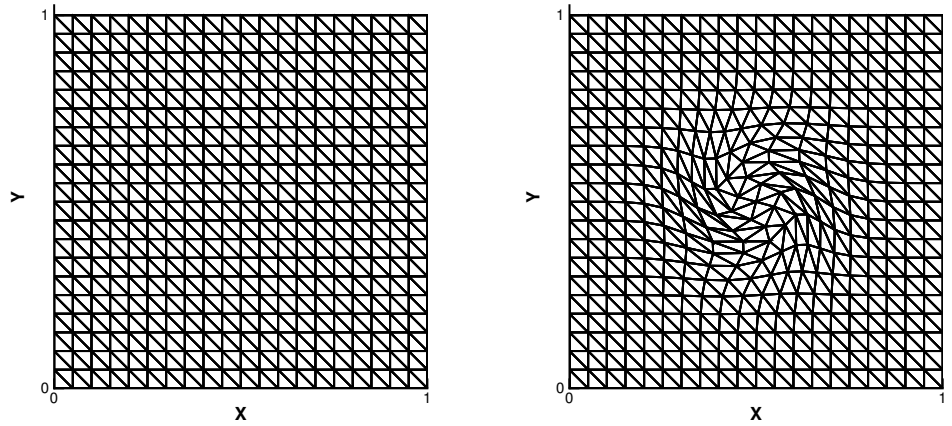
TABLE 5.2: The same as Table 5.1 but computed by conventional FVM method.

| Cells | Static mesh with $\mathbf{u}_g = \mathbf{0}$ | | | | | Moving mesh with $\mathbf{u}_g$ by (5.37) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $L_1$ error | Rate | $L_\infty$ error | Rate | Time(s) | $L_1$ error | rate | $L_\infty$ error | rate | Time(s) |
| 200 | $7.58 \times 10^{-2}$ | – | $1.18 \times 10^{-1}$ | – | 0.11 | $9.01 \times 10^{-2}$ | – | $1.54 \times 10^{-1}$ | – | 0.60 |
| 800 | $1.02 \times 10^{-2}$ | 2.89 | $1.60 \times 10^{-2}$ | 2.89 | 0.69 | $1.34 \times 10^{-2}$ | 2.75 | $2.80 \times 10^{-2}$ | 2.46 | 4.11 |
| 3200 | $1.29 \times 10^{-3}$ | 2.98 | $2.03 \times 10^{-3}$ | 2.98 | 5.95 | $1.77 \times 10^{-3}$ | 2.92 | $4.07 \times 10^{-3}$ | 2.78 | 31.49 |
| 12800 | $1.62 \times 10^{-4}$ | 3.00 | $2.55 \times 10^{-4}$ | 3.00 | 55.29 | $2.25 \times 10^{-4}$ | 2.98 | $5.26 \times 10^{-4}$ | 2.95 | 247.39 |
| 51200 | $2.03 \times 10^{-5}$ | 3.00 | $3.18 \times 10^{-5}$ | 3.00 | 438.70 | $2.82 \times 10^{-5}$ | 3.00 | $6.61 \times 10^{-5}$ | 2.99 | 1933.69 |

computation of inverse matrix. The comparison proves that the VPM method significantly outperforms the 3rd-order FVM method for implementing ALE schemes in term of numerical accuracy and efficiency.

### 5.5.2 Lid-driven cavity flow

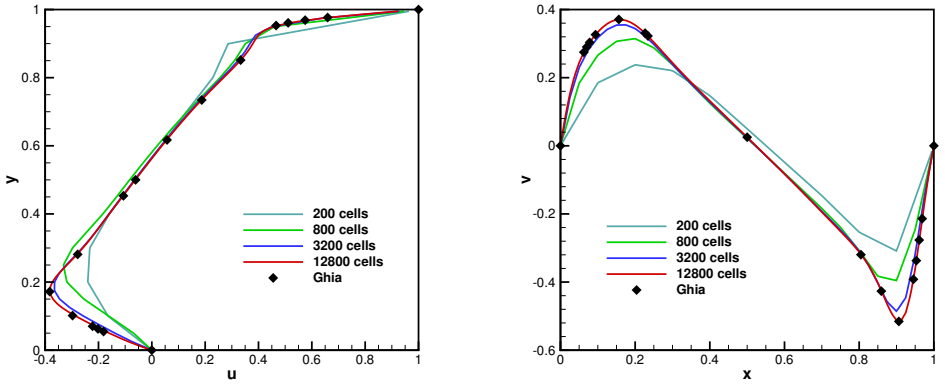
Lid-driven cavity flow is a widely used benchmark test for numerical codes designed to solve viscous compressible flow [119]. Incompressible flow is enclosed in a square domain $[0, 1] \times [0, 1]$. Non-slip conditions are imposed on the 4 walls. The upper wall $y = 1$ is moving at a constant speed $\boldsymbol{u} = (1, 0)$. Reynolds number is set as $Re = 1000$. Numerical tests are initially carried out on the uniform triangular mesh of different mesh resolutions: $200, 800, 3200$, and $12800$ cells. Figure 5.4 (a) shows the initial mesh with $800$ elements. We run the cases up to time $t = 100s$ and plot the numerical results on




(a) t=0s (b) t=120s

FIGURE 5.4: Computational mesh at t=0s and t=120s for the case with 800 cells.

different grid resolutions in Figure 5.5. Figure 5.5 (a) and (b) display $u$ profiles along $x = 0.5$ and $v$ profiles along $y = 0.5$ at time $t = 100s$ on grids of different resolutions

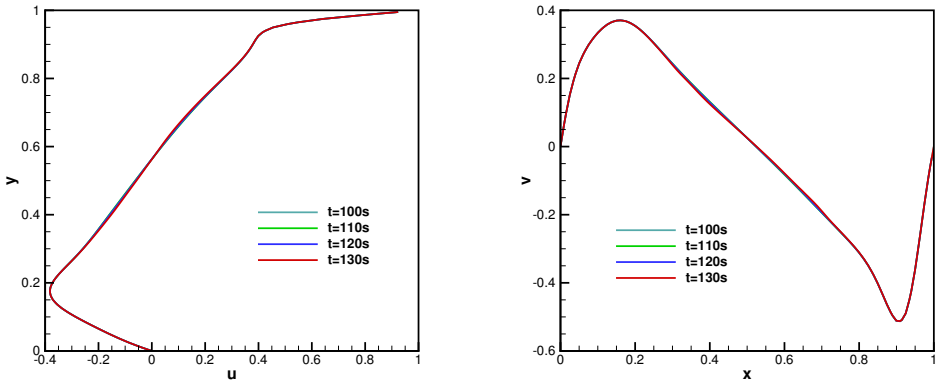



(a) (b)

FIGURE 5.5: (a) u profiles along $x = 0.5$, (b) v profiles along $y = 0.5$ at time $t = 100s$ on grids of different resolutions.

respectively. It is clear that as the grids refined, the velocity profile converges to Ghia's results [119].

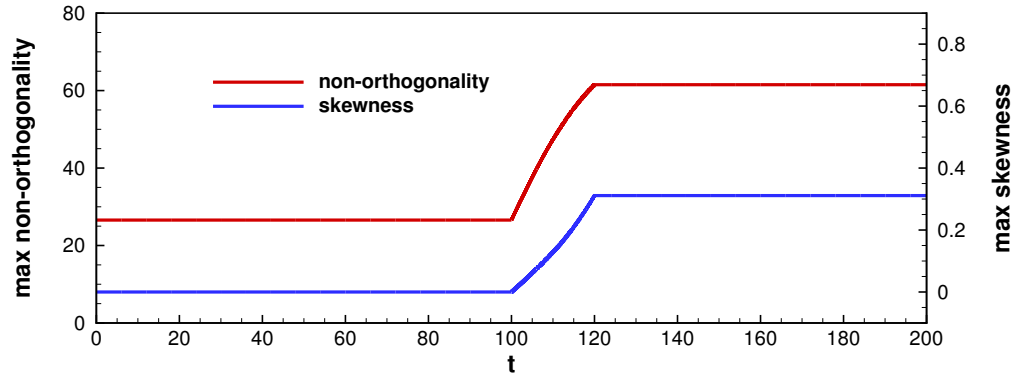
To verify the performance of present solver on moving grids, we move the computational grids from $t = 100s$ where the fluid flow is observed to be in a steady state. The grid velocity is set as $\boldsymbol{u}_g = (-0.01(y-0.5)e^{2-50r^2}, 0.01(x-0.5)e^{2-50r^2})$ with $r^2 = (x-0.5)^2 + (y-0.5)^2$. As shown in Figure 5.4 (b), mesh is heavily distorted in the central area at time $t = 120s$. We plot the velocity profiles again at different time sequences in Figure 5.6. We can find that mesh movement almost has no impact for the numerical solution.



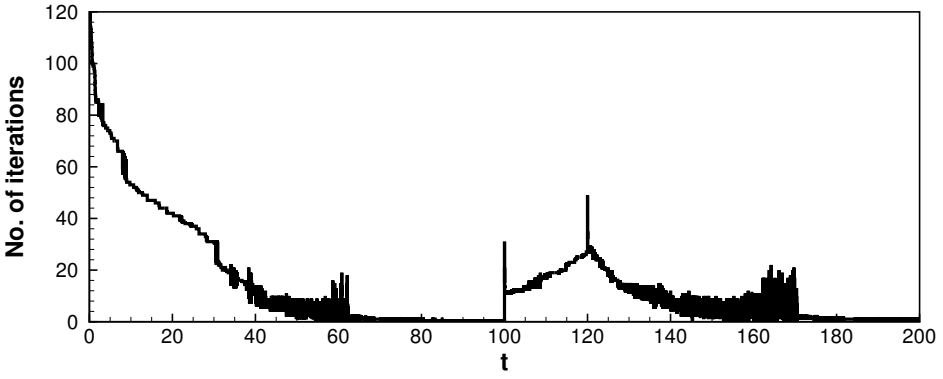

(a) (b)

FIGURE 5.6: (a) u profiles along $x = 0.5$, (b) v profiles along $y = 0.5$ at different time sequences on the mesh with 12800 cells.

In this computation, Poisson equation is solved by the preconditioned conjugate gradient (PCG) method with diagonal-based incomplete cholesky (DIC) preconditioner. We also investigate the effect of mesh quality on the iterative numbers for solving Poisson equation, since it serves as the major part of computational cost. The iterative tolerance is set as $1e^{-6}$. The test is performed up to $200s$ for the case with $800$ cells, where computational grid is rotated between $100 - 200s$ and keeps stationary for other periods. Figure 5.7(a) plots the time history of mesh quality, which is evaluated based on non-orthogonality and skewness, as defined in Appendix D. The mesh quality gets worse from $100s$ to $120s$, since the mesh rotates in the domain center. We show the time history of iterative numbers in Figure 5.7 (b). In the initial stage of computation where mesh keeps stationary, the iterative number decreases with time and finally reduces to only $0$ or $1$ iteration. When the mesh deforms, the iterative number increases but still in an acceptable value. Mesh quality reaches to maximum non-orthogonality of $61.52^o$




(a)




(b)

FIGURE 5.7: Time history of mesh quality (a) and iterative numbers (b) for the case with 800 cells.

and skewness of $0.31$ at time $t = 120s$. After that, the iterative number again decreases

and finally still needs only $0$ or $1$ iterations since the mesh keeps still.

### 5.5.3 Oscillating cylinder in quiescent water

The numerical model for incompressible Navier-Stokes equations on moving grids is firstly validated by a test case about a circular cylinder oscillating in a quiescent fluid. The cylinder with the diameter of unit ($D = 1$) is forced to have a harmonic oscillation in the y-direction. The analytical displacement of cylinder is given by

$$y(t) = -A\sin(2\pi ft), \tag{5.39}$$

where $A$ is the amplitude of the motion and $f$ is the frequency. As reported in the literature [3, 120–122], the phenomenon is characterized by two non-dimensional parameters, the Reynolds number $Re = \frac{U_{\max}D}{\nu}$ and the Keulegan-Carpenter number $KC = \frac{U_{\max}}{fD}$, where $U_{\max}$ is the maximum velocity of the cylinder motion. The parameters are set as $Re = 100$ and $KC = 5$ in this calculation. The computational domain is set as a circular surface of diameter $50D$ with the circular cylinder located at the center of domain initially. We partition the computational domain into 12,624 triangular elements in total with 120 and 80 segments around the cylinder and the outer far field boundary respectively. Since the cylinder moves in a prescribed velocity, the boundary conditions of the moving interface can be set exactly.

We present a time sequence of flow patterns in Figure 5.8, which show a good agreement with results in published works [3, 120]. Figure 5.9 plots the fluid velocities at four cross-sections for three different phases, which agrees well with experiments [3]. Quantitative analysis is performed on the time variation of inline force, which is compared with experimental results of Dütsch in Figure 5.10. A least-square fitting is calculated for the inline force to obtain the drag and added-mass coefficients ($c_d$ and $c_i$). The computed coefficients are compared with some existing works as shown in Table 5.3. Both qualitative and quantitative analyses prove the feasibility of present numerical

TABLE 5.3: The comparison of the drag and added-mass coefficients.

| | $c_d$ | $c_i$ |
|---|---|---|
| present | 2.07 | 1.46 |
| Dütsch et al. [3] | 2.09 | 1.45 |
| Su, Cao & Zhao [120] | 2.08 | 1.51 |

model on moving domain.

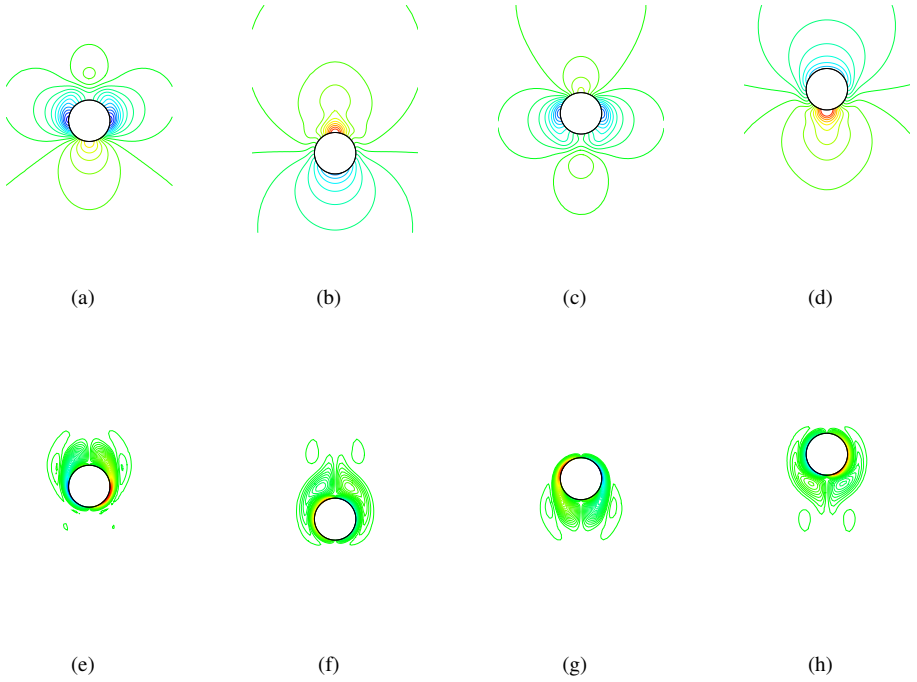



FIGURE 5.8: A time sequence of pressure isolines and vorticity isolines around the cylinder. (a) (b) (c) and (d) are pressure isolines at phase position $0^o, 96^o, 192^o$ and $288^o$ respectively. (e) (f) (g) and (h) are corresponding vorticity isolines.

### 5.5.4 Vortex Induced Vibration (VIV) of a circular cylinder

In this section, we validate the numerical model for FSI problems which concerns an elastically mounted circular cylinder in a free-stream. The cylinder is allowed to have two translational degrees of freedom. Supposing that the spring system has the same specification in each direction, the cylinder motion can be expressed as

$$m_s\ddot{\mathbf{x}} + c_s\dot{\mathbf{x}} + k_s\mathbf{x} = \mathbf{f}, \tag{5.40}$$

where $\mathbf{x}$ is the displacement of cylinder center, $\mathbf{f}$ the force evaluated from fluid flow, $m_s$ the mass per unit length of solid body, $c_s$ and $k_s$ the damping constant and the stiffness constant respectively. To make comparisons with published works [123, 124], we define the non-dimensional parameters as follows. The mass ratio $m^* = \frac{m_s}{\rho D^2}$, the reduced velocity $U^* = \frac{U_\infty}{f_s D}$, the natural vibration frequency of the cylinder $f_s = \frac{1}{2\pi}\sqrt{\frac{k_s}{m_s}}$, and the damping ratio $\zeta_s = \frac{c_s}{2\sqrt{k_s m_s}}$. Here, $D$ is the diameter of cylinder, $U_\infty$ the free-stream speed. We set $D = 1$, $U^* = 5$, $\zeta_s = 0.01$ and $Re = 200$ in the simulation.

We first calculated the case that fluid and solid body have similar density with $m^* = 1$ by the weakly coupled scheme for computational efficiency. The computational domain

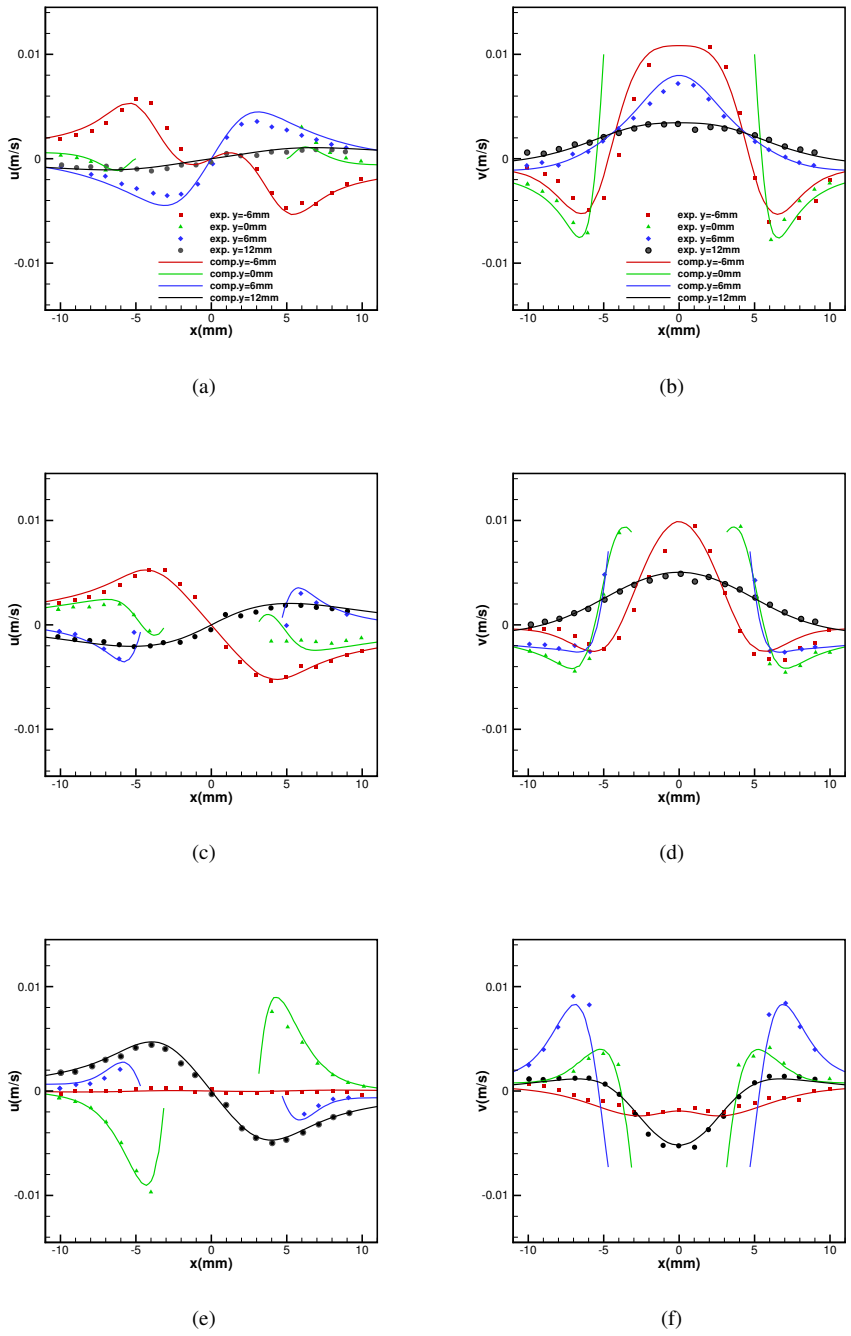



FIGURE 5.9: Comparisons of numerical and experimental results of x-component (left column) and y-component (right column) of fluid velocity in four different cross-sections at three phases: (1) $180^o$: (a) (b); (2) $210^o$: (c) (d); (3) $330^o$: (e) (f).

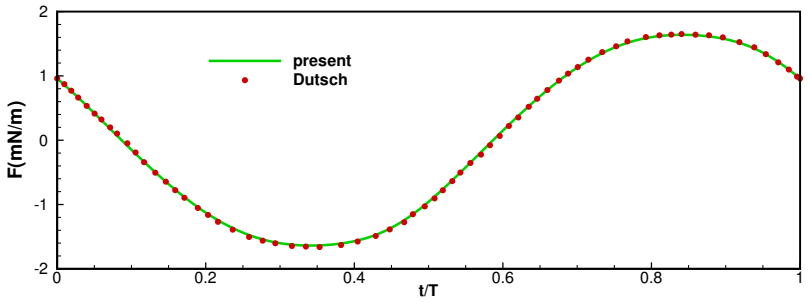



FIGURE 5.10: Comparison of inline force between present results and Dütsch's results [3].

is set as $[-10D, 30D] \times [-10D, 10D]$ and the cylinder is located at the origin initially. The velocity of the inlet flow from the left boundary $U_\infty$, and a outlet condition is set to the right boundary. The slip boundary condition is set on the top and bottom walls. The simulation is carried out up to time $t = 100$ on three kinds of grids: grid A, grid B and grid C where 160, 320 and 640 grid nodes are uniformly distributed around the cylinder, respectively. For saving computational cost, triangular elements with relatively small sizes are clustered around the cylinder and its downstream region to accurately resolve the fluid/solid interaction and the vortex shedding, while coarse mesh is generated in the region far from the cylinder where the flow has less variation in space. To achieve the synchronization regime of VIV, the natural frequency of cylinder $f_s$ is set as same as the frequency of the vortex shedding. In the initial stage of computation, the symmetrical wake forms around the cylinder which is pushed forward by the free stream. With time evolving, the symmetry breaks due to the instability, and finally vortex shedding forms and interacts with the periodically moving cylinder. Figure 5.11 gives a snapshot of vorticity contours for the free oscillating cylinder, which is a classic 2S mode (two single vortices per cycle of motion) [125].

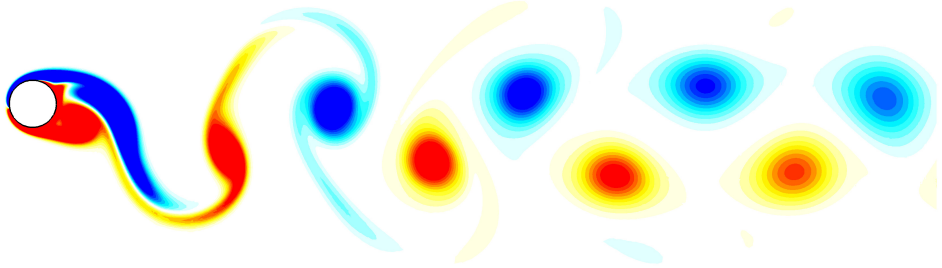


FIGURE 5.11: A snapshot of vorticity contours for a free oscillating cylinder in a free stream at $Re = 100$ and $m^* = 1$.

As the vortex shedding develops, the cylinder moves in a periodic trajectory. Figure 5.12 (a) presents the displacement trajectory of the cylinder computed on different meshes. With the gradually refined mesh, the displacement trajectories converged to an eight-shape and the equilibrium position of the oscillating cylinder is at $x = 0.64D$, which is close to Blackburn's results ($x = 0.62D$) and Yang and Stern's prediction ($x = 0.65D$). Displacement-velocity phase displayed in 5.12 (b) and (c) present the convergence trend. We also plot the time history of drag and lift coefficients, as shown in Figure 5.13.

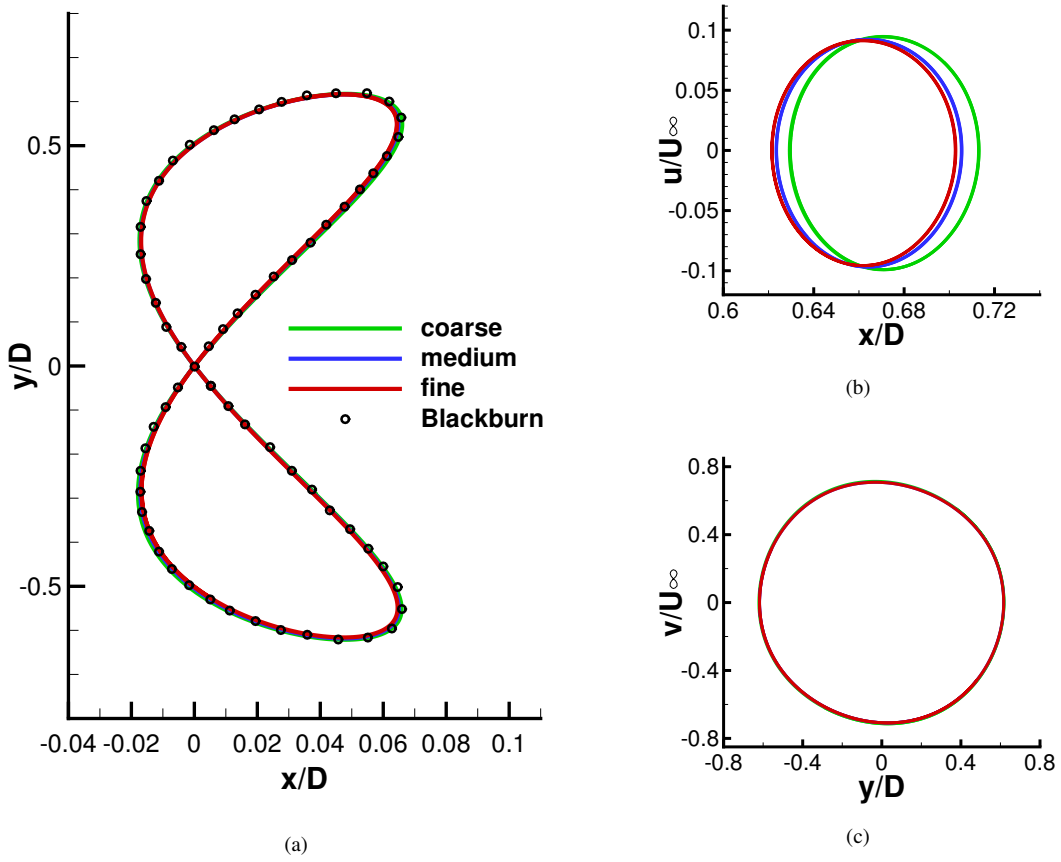



FIGURE 5.12: Displacement diagram of a free oscillating cylinder in a free-stream of Re=200. Numerical results on grid A (coarse), grid B (medium) and grid C (fine) are compared with Blackburn's results [4] where a spectral element method is used.

Semi-implicit interface coupling scheme is used for computation of VIV with small mass ratio $m^* = 0.5$, since the weakly coupled scheme is unstable for light-body-involved free oscillation. The simulation is performed on grid B and the same boundary condition is specified. We perform the simulations by setting different tolerance error. The trajectories of cylinder center are plotted in Figure 5.14. It is observed that the numerical result converges with the tolerance error smaller than $1e-5$. We then

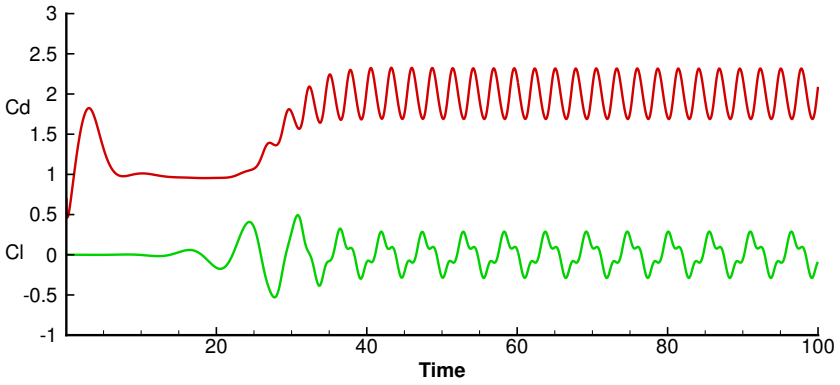



FIGURE 5.13: Drag and lift force coefficients for a free oscillating cylinder in a free-stream at Re=200.

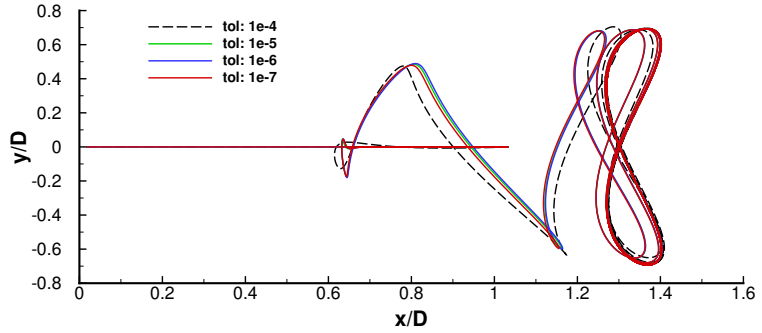



FIGURE 5.14: Trajectories of cylinder center with different tolerance error.

compare the iteration numbers of semi-implicit coupling as shown in Figure 5.15 and Figure 5.16. Figure 5.15 plots the iteration numbers of semi-implicit coupling at the initial stage of the computation (0-0.25s). We can find that the iteration numbers rapidly reduce to acceptable values. Meanwhile, we can observe that the smaller tolerance error requires more iterations. Figure 5.16 shows the iteration numbers for whole computation times. After the initial stage, the simulation just takes a few iterations by setting different tolerance value, which justifies the efficiency of present semi-implicit coupling scheme.

Figure 5.17 plots the displacement and velocity phase diagrams of both cases ($m^* = 1$ and $m^* = 0.5$) for comparison. The equilibrium position for the lighter body is driven far away which can be clearly observed in Figure 5.17 (a). The phase diagrams are quite similar to the simulation of Liu and Hu [124] by using immersed boundary method.

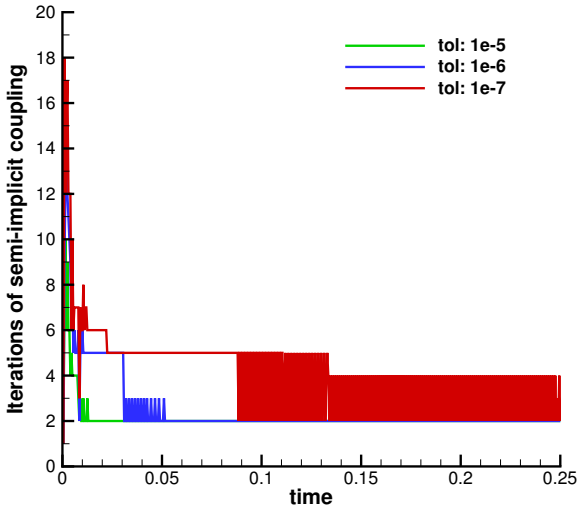
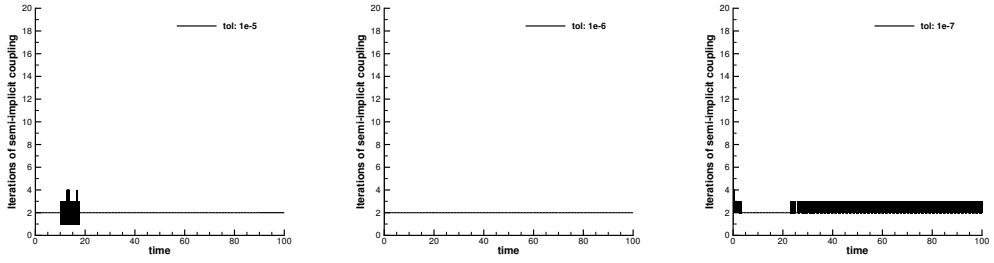



FIGURE 5.15: Comparisons of iteration numbers for semi-implicit coupling by setting different tolerance error at the initial stage of computation (0-0.25s).




(a) (b) (c)

FIGURE 5.16: Iterative numbers for semi-implicit coupling by setting different tolerance error during whole computation time (0-100s). (a) $\epsilon = 1e-5$, (b) $\epsilon = 1e-6$, (c) $\epsilon = 1e-7$.

### 5.5.5 Galloping rectangular bodies

In addition to vortex-induced vibration (VIV), galloping is also a well-known fluid-structure interaction phenomenon. For VIV, the natural frequency of structure is comparable to the vortex shedding frequency, and it is always characterized with synchronization or lock-in phenomenon. However, galloping is the flow-induced vibration that happens at frequencies much lower than the wake vortex shedding frequency behind the structure which always associates with a non-circular structure in a flow with a large incoming velocity. We limit our numerical tests to case where there is only one DOF in the structure motion, i.e. transverse or rotational galloping. We are interested in the galloping phenomenon of solid bodies shaped with a rectangular cross-section, which is

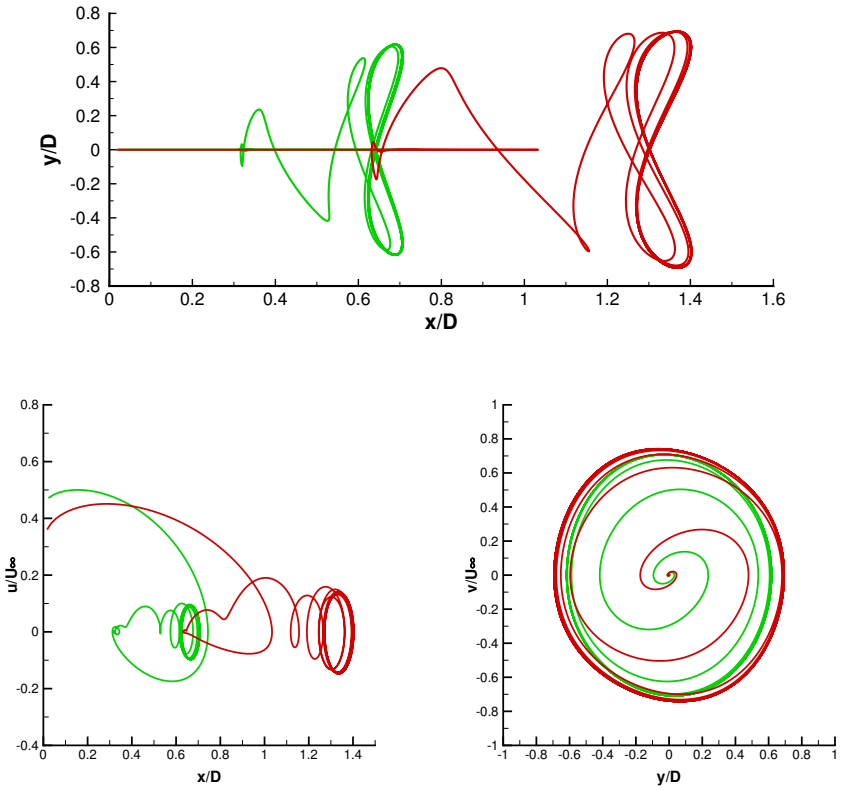



FIGURE 5.17: The displacement and velocity phase diagrams for a free oscillating cylinder in a free-stream at $Re = 200$. — is for mass ratio $m^* = 1$ and — for mass ratio $m^* = 0.5$.

calculated for validating our FSI numerical model. The rectangular cross-section bodies are characterized with the thickness ratio $\Lambda = \frac{L}{D}$, where $L$ is the length of the body and D its thickness ($D = 1$).

#### 5.5.5.1 Transverse galloping

A fixed rectangular cross-section body with $\Lambda = 1$ is firstly put in the free-stream of Reynolds number $Re = 250$. The computational domain and boundary conditions are set as same as in section 5.5.4. The triangular mesh is also refined around the cylinder and the segments located on the rectangular body are of size $D/60$. The mass ratio $m^* = \frac{m_s}{\rho D^2}$ is set as 20 for comparison with [5]. We carry out the numerical calculation up to time $t = 1000$, and a snapshot of periodic vortex shedding is plotted in Figure 5.18.

For transverse galloping, the motion for the solid body is also modelled as a mass-spring-damping system, but with one degree of freedom motion in y-direction. Thus, it

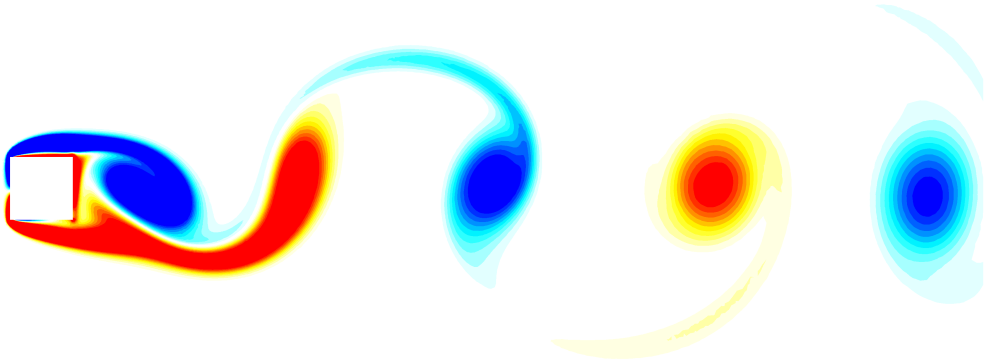


FIGURE 5.18: Vorticity contour of a fixed rectangular cross-section body with $\Lambda = 1$ in a free-stream at $Re = 250$.

can be formulated as

$$m_y\ddot{y} + c_y\dot{y} + k_y y = f_y, \tag{5.41}$$

where $y$ and $f_y$ are the y-component of the solid body displacement and the force exerted from fluid flow respectively. The non-dimensional parameters are: the reduced velocity $U_y = 40$, the damping ratio $\zeta_y = 0.0037$ and the natural frequency $f_y = 0.025$ which is much lower than the vortex shedding frequency. Figure 5.19 presents the responds of the body in the cross flow, which has a good agreement with the simulation by a spectral element simulation [5]. The frequency of vortex shedding is 0.14. A snapshot

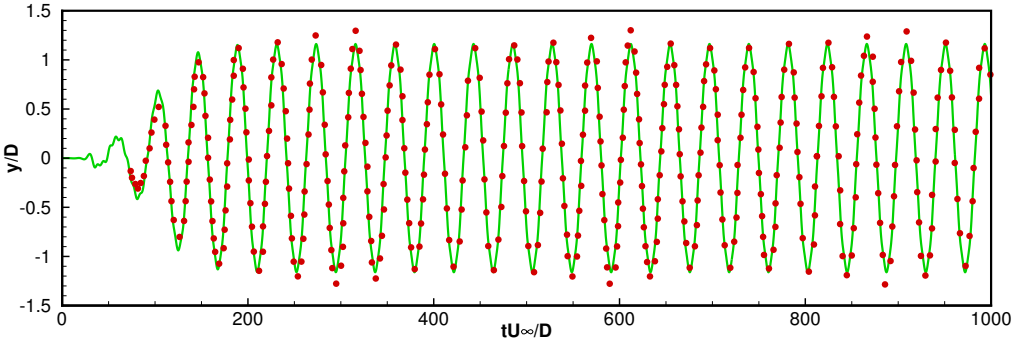



FIGURE 5.19: The y-component displacement of a square cross-section body $\Lambda = 1$ in a free-stream at $Re = 250$, $U_y^* = 40$, $\zeta_y = 0.0037$ and $m^* = 20$. — is the present simulation and • denotes the results by a spectral element simulation [5].

of vorticity contour is given in Figure 5.20.

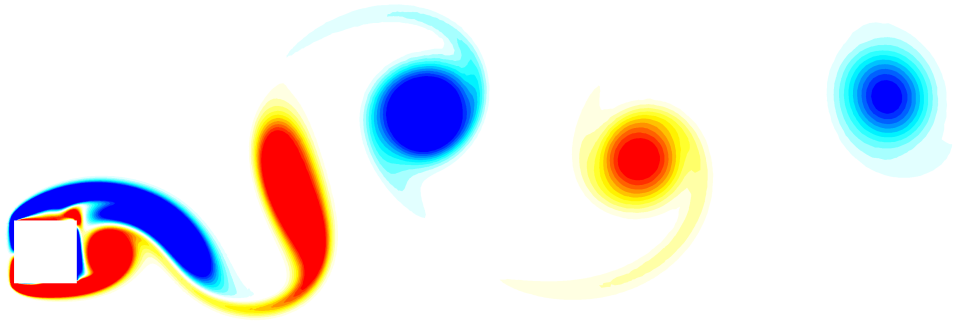


FIGURE 5.20: Vorticity contour of transverse galloping with $\Lambda = 1$ in a free-stream at $Re = 250$, $U_y^* = 40$, $\zeta_y = 0.0037$ and $m^* = 20$.

#### 5.5.5.2 Rotational galloping

One degree-of-freedom torsional model of a rectangular body can be formulated as

$$I_\theta \ddot{\theta} + c_\theta \dot{\theta} + k_\theta \theta = T_\theta \tag{5.42}$$

where $\theta$ is the rotational angle of the body around its center, $I_\theta = \frac{1}{12}\rho_s DL(D^2 + L^2)$ the moment of inertia of the body per unit length, $c_\theta$ the torsional damping constant, $k_\theta$ the torsional stiffness constant and $M_\theta$ the torque. To compare the results with [5], we define some non-dimensional parameters as: the mass moment of inertia ratio $I^* = \frac{I_\theta}{\rho D^4}$, the reduced velocity $U_\theta^* = \frac{U_\infty}{f_\theta D}$, the natural vibration frequency of the structure $f_\theta = \frac{1}{2\pi}\sqrt{\frac{k_\theta}{I_\theta}}$ and the damping ratio $\xi_\theta = \frac{c_\theta}{2\sqrt{k_\theta I_\theta}}$. In this calculation, we set parameters as: $\Lambda = 4$, $U_\infty = 1$, $Re = 250$, $U_\theta^* = 40$, $\xi_\theta = 0.25$ and $I_\theta^* = 400$.

The computational domain is set as same as the previous test with $[-10D, 30D] \times [-10D, 10D]$, and the same boundary conditions are prescribed. The domain is partitioned into 209,266 triangular elements and the size of element segments around the rectangular body is $D/60$. In the calculation, the rectangular cross-section body has one DOF rotational motion after the stable vortex shedding breaks. We measure the rotational displacement responds of the solid body and compare it with Robertson's simulation results [5] as shown in Figure 5.21. The body has the maximum rotational responds of $15^o$ and the rotational frequency is 0.0188.

We give the vorticity snapshots around the body during a clockwise rotation from the balanced position to the positive peak position in Figure 5.22 (a) (b) (c) and from the

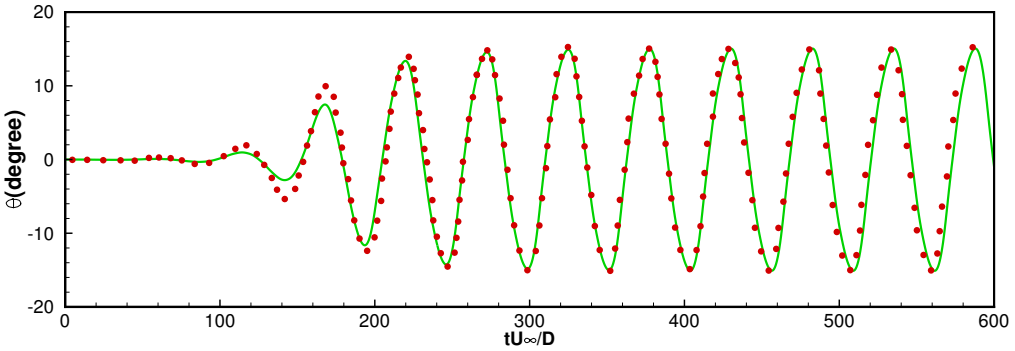
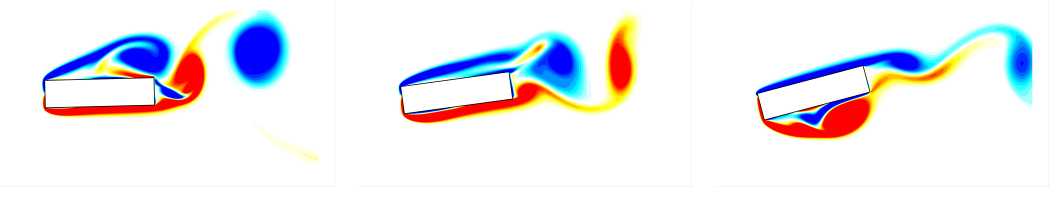
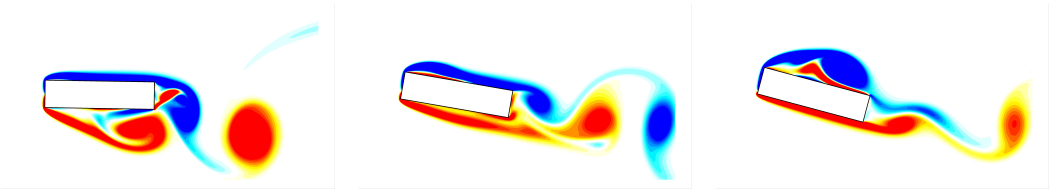



FIGURE 5.21: Rotational galloping responds of a rectangular cross-section body with $\Lambda = 4$ in a free stream at $Re = 250$, $U^*_\theta = 40$, $\xi_\theta = 0.25$ and $I^*_\theta = 400$. — is the present simulation and • denotes the result by a spectral element simulation [5].

balanced position to the negative peak position in Figure 5.22 (d) (e) (f). The boundary layer separates at the leading sharp corner of a leeward surface and then the complex vortices form around the leeward surface. Finally, vortices shed at the rear end of the rectangular body.



(a) $-0.3^o$ (b) $6.4^o$ (c) $14.9^o$



(d) $0.8^o$ (e) $-7.4^o$ (f) $-15.0^o$

FIGURE 5.22: Vorticity snapshots of a one DOF rotational rectangular cross-section body with $\Lambda = 4$ in a free stream at $Re = 250$, $U^*_\theta = 40$, $\xi_\theta = 0.25$ and $I^*_\theta = 400$.

#### 5.5.5.3 Two galloping bodies

Two interactive galloping bodies are calculated to present the capability of the present method for multiple bodies in this section. A transverse galloping body with $\Lambda = 1$ and a rotational galloping body with $\Lambda = 4$ which studied in the previous section are placed in a free stream in tandem arrangement. The free-stream is still set at the same Reynolds number with $Re = 250$. The parameters for the transverse galloping body with square

cross-section are: $U_y^* = 40$, $\zeta_y = 0.0037$ and $m^* = 20$. The rotational galloping body with rectangular cross-section body is placed behind the square cross-section body with $U_\theta^* = 40$, $\xi_\theta = 0.25$ and $I_\theta^* = 400$.

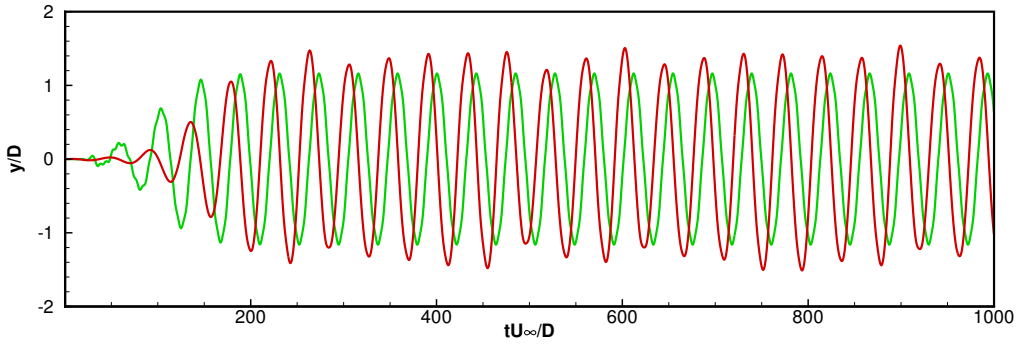



FIGURE 5.23: The transverse galloping responds of a square cross-section body $\Lambda = 1$ in a free-stream at $Re = 250$, $U_y^* = 40$, $\zeta_y = 0.0037$ and $m^* = 20$. — denotes a single body in a free stream and — denotes two bodies in a tandem arrangement.

To mitigate the boundary effect, a relatively larger computational domain with $[-20D, 60D] \times [-20D, 20D]$ is used. The size of element segments around the two bodies are still set as $D/60$. Figure 5.24 shows the comparison of the transverse galloping responds of a square cross-section body in the single-body and two-body configurations. For the upstream body, which is slightly affected by the downstream body, the transverse amplitude gets a little higher and the displacement profile becomes irregular for each period. The rotational body is significantly affected by vortices behind the upstream body. The rotational amplitude becomes obviously lower comparing to the case with a single rectangular body as shown in Figure 5.24. The trend of galloping amplitudes for both bodies is similar to Yang's simulation results [123].

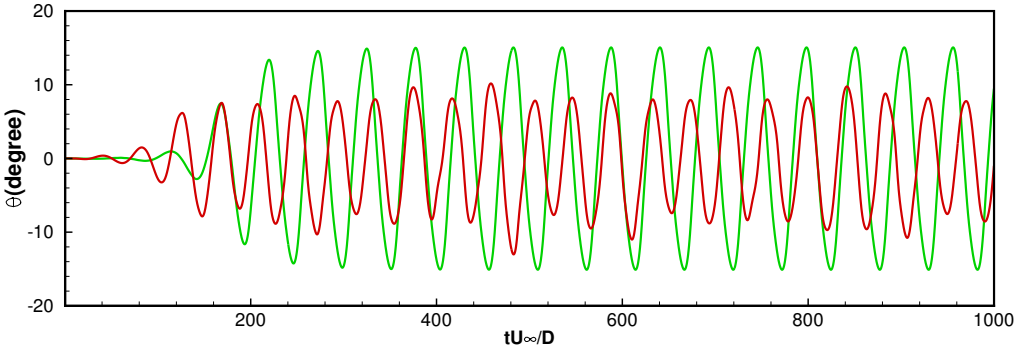



FIGURE 5.24: The rotational galloping responds of a square cross-section body $\Lambda = 4$ in a free-stream at $Re = 250$, $U_\theta^* = 40$, $\xi_\theta = 0.25$ and $I_\theta^* = 400$. — denotes a single body in a free stream and — denotes two bodies in a tandem arrangement.

Figure 5.25 shows a few snapshots of vorticity contour around two bodies. Figure 5.25 (a) (b) (c) give the image for the upstream body moving from the equilibrium position to the positive translational peak, while Figure 5.25 (d) (e) (f) represent the transformation from the balanced position to the negative peak position.

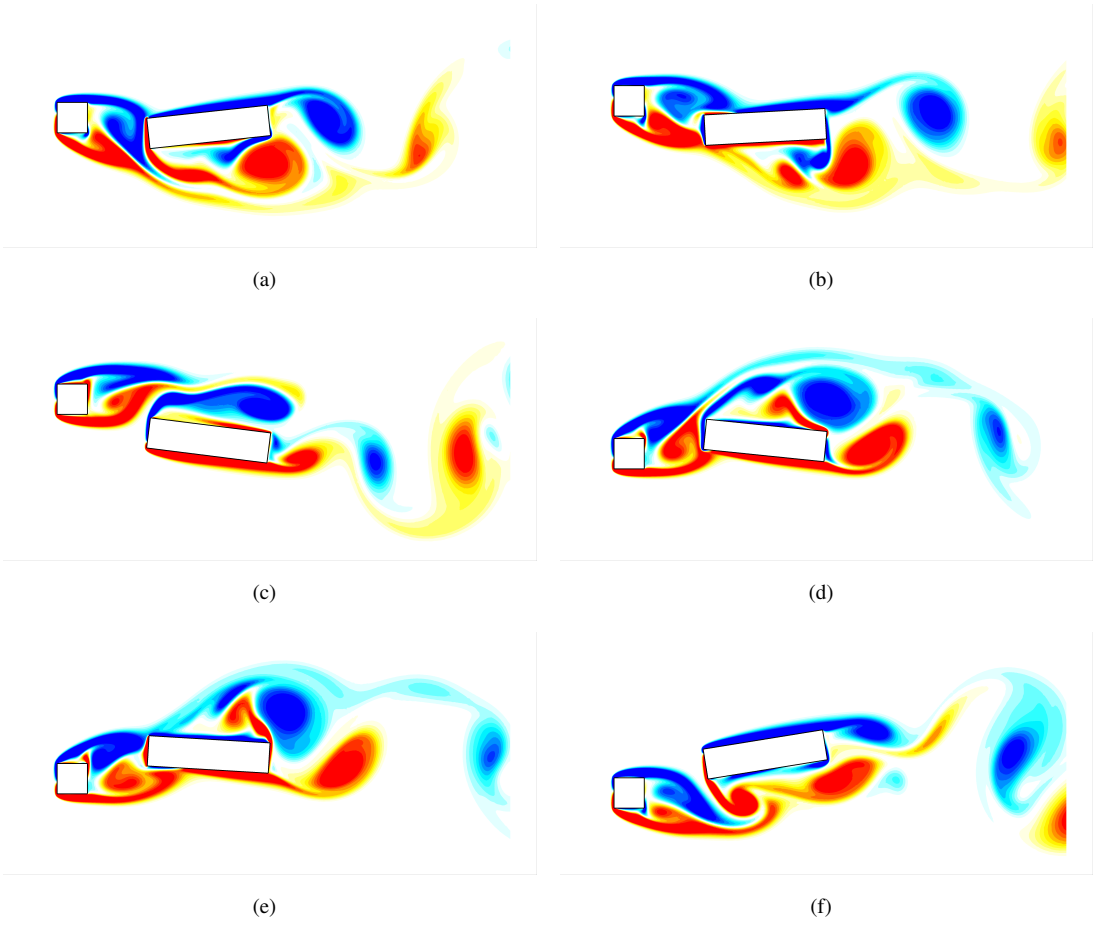



FIGURE 5.25: Vorticity snapshots of two galloping bodies in a free stream at $Re = 250$. (a) $y/D = 0.09, \theta = 6.87$; (b) $y/D = 0.69, \theta = 4.22$; (c) $y/D = 1.37, \theta = -6.20$; (d) $y/D = -0.01, \theta = -7.43$; (e) $y/D = -0.63, \theta = -4.73$; (f) $y/D = -1.36, \theta = 7.92$.

### 5.5.6 Vortex Induced Vibration (VIV) of a sphere

The VIV of a sphere is calculated in this part for 3D validation. It is observed that the flow regime is dominated by a highly organized periodic vortex shedding when the Reynolds number is larger than 270. Thus, we set the Reynolds number $Re = 300$ in this section, with which many published works are available for comparison.

#### 5.5.6.1 A stationary sphere

A stationary sphere with a diameter of $D = 1$ in a free-stream $U_\infty = 1$ is firstly calculated to verify the performance of fluid solver. The computational domain is set as a cuboid with $[-8D, 32D] \times [-8D, 8D] \times [-8D, 8D]$. The domain is then partitioned into 5.85 million tetrahedral elements with smaller elements clustered around the sphere as shown in Figure 5.26. The edge size of triangular elements on the sphere surface is about 0.02D. Inlet and outlet boundary conditions are set for left and right patches respectively, and slip condition is prescribed for other surrounding walls.

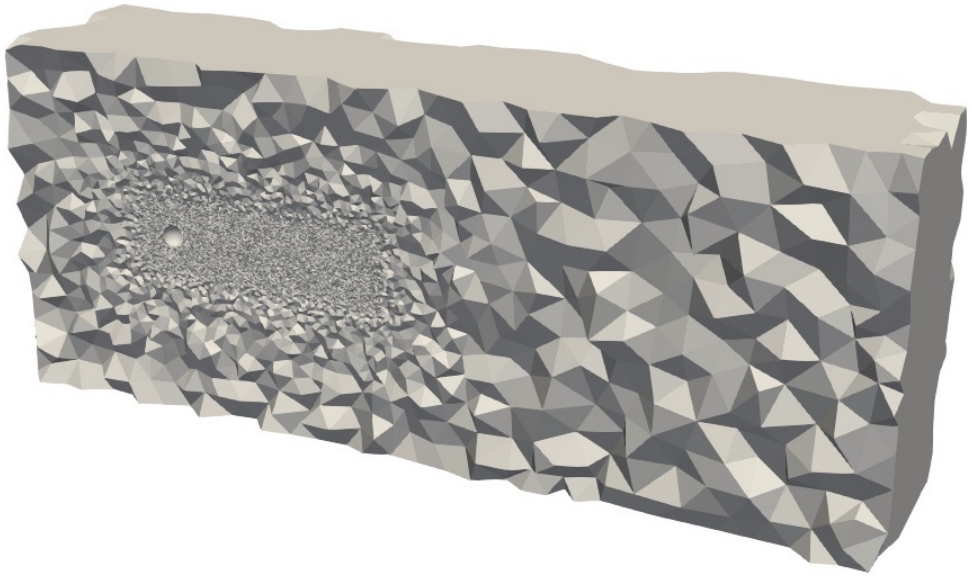


FIGURE 5.26: Computational mesh for a sphere in a free stream (5.85 million tetrahedral elements).

We performed the simulation up to time $t = 110s$. The computational results are quantitatively analyzed in terms of the drag coefficient $C_d$, lateral coefficient $C_l$, and side coefficient $C_s$, which are defined as the followings.

$$C_d = \frac{f_x}{\frac{1}{8}\pi\rho U_\infty^2 D^2}, C_l = \frac{f_l}{\frac{1}{8}\pi\rho U_\infty^2 D^2} \text{ and } C_s = \frac{f_s}{\frac{1}{8}\pi\rho U_\infty^2 D^2}, \tag{5.43}$$

where $f_x$ is the force exerted on the sphere in the streamwise direction, $f_s$ the force normal to the mean flow symmetry plane, and $f_l$ the force perpendicular to the plane spanned by $f_x$ and $f_s$. We plot the time history of these three coefficients in Figure 5.27. Then, the time-average drag coefficient $\overline{C}_d$ were calculated by $\overline{C}_d = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} Cd(t)dt$ with $[t_1 : t_2]$ covering several vortex periods for reducing the statistical error. Besides,

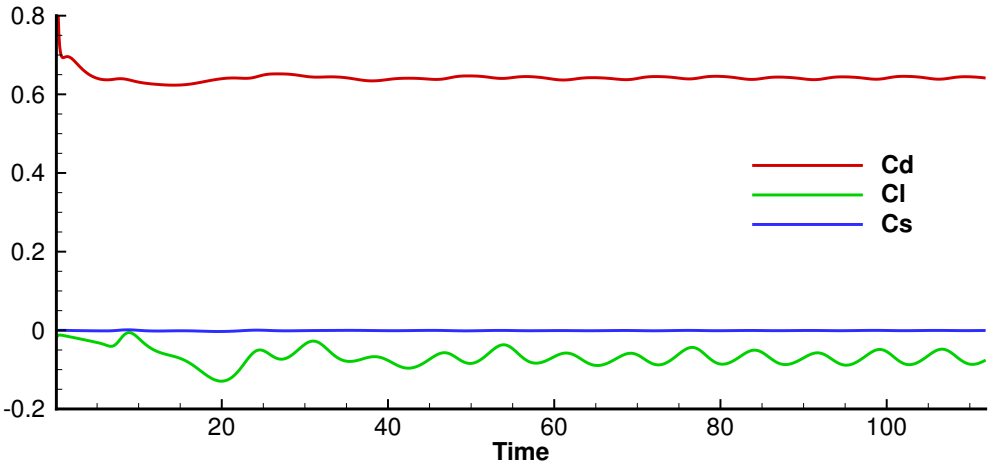



FIGURE 5.27: Time history of three force coefficients: $C_d$, $C_l$ and $C_s$ for a stationary sphere in a free stream at $Re = 300$.

TABLE 5.4: Time-averaged drag coefficient and the Strouhal number for 3D stationary sphere $Re = 300$.

| | Time-averaged $\overline{C}_d$ | Strouhal number St |
|---|---|---|
| Present | 0.642 | 0.133 |
| Behara et al. [126] | 0.600 | 0.132 |
| Tomboulides [127] | 0.671 | 0.136 |
| Johnson & Patel [128] | 0.656 | 0.137 |
| Roos & Willmarth [129] | 0.626 | - |

we computed the Strouhal number as $St = f_s \frac{d}{U_\infty}$ where $f_s$ is the vortex frequency. We compare $C_d$ and $St$ of our numerical results with other existing works in Table 5.4.

Furthermore, we give a snapshot of the wave structure around the sphere visualized with Q-criterion [6] in Figure 5.28. A chain of hairpin vortices is clearly observed in the downstream of the sphere, which is the typical structure of a stationary sphere in the freestream of Re=300. Both qualitative and quantitative analyses verify the capability of the present fluid solver to accurately simulate the 3D flow structures around stationary objects. The fluid solver were then used for the computations of fluid-sphere interactions shown next.

#### 5.5.6.2 Vortex Induced Vibration (VIV) of an elastically mounted sphere

In this section, we present the numerical results of an elastically mounted sphere $D = 1$ with three degrees of freedom immersed in a free stream of Re=300. The sphere is supposed to move translationally in three dimensions without rotational motions. Thus, the motion equation of the sphere is same as Eq (5.40). All non-dimensional parameters

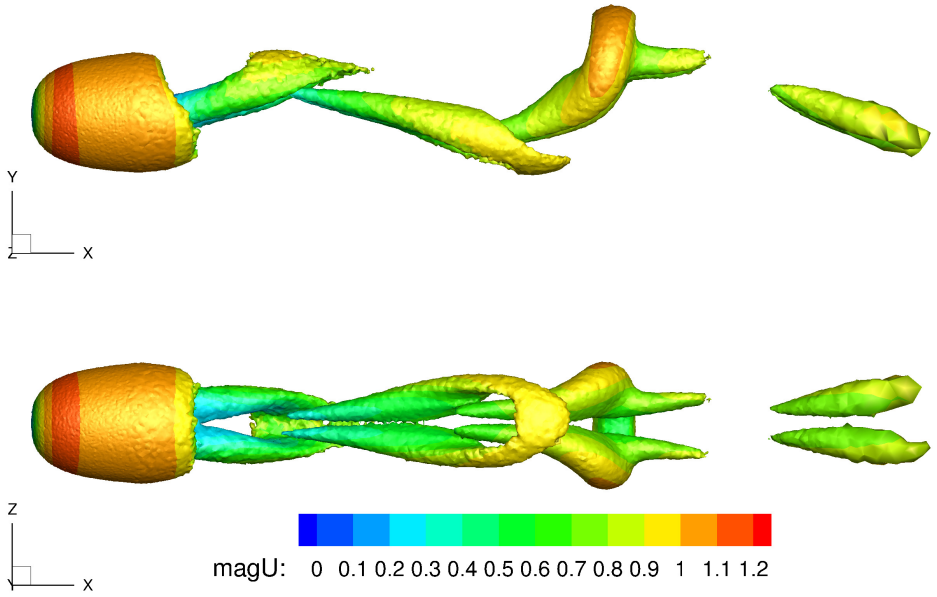
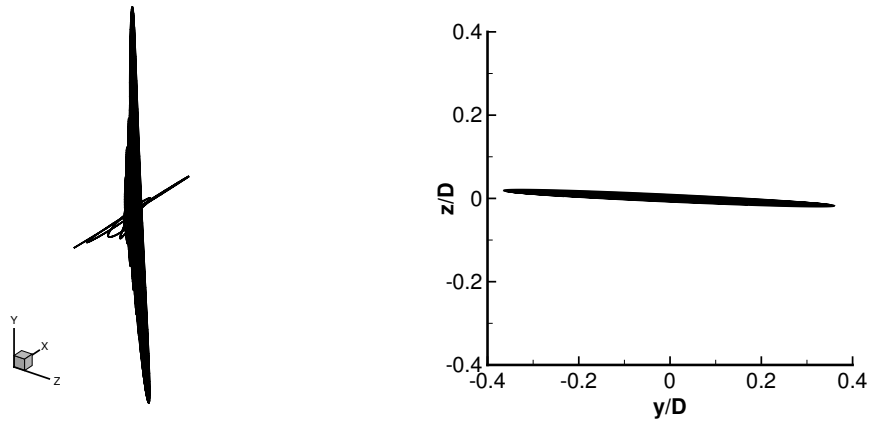



FIGURE 5.28: Wave structures visualized with Q-criterion [6] for a stationary sphere in a free stream at $Re = 300$. The contour surface ($Q = 0.02$) is colored by the magnitude of fluid velocity. The top figure is viewed in $x-y$ plane and the bottom one is viewed in $x-z$ plane.

used in 2D case are applied to 3D, except the mass ratio which is formulated as $m^* = \frac{m_s}{\rho D^3}$. The parameters are set as $m^* = 2, \zeta_s = 0$ and $U^* = 7$ for comparison with [126]. The weak coupling scheme is used for fluid/solid coupling since the mass ratio is not crucial.

The simulation was carried out from an initially stationary sphere in a free stream of Re=300. Figure 5.29 plots the trajectories of the sphere, where Figure 5.29 (a) and (b) are viewed in 3D space and the y-z plane, respectively. Figure 5.30 gives the time history of the three force coefficients ($C_d$, $C_l$, and $C_s$) for the sphere.

At the initial stage of computation, the sphere is significantly affected by the drag force while the lift force and side force are less dominant, which causes the motion of the sphere in the streamwise direction immediately. Since the sphere is mounted on the spring, the force exerted from spring always drives it back to the equilibrium position. Thus the sphere has a long trajectory in the x-direction as shown in Figure 5.29 (a). The oscillation amplitude of drag force gradually decreases, meanwhile the vortex around the sphere becomes unstable. Finally, the periodical vortices formed in the downstream of the sphere. The drag coefficient and lift coefficient vary periodically in time. The peak-to-peak amplitude of drag coefficient $C_d$ and $C_l$ are 0.113 and 0.326 respectively. The larger fluctuation in the lift force drives a more significant motion of the sphere in




(a) (b)

FIGURE 5.29: Trajectories of the mass center viewed in 3D space (a), and y-z plane (b) for an elastically mounted sphere in a free stream of $Re = 300$, $m^* = 2, \zeta_s = 0$, and $U^* = 7$.

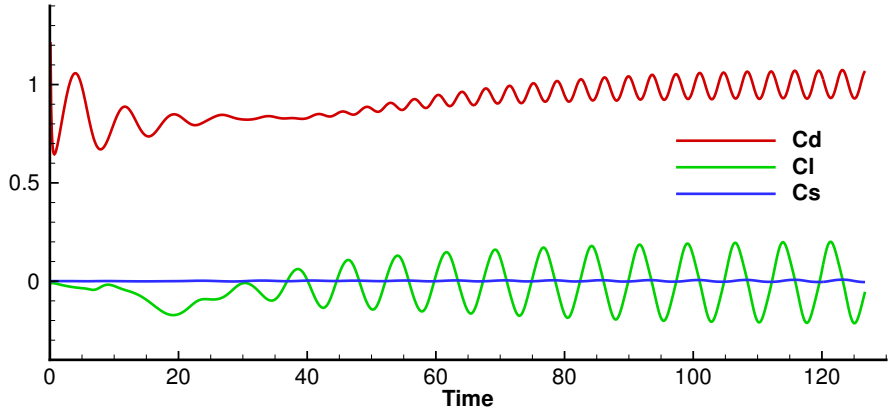



FIGURE 5.30: Time history of three force coefficients: $C_d$, $C_l$ and $C_s$ for a moving sphere in a free stream at Re=300, $m^* = 2, \zeta_s = 0$ and $U^* = 7$.

the transverse plane (x-z plane) compared to the streamwise direction. It is observed that the sphere oscillates along a line path in the transverse plane after the periodic vortex shedding has developed as shown in Figure 5.29 (b), which agrees well with Behara's observation [126]. The length of line path in the transverse plane in our simulation is 0.72D and the length of Behara's result is around 0.70D. A time sequence of the wave structures generated by the sphere is shown in Figure 5.31, where the Q-criterion iso-surfaces are snapshotted at every other second from top to bottom. We can see that the hairpin vortex structure is well resolved under the present mesh resolution.

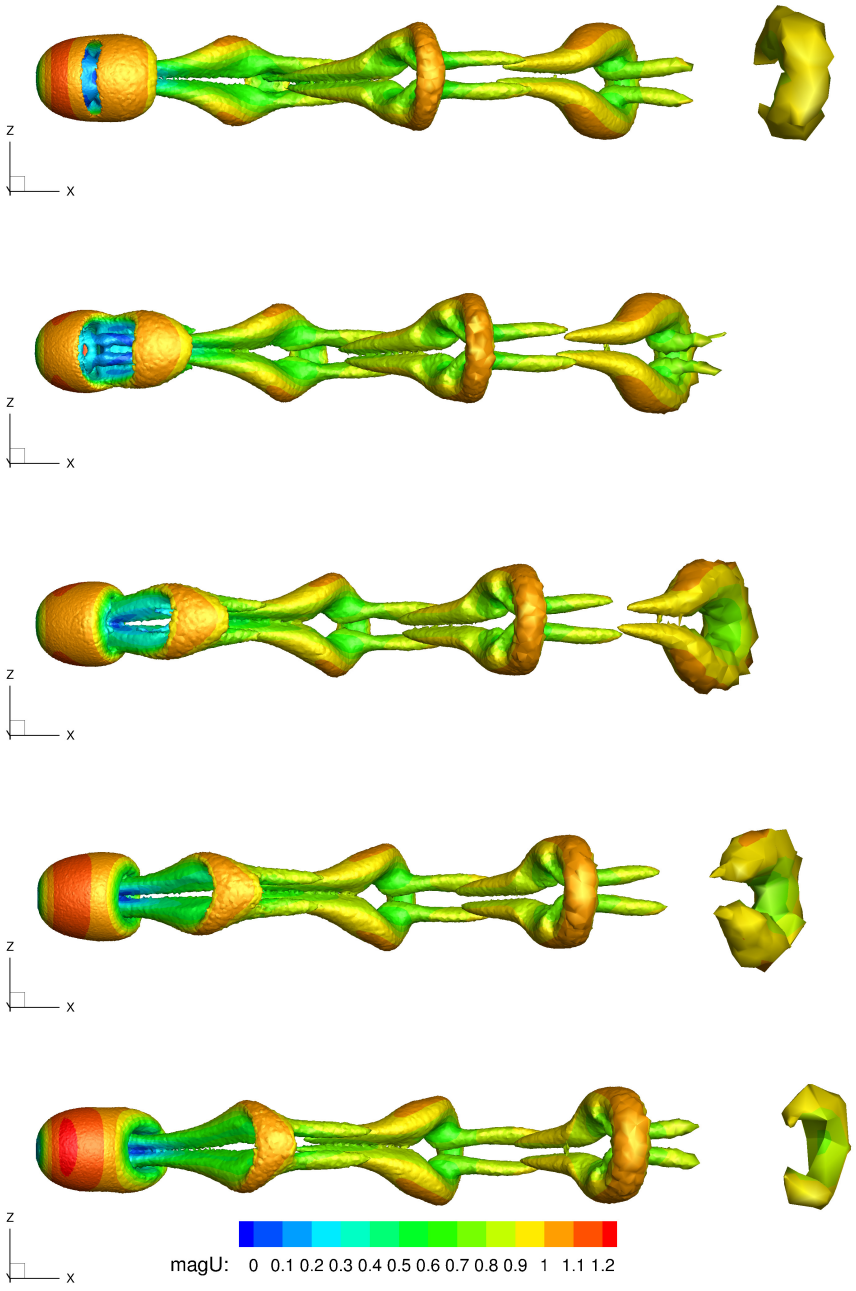



FIGURE 5.31: A time sequence of wave structures visualized with Q-criterion [6] for an elastically mounted sphere in a free stream at $Re = 300$, $m^* = 2, \zeta_s = 0$ and $U^* = 7$. The contour surface ($Q = 0.02$) is flooded by the magnitude of fluid velocity. The snapshots are captured every other second from top to bottom.

## 5.6 Short summary

In this chapter, numerical model for incompressible flows on moving grids is developed based on multi-moment finite volume method and applied to fluid-rigid body interaction problems. In this multi-moment finite volume method, both VIA and PV are treated as the computational variables and used for high order polynomial reconstruction on unstructured grids. The spatial discretization is based on the multi-moment reconstruction of quadratic polynomials for 2D triangular or 3D tetrahedral element. Numerical comparisons of the present scheme and a conventional 3rd order FVM for advection test demonstrate the superiority of the present multi-moment finite volume method as an accurate and efficient spatial discretization approach.

Incompressible Navier-Stokes equations are solved by a fractional step procedure, where the momentum equation is cast into an integral and a differential formulations for updating VIA and PV of fluid velocity respectively in an ALE framework. Convection and diffusion equations are firstly computed to put forward the intermediate fluid velocity. Velocity-pressure coupling equation is then solved through a semi-implicit step to obtain the pressure field at next time level, which is then used to correct both VIA and PV of fluid velocity to satisfy the continuity equation.

It is noted that the multi-moment finite volume method possesses at least two advantages, when applied to the ALE framework for fluid/solid interactions on unstructured grids. Firstly, with more locally increased degrees of freedom, quadratic or higher-order polynomial reconstruction can be built on compact stencil, which is not a trivial task for conventional FVM on unstructured grids, and secondly, the fluid-solid interface can be treated in a more accurate and convenient manner since the PVs are always located on the material interface so that the extra numerical step to retrieve the physical values is not needed.

In this study, the rigid body is moved by either a prescribed motion or an interactive motion with the surrounding fluid where the body is supposed to be mounted on elastic supports. In the latter case, the mass-spring-damping system of the solid motion is cast into second order ordinary differential equations (ODEs). We have developed two coupling schemes for fluid-rigid body interaction in this work, i.e. the explicit weak coupling scheme and the semi-implicit strong coupling scheme. Both schemes have been verified with benchmark tests that have different mass ratios between fluid and solid. In the cases where the mass of the solid body is light, the semi-implicit scheme has to be used to realize the equilibrium between fluid and solid at each time step through an iteration computation, so as to stabilize the FSI computation.

We have extensively verified the present numerical model with various benchmark tests in 2 and 3 dimensions in comparison with other methods reported in the literature. One typical benchmark test is VIV of a circular cylinder, where the case with mass ratio $m^* = 1$ is calculated by weak coupling scheme for efficiency, while semi-implicit coupling scheme is employed for the case with mass ratio $m^* = 0.5$ to ensure the numerical stability. The iterations rapidly reduce to only $2 \sim 3$ times in each time step after the initial stage of computation, by setting convergence tolerance between $1e^{-5} \sim 1e^{-7}$, which verifies the efficiency of semi-implicit coupling. In 3D case, VIV of an elastically mounted sphere is computed with Re=300. When periodic vortex formed, the sphere moves almost in a straight-line in transverse plane. The length of line path of our result is $0.72D$, which is very close to Behara's result $0.70D$. The numerical results substantiate the accuracy and robustness of our numerical model.

# Chapter 6

# Moving body in free surface flow

## 6.1 Brief introduction

Two-phase flows interacting with moving bodies occur in many engineering applications, such as naval hydrodynamics, process engineering like ink jet printers, etc.. In this phenomenon, three kinds of materials (usually gas, liquid and solid) interact with time where not only the moving boundary of fluid and structure, but also the interface of different fluids are of great interest. The numerical simulation is rather challenging due to the complexity of the geometrical shape and multi-physics interaction. For two-phase flows, the movement and deformation of the free interface is very flexible, and it may be associated with breakup and coalescence. Though there are some attempts to track the free surface by using moving grids, it is limited to simple geometries and simple physical problems. And undoubtedly it can hardly handle the phenomena with breakup and coalescence. Thus, the gas and liquid is formulated into one-fluid model, and the free interface is supposed to captured in this study. Then the remaining problem will be the interaction of fluid and structure which can be directly solved by the proposed model as shown in Chapter 5. In this chapter, a preliminary numerical model for solid-liquid-gas interaction is constructed as shown in Figure 6.1, where free surface flow is interacted with a moving body.

## 6.2 One fluid model for two-phase flow

The basic well-known governing equations for single phase flow, so-called Navier-Stokes equation, can be applied to describe two-phase flow. A simple and most popular

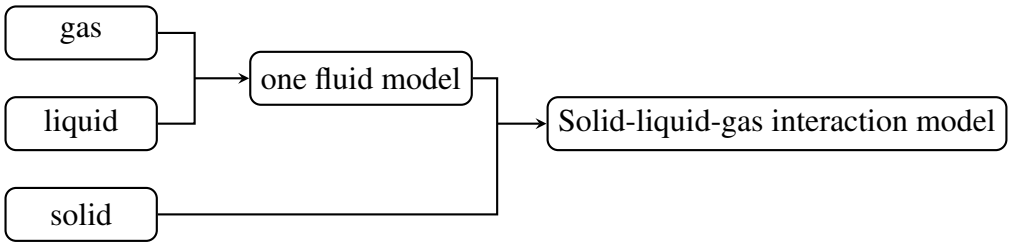



FIGURE 6.1: Illustration of building solid-liquid-gas interaction model.

numerical model for simulating two-phase flow is one fluid model (or homogeneous fluid model), which is adopted in this study.

To introduce one fluid model, we first consider two kinds of fluids filling in computational domain $\Omega$, where $\Omega_1$ denotes the region of fluid 1 and $\Omega_2$ for fluid 2. As shown in Figure 6.2, fluid 1 and fluid 2 are separated by interface $\partial\Omega_I$. In order to distinguish

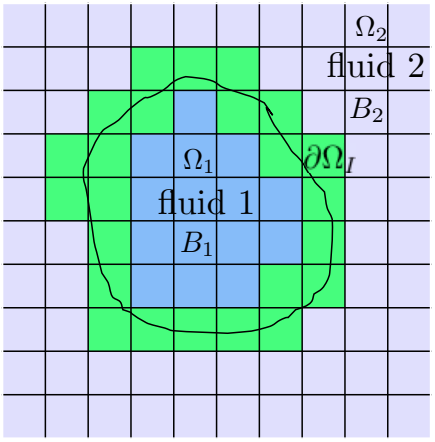



FIGURE 6.2: Computational domian with two kinds of fluids.

two fluids, it is nature to define a field function $H(\boldsymbol{x}, t)$ to indicate a given point $\boldsymbol{x}$ falls in fluid 1 or fluid 2 by

$$H(\boldsymbol{x}, t) = \begin{cases} 1 & \boldsymbol{x} \in \Omega_1 \\ 0 & \boldsymbol{x} \in \Omega_2. \end{cases} \tag{6.1}$$

Provided that the computational domain $\Omega$ is divided into non-overlapped elements $\Omega_i$, $(i = 1, 2, \cdots, \mathbb{I})$, the volume fraction in cell $\Omega_i$ is then defined by

$$\phi_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i(\boldsymbol{x})} H(\boldsymbol{x}, t) d\boldsymbol{x}, \tag{6.2}$$

where $|\Omega_i|$ is the volume of cell $i$. Thus, volume fraction $\phi_i$ is equal to 1 for the cell containing only fluid 1 (blue region in Figure 6.2), 0 for fluid 2 (grey region), and is between 0 and 1 for interface cells (green region).

Instead of using two sets of governing equations to describe fluid 1 and fluid 2 respectively, one fluid model uses only one set of averaged physical quantities, and separate different fluids with explicitly solved interfaces. We denote the mixed or averaged physical quantities in cell $\Omega_i$ as $B_i$, which are expressed by

$$B_i = \phi_i B_1 + (1 - \phi_i) B_2. \tag{6.3}$$

Here, physical quantities $B$ represent density $\rho$, pressure $p$, velocity $\boldsymbol{u}$ and so on, $B_1$ and $B_2$ are quantities of fluid 1 and fluid 2 respectively. In this way, single-phase governing equation can be extended to describe two-phase flow with a set of averaged physical quantities by just introducing surface tension force at interface. Thus our proposed numerical model can be directly adopted for two-phase flow based on one-fluid model.

## 6.3 Governing equations for two-phase flow

The indicator function $H(\boldsymbol{x}, t)$ follows the Lagrangian invariant relation of advection,

$$\frac{d}{dt}(H(\boldsymbol{x}, t)) = 0, \tag{6.4}$$

where $\frac{d}{dt}$ denotes the material derivative in respective to time. Given the velocity of the fluid particle $\boldsymbol{u}$, the time evolution of the indicator of Eulerian representation can be cast in the following advection equation,

$$\frac{\partial H}{\partial t} + \boldsymbol{u}\nabla \cdot H = 0. \tag{6.5}$$

By integrating over a spatial control volume, it also can be represented as

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\phi) = \phi \nabla \cdot \boldsymbol{u}. \tag{6.6}$$

In one-fluid model, *Navier-Stokes* equations are used to describe the incompressible two-phase flows in Eulerian coordinates as

$$\nabla \cdot \boldsymbol{u} = 0, \tag{6.7}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \vec{\tau} + \rho \mathbf{g} + \rho \mathbf{F}_s. \tag{6.8}$$

In these equations, $\rho$ is the density of the fluid, $p$ the pressure, $\boldsymbol{u}$ the velocity of the fluid, $\vec{\tau}$ the shear stress tensor, $\mathbf{g}$ the gravitational accelerate, and $\mathbf{F}_s$ surface tension force acts on two-phase interface. For *Newtonian fluids*,the shear stress can be obtained from the constitutive equation below,

$$\tau_{ij} = \begin{cases} \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), & i \neq j \\ \mu\left(2\frac{\partial u_i}{\partial x_i} - \frac{2}{3}\nabla \cdot \boldsymbol{u}\right), & i = j \end{cases} \tag{6.9}$$

where, $\mu$ is dynamic viscosity of the fluid. The kinematic viscosity is denoted as $\nu = \frac{\mu}{\rho}$. The shear stress $\vec{\tau}$ can also be written into $\nabla \cdot \vec{\tau} = \nabla \cdot (\mu \nabla \boldsymbol{u}) + \nabla \boldsymbol{u} \cdot \nabla \mu$.

Surface tension forces $\mathbf{F}_s$ are singular forces active on the phase interface, which is an import interface force that should be seriously treated in the simulation of two-phase flow, the expression of surface tension force shows as

$$\mathbf{F}_s = \int_\Gamma \sigma\kappa\delta(\boldsymbol{x} - \boldsymbol{x}_s)\mathbf{n}d\Gamma(\boldsymbol{x}_s), \tag{6.10}$$

where, $\Gamma$ is the two fluid interface, $\delta(\boldsymbol{x} - \boldsymbol{x}_s)$ is the three dimensional (3D) Dirac delta function, $\boldsymbol{x}$ represents the point coordinate in all fluid region, and $\boldsymbol{x}_s$ only the point coordinate in interface, $\mathbf{n}$ is the normal vector in the interface.

The Continuum Surface Force (CSF) model, proposed by Brackbill [130], is employed in this research to approximate the singular force. In the CSF model, the interface is represented as a region of finite thickness, where density and viscosity are regularized as smooth functions and the surface tension acting on the interface is transformed into a localize volume force. We perform a volume integral over $\Omega_i$ for equation (6.10), This gives for the surface tension term

$$\int_{\Omega_i} \mathbf{F}_s d\Omega = \int_{\Gamma\cap\Omega_i} \sigma\kappa\mathbf{n}d\Gamma(\boldsymbol{x}_s) = \int_{\Omega_i} \sigma\kappa\nabla\phi d\Omega$$

where, $\Gamma \cap \Omega_i$ is the section of the interface that lies in $\Omega_i$. Furthermore, the local interfacial curvature $\kappa$ is obtained from the volume fraction using

$$\kappa = -\nabla \cdot \mathbf{n} \tag{6.11}$$

with

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{6.12}$$

In order to devise a balanced model for gravity force, a modified pressure is used as $p_r = p - \rho \mathbf{g} \cdot \boldsymbol{x}$, where $p_r$ represents the pressure obtained by removing the hydrostatic pressure from the total pressure, which results in

$$\nabla p = \nabla p_r + \nabla \left( \rho \mathbf{g} \cdot \boldsymbol{x} \right) = \nabla p_r + \rho \mathbf{g} + \mathbf{g} \cdot \boldsymbol{x} \nabla \rho. \tag{6.13}$$

Thus, the final form of momentum equation can be formulated as

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = (\nabla \cdot (\mu \nabla \boldsymbol{u}) + \nabla \boldsymbol{u} \cdot \nabla \mu) - \nabla p_r - \mathbf{g} \cdot \boldsymbol{x} \nabla \rho + \sigma \kappa \nabla \phi. \tag{6.14}$$

## 6.4 Numerical solution procedure

By treating body movement as boundary conditions of fluid domain, the problem can be solved by one-fluid model on moving mesh. We summarize the numerical procedure for solving two-phase flow as follows:

1. Advection equation for volume fraction Eq.(6.6) is specially solved by an algebraic interface capturing scheme, THINC-QQ scheme, which will be introduced in section 6.5;

2. In one fluid model, the dynamic effects of different fluids are realized by updating the material quantities, like density and viscosity, based on updated volume fraction $\phi$ as:

$$\rho = \rho_1 \phi + \rho_2 (1 - \phi) \quad \text{and} \quad \mu = \mu_1 \phi + (1 - \phi) \mu_2 \tag{6.15}$$

   where $\rho_1$ and $\rho_2$ are the densities, $\mu_1$ and $\mu_2$ the viscosity coefficients of fluid 1 and fluid 2.

3. Navier-Stokes equations for variable density incompressible flows are numerically solved by projection method, which are summarized as follows for fluid variables from time $t^n$ to $t^{n+1}$:

   (a) Given the velocity $\boldsymbol{u}^n$ at step $n$, compute the convection and diffusion terms of momentum equation (6.14) to put forward the intermediate velocity $\boldsymbol{u}^*$

by ALE integral and differential formulations as,

$$\begin{aligned}\frac{\partial}{\partial t}\int_{\Omega_{\boldsymbol{\chi}}(t)}\rho\boldsymbol{u}d\Omega &= -\int_{\Gamma(t)}(\boldsymbol{u}-\boldsymbol{u}_g)\otimes\rho^n\boldsymbol{u}^n\cdot\mathbf{n}d\Gamma+\int_{\Gamma(t)}\mu^n\nabla\boldsymbol{u}^n\cdot\mathbf{n}d\Gamma\\ &\quad+\nabla\boldsymbol{u}^n\int_{\Gamma(t)}\mu^n\cdot\mathbf{n}d\Gamma\\ \frac{\partial\boldsymbol{u}}{\partial t}\Big|_{\boldsymbol{\chi}} &= -(\boldsymbol{u}-\boldsymbol{u}_g)\cdot\nabla\boldsymbol{u}^n+\nu^n\nabla^2\boldsymbol{u}^n+\nabla\boldsymbol{u}^n\cdot\nabla\nu^n.\end{aligned} \tag{6.16}$$

(b) Update the intermediate velocity from $\boldsymbol{u}^*$ to $\boldsymbol{u}^{**}$ by adding the effect of surface tension and gravity force,

$$\frac{\boldsymbol{u}^{**}-\boldsymbol{u}^*}{\Delta t}=\frac{1}{\rho}\left(\sigma\kappa\nabla\phi-\boldsymbol{g}\cdot\boldsymbol{x}\nabla\rho\right) \tag{6.17}$$

(c) Generally, the intermediate velocity $\boldsymbol{u}^{**}$ does not satisfy the continuity equation. Thus, we calculate the pressure by solving Poisson equation to enforce the divergence-free constraint condition

$$\nabla\cdot\left(\frac{1}{\rho}\nabla p_r^{n+1}\right) \;=\; \frac{1}{\Delta t}\nabla\cdot\boldsymbol{u}^{**}. \tag{6.18}$$

(d) Correct the velocity by the correction step equivalently by

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{**}}{\Delta t}=-\frac{1}{\rho}\nabla p_r^{n+1}. \tag{6.19}$$

It is obvious that by this procedure the updated velocity in the next time step satisfies the continuity property.

As we can see from the procedure, step 2 is very straightforward and step 3 is similar as section 5.2.2, thus does not need further explanation. The problem remained here is to solve advection equation of volume fraction, which will be presented in next section.

## 6.5 THINC/QQ scheme for interface capturing

### 6.5.1 Semi-discrete finite volume formulation for advection equation

To achieve mass conservativeness, the advection equation Eq.(6.5) is formulated as

$$\frac{\partial H}{\partial t}+\nabla\cdot(\boldsymbol{u}H)=H\nabla\cdot\boldsymbol{u}. \tag{6.20}$$

Thus, the integration over a moving control volume casts into

$$\frac{\partial}{\partial t}\int_{\Omega_{\boldsymbol{\chi}}(t)} H d\Omega + \int_{\Gamma(t)} (\boldsymbol{u}-\boldsymbol{u}_g) H \cdot \boldsymbol{n} d\Gamma = \int_{\Omega(t)} H \nabla \cdot \boldsymbol{u} d\Omega, \tag{6.21}$$

where $\boldsymbol{u}_g$ is the grid velocity. Finally, we can obtain the semi-discrete finite volume formulation for advection equation on arbitrarily moving domain as

$$\frac{\partial \overline{\phi} V_i}{\partial t} = -\sum_{j=1}^{\mathbb{J}} \left(\overline{\boldsymbol{u}}_{\Gamma_{ij}} - \overline{\boldsymbol{u}}_{\Gamma_{gij}}\right) \cdot \boldsymbol{n}_{ij} |\Gamma_{ij}| \sum_{q=1}^{Q} \omega_q \tilde{H}_i(\boldsymbol{x}_q)_{iup} + \overline{\phi}_i \sum_{i=1}^{\mathbb{J}} \left(\boldsymbol{u}_{ij} \boldsymbol{n}_{ij} |\Gamma_{ij}|\right), \tag{6.22}$$

where $\tilde{H}_i(\boldsymbol{x}_q)_{iup}$ stands for the reconstruction value at Gaussian quadrature point $\boldsymbol{x}_q$ of upwinding cell. 4-point Gaussian quadrature is used for 2D edge line. For time integration, we use 3rd-order TVD-Runge-Kutta scheme.

### 6.5.2 Reconstruction of indicator function

When we get the distribution of volume fraction in the field, the key point is how to exactly decide the interface distribution in spatial space, which is so called interface reconstruction. Generally, there are two ways to do reconstruction, the one is geometrical reconstruction and the other is algebraic reconstruction. It is known that the conventional geometrical reconstruction like PLIC is one of the most accurate reconstruction methods, however, a 3D implementation is a troublesome task. The other one is algebraic VOF method, so-called tangent of hyperbola for interface capturing (THINC) [131] method, which uses a hyperbolic tangent function to retrieve the jump distribution in the volume fraction function. The numerical flux is directly computed from the THINC reconstruction function. Fully multi-dimensional THINC schemes can be devised in a straightforward way by using multi-dimensional hyperbolic tangent function. In this study, quadratic multi-dimensional THINC formulation, proposed by Xie [132], is adopted for capturing the interface.

To evaluate the numerical solutions in Eq.(6.22), the indicator function is approximated by a hyperbolic tangent function in 2D local coordinate system $\xi - \eta$ for the target element $\Omega_i$ as

$$\tilde{H}_i(\xi,\eta) = \frac{1}{2}\left(1 + \tanh\left(\beta\left(\mathcal{P}_i(\xi,\eta) + d_i\right)\right)\right), \tag{6.23}$$

where $\beta$ determines the steepness of the jump in the interpolation function and is set as $3.0$ in this study. $\mathcal{P}_i(\xi,\eta) + d_i$ can be explained as the signed distance to the interface [133], thus $\mathcal{P}_i(\xi,\eta) + d_i = 0$ implicitly describes the interface. Interested readers may

refer to [132] for 3D formulations. For quadratic representation of the interface, $\mathcal{P}_i(\xi,\eta)$ is approximated by a quadratic polynomial

$$\mathcal{P}_i(\xi,\eta) = a_{20}\xi^2 + a_{02}\eta^2 + a_{11}\xi\eta + a_{10}\xi + a_{01}\eta, \tag{6.24}$$

where coefficients $a_{st}(s,t=0,1,2,s+t<=2)$ are determined by the constraints based on the interfacial geometrical features, i.e. the unit normal vector and curvature tensors, as follows

$$\begin{cases} \frac{\partial \mathcal{P}}{\partial \xi}(\boldsymbol{\xi}_{ic}) = \varphi_\xi \\ \frac{\partial \mathcal{P}}{\partial \eta}(\boldsymbol{\xi}_{ic}) = \varphi_\eta \\ \frac{\partial^2 \mathcal{P}}{\partial \xi^2}(\boldsymbol{\xi}_{ic}) = \varphi_{\xi^2} \\ \frac{\partial^2 \mathcal{P}}{\partial \eta^2}(\boldsymbol{\xi}_{ic}) = \varphi_{\eta^2} \\ \frac{\partial^2 \mathcal{P}}{\partial \xi\eta}(\boldsymbol{\xi}_{ic}) = \varphi_{\xi\eta} \end{cases} \Longrightarrow \begin{cases} a_{10} = \varphi_\xi - \varphi_{\xi^2} - \eta_{ic}\varphi_{\xi\eta} \\ a_{01} = \varphi_\eta - \varphi_{\eta^2} - \xi_{ic}\varphi_{\xi\eta} \\ a_{11} = \varphi_{\xi\eta} \\ a_{20} = \frac{1}{2}\varphi_{\xi^2} \\ a_{02} = \frac{1}{2}\varphi_{\eta^2} \end{cases} \tag{6.25}$$

In these formulations, the unit normal vector $(\varphi_\xi, \varphi_\eta)$ and curvature tensors

$$\varphi_{ab} = \frac{1}{2}\left(\frac{\partial \varphi_a}{\partial b} + \frac{\partial \varphi_b}{\partial a}\right) \tag{6.26}$$

are approximated at cell center by following procedures:

1. Transform the first order derivatives at cell vertices from global coordinates $(\phi_{xik}, \phi_{yik})$ to local coordinates $(\phi_{\xi ik}, \phi_{\eta ik})$.

2. Compute the normalized derivatives at each vertex:

$$\begin{cases} \varphi_{\xi ik} = \phi_{\xi ik}/|\phi_{ik}| \\ \varphi_{\eta ik} = \phi_{\eta ik}/|\phi_{ik}| \end{cases} \quad \text{for} \quad k = 1, 2, \cdots, \mathbb{P} \tag{6.27}$$

3. Approximate the unit normal vector and curvature tensors at mass center by interpolation and differentiation from the piecewise polynomial $\varphi_\gamma(\boldsymbol{\xi}) = \sum_{k=1}^{\mathbb{P}} \varphi_{\gamma ik}\mathcal{N}_{ik}(\boldsymbol{\xi})$ with $\gamma$ denoting either $\xi$ or $\eta$, $\mathcal{N}_{ik}(\boldsymbol{\xi})$ being the basis functions as presented in Table 3.4.

Once the coefficients $a_{st}$ are determined, the only unknown $d_i$, which indicates the location of interface, can be determined by the volume fraction values by

$$\frac{1}{|\Omega_i|}\int_{\Omega_i(\boldsymbol{\xi})} \tilde{H}_i(\xi,\eta)d\xi\eta = \overline{\phi}_i. \tag{6.28}$$

Since there is no general analytical expression for multi-dimensional integral of the function $\tilde{H}(\boldsymbol{\xi})$, A fully multi-dimensional Gaussian quadrature is used to approximate the integral of the tangent hyperbolic function, which greatly simplifies the numerical procedures for arbitrary unstructured grids. The Gaussian quadrature is formulated as following

$$\sum_{q=1}^{Q} \omega_{iq} \left( \frac{1}{2} \left( 1 + \tanh \left( \beta \left( \mathcal{P}_i(\boldsymbol{\xi}_{iq}) + d_i \right) \right) \right) \right) = \overline{\phi}_i, \tag{6.29}$$

where $\boldsymbol{\xi}_{iq}$ and $\omega_{iq} (q = 1, 2, \cdots, Q)$ are the coordinates and weights of Gaussian points in element $\Omega_i$. It is noted that the weights satisfy $\sum_{q=1}^{Q} \omega_{iq} = 1$. Thus, the unknown $d_i$ can be computed by employing Newton-Raphson method which takes a few iterations to converge in practice.

## 6.6 Numerical tests

It is challenging to calculate the phenomena where a moving body is close to or intersects a free surface, since the interaction of three phases should be formulated and integrated correctly. Some numerical experiments are conducted in this section to verify the robustness and accuracy of the model.

### 6.6.1 Generation of a solitary wave by moving a wave paddle

Wave generation is an active research field in coastal engineering. According to the wave maker theory [134], a specific wave can be generated with an appropriate paddle movement. A typical example for generating a solitary wave [135, 136] is by setting the wave paddle velocity as

$$u[x(t), t] = \frac{c\eta}{h + \eta} \tag{6.30}$$

where $\eta$ is the free surface displacement that is defined as

$$\eta(x(t), t) = H sech^2 \left( \sqrt{\frac{3H}{4h^3}} \left( x(t) - ct + x_0 \right) \right). \tag{6.31}$$

In these formulations, $H$ is wave height, $h$ still water depth, $c = \sqrt{g(h + H)}$ the phase velocity of the wave train with $g$ being the gravitational acceleration, $x_0 = \frac{4h}{\sqrt{H/h}}$ the distance between the origin and the crest of the wave train at $t = 0$, $x(t) = \int_0^t u dt$ the

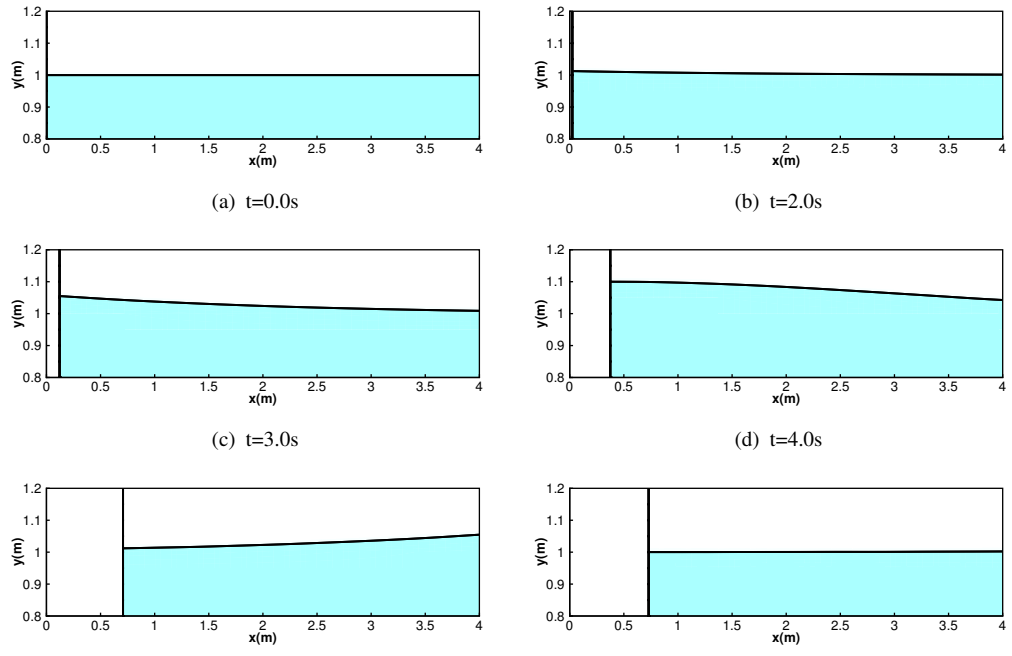
location of the paddle at time $t$. The parameters are set as $H = 0.1m$, $h = 1.0m$ and $g = 9.81m/s^2$.

The computational domain is set as $100m \times 1.4m$ with a wave paddle on the left. We partitioned the domain into a finite number of uniform quadrilateral elements. For grid independent study, coarse, medium and fine mesh are used with grid size ($\Delta x = \Delta y$) being $0.2m$, $0.1m$ and $0.05m$ respectively. A radiation boundary condition is prescribed at the right wall. In the simulation, the mesh movement is given as

$$\boldsymbol{u}_g(\boldsymbol{x}, t) = (\frac{100 - x}{100 - x_p} u_p, 0), \tag{6.32}$$

where $x_p$ and $u_p$ are the x-component of displacement and velocity of wave paddle respectively.

We show the free surface variation from $t = 0s$ to $t = 8s$ in Figure 6.3. The wave paddle accelerates from rest and moves towards right, which makes the free surface pushed up and transmit toward right. Afterwards, the wave paddle slows down and finally comes down to be silent. As a result, a solitary wave is sent out.




(a) t=0.0s (b) t=2.0s

(c) t=3.0s (d) t=4.0s

(e) t=6.0s (f) t=8.0s

FIGURE 6.3: Generation of a solitary wave train by the prescribed wave paddle motion

We plot the time history of free surface displacement at $x = 5m$ and $x = 20m$ for different grids and compare with the analytical solution in Figure 6.4. As the grid

refined, the numerical solution converges to the exact solution, which demonstrates the feasibility of present solver for wave generation.

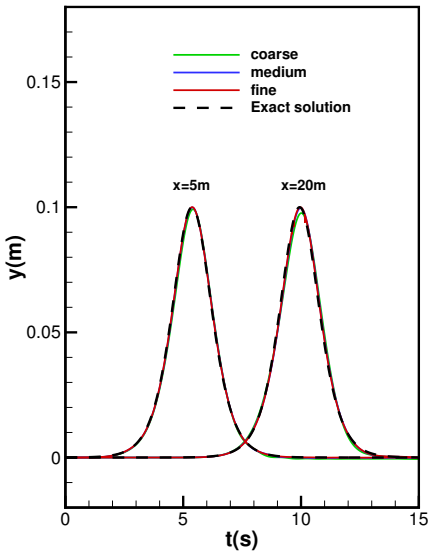



FIGURE 6.4: Free surface displacement at $x = 5m$ and $x = 20m$ for different meshes and the comparison with the exact solution

### 6.6.2 Water exit of a circular cylinder

We then calculate the case about the water exit of a circular cylinder in 2D by following the set up of [7, 136]. As shown in Figure 6.5, the computational is given as a rectangular with length $L = 40m$ and height $H = 40m$. The water depth is set as $h = 22.5m$. A horizontal circular cylinder with radius $r = 1m$ is initially put below the water surface at a distance of $d = 1.25m$. We partition the domain by structured quadrilateral elements for most area with size of $\Delta x = \Delta y = 0.25m$ and triangular elements around the cylinder whose perimeter is uniformly divided by 128 segments as shown in Figure 6.6. In the calculation, the cylinder is forced to move upward with a constant velocity $V = 0.39m/s$, and the gravity acceleration is fixed to be $g = 1.0m/s$. The set up gives the non-dimensional parameters $\epsilon = r/d = 0.8$ and $F_r = \frac{V}{\sqrt{gd}} = 0.39$.

We carry out the computation up to a normalized time $T = \frac{Vt}{d} = 3$ and a series of snapshots about the water surface profile are presented in Figure 6.7. The surface profiles agree well with some references [136, 137]. We also compare the obtained surface profile with a boundary element method as shown in Figure 6.8. The good agreement proves the feasibility of our numerical solver for simulating water exit problem.

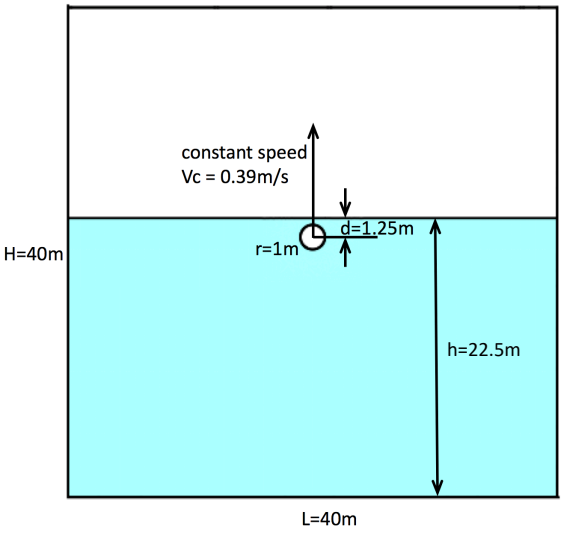



FIGURE 6.5: Computational domain of water exit of a circular cylinder

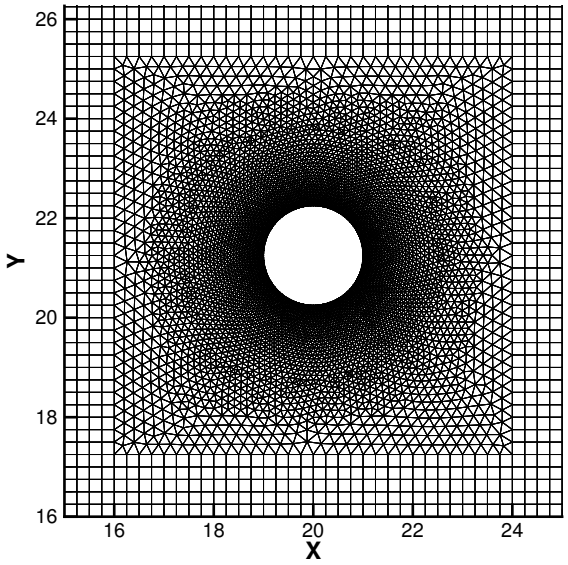
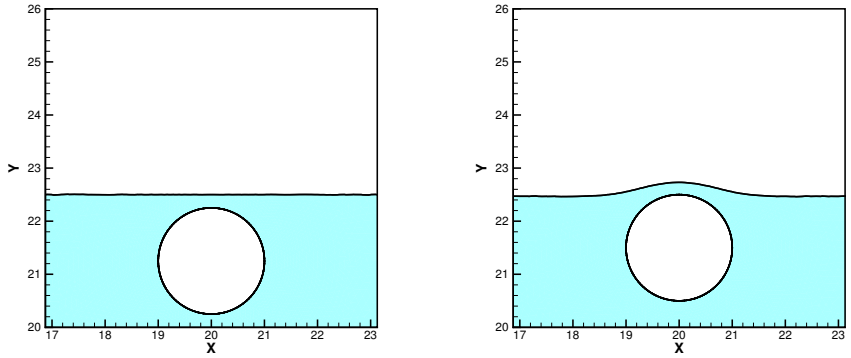


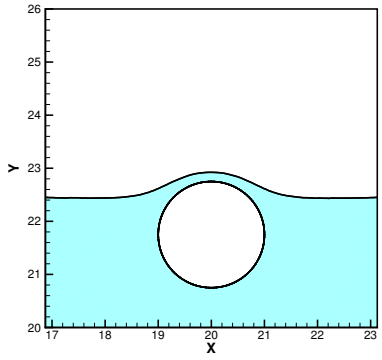

FIGURE 6.6: Computational mesh around the circular cylinder.



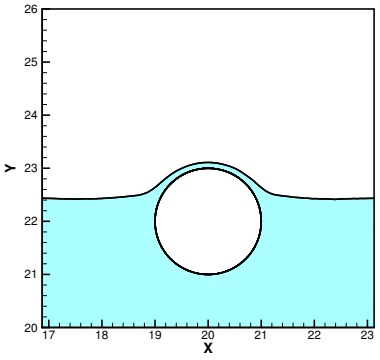

(a) T=0.0



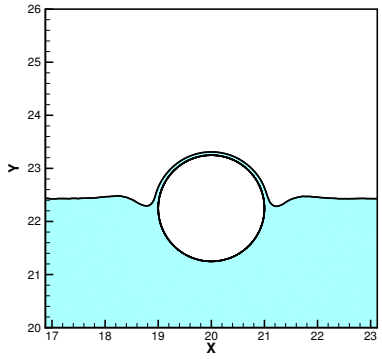

(b) T=0.2



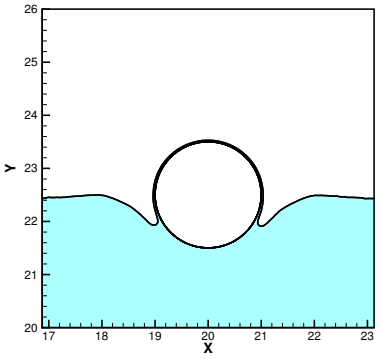

(c) T=0.4



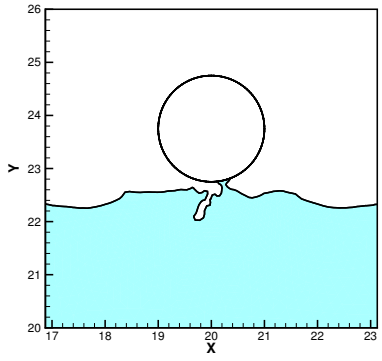

(d) T=0.6



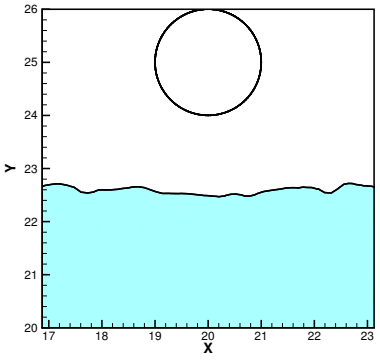

(e) T=0.8




(f) T=1.0

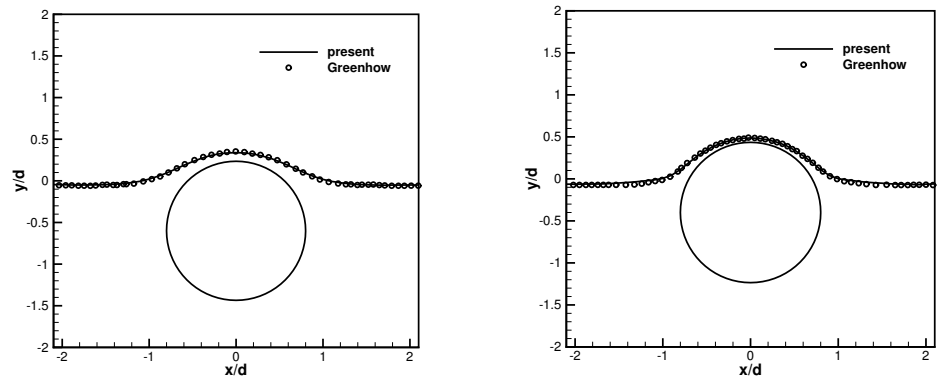



(g) T=2.0




(h) T=3.0

FIGURE 6.7: Free surface profile of horizontal cylinder




(a) T=0.4 (b) T=0.6

FIGURE 6.8: Comparisons of free surface profile with a simulation by boundary element method [7].

### 6.6.3 Free falling of a wedge

To verify the dynamic coupling between fluid and structure, we compute a free falling symmetric wedge impacting on the free surface by following the study of Zhao et al. [8]. The wedge is given with deadrise angle $30^o$ and the breadth of $0.5m$ as presented in Figure 6.9. In the experiment of Zhao et al. [8], it is three dimensional structure with the total length of $1m$ where the length of measure section is $0.2m$. The total weight of drop rig is $241kg$ and weight of the measure section is $14.5kg$. Five points ($P1, P2, P3, P4$ and $P5$ as shown in Figure 6.9) are located on the edge of the wedge to measure the pressure in the experiment. The wedge is given only degree of freedom in vertical motion and with the initial vertical velocity $V = -6.15m/s$.

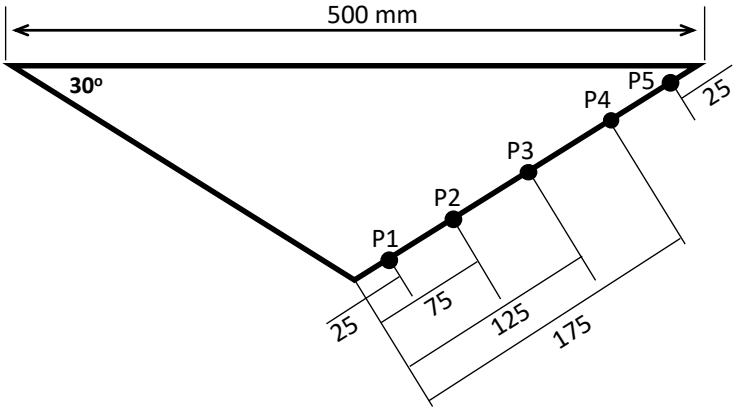
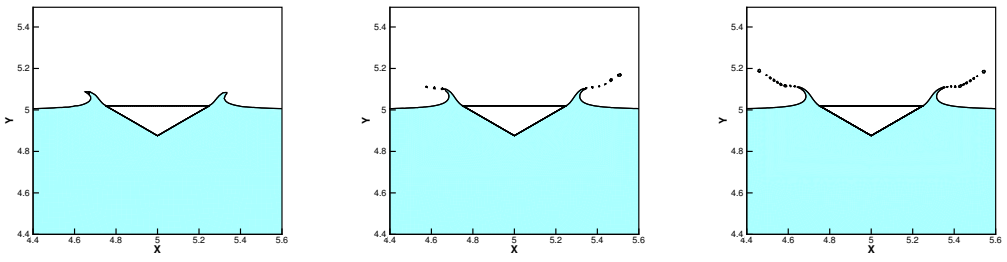



FIGURE 6.9: The geometry of the wedge.

The computational domain is set as $[0, 10]m \times [0, 10]m$ and partitioned by triangular and quadrilateral elements where the area near the wedge is refined. To study the grid dependency, three kinds of grids (coarse, medium and fine) are used where the sizes of cell edges along the wedge are $0.01m$, $0.005m$ and $0.0025m$ respectively. The free surface is initially located at the middle of the domain with $y = 5m$. The water has density of $1000kg/m^3$ and kinematic viscosity of $1.0 \times 10^{-6}m/s^2$. The density and kinematic viscosity of air are $1kg/m^3$ and $1.8 \times 10^{-5}m/s^2$. The gravity acceleration is set as $-9.8m/s^2$.

We present the free surface profiles at time $t = 0.0202s$ calculated on the above three grids as shown in Figure 6.10. Obviously, the water jets can be captured more clearly with finer mesh. We also plot the time history of falling velocity and slamming force



(a) coarse grid (b) medium grid (c) fine grid

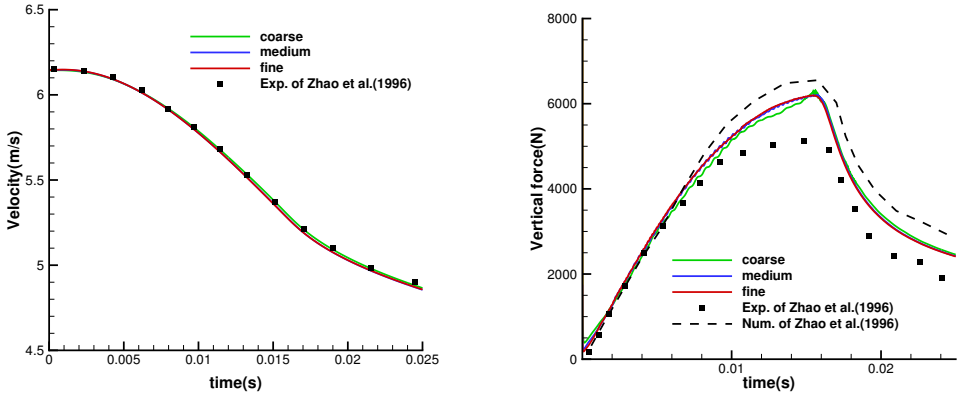
FIGURE 6.10: Free surface profiles at time t=0.0202s for different grids.

exerted on the wedge in Figure 6.11. Our numerical results converge with the refined mesh. And the good agreement is achieved with the solution of Zhao et al. [8].

We further present the pressure contours around the wedge at three time instants as shown in the left panels of Figure 6.12, and plot the pressure along the wedge boundary in the right panels of Figure 6.12. The pressure distributions from the present model are compared with the experimental and numerical results of Zhang et al. [8]. The close pressure profiles prove the feasibility of this solver for fluid-solid interaction.

## 6.7 Short summary

In this chapter, a preliminary numerical model is presented for free surface flow involving with a moving body. We use moving grids to track the boundary of fluid and solid, and capture the free surface of two phase flow on a modestly moving mesh. In one-fluid model, two kinds of fluids are described by Navier-Stokes equations with a set of

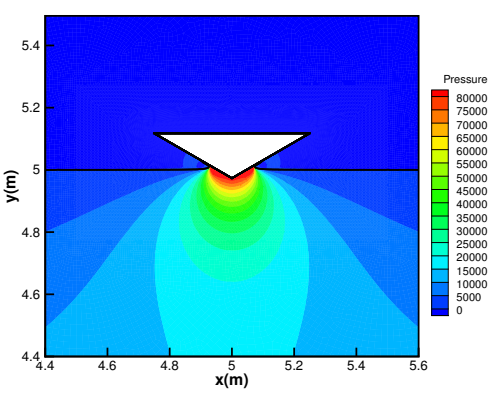



(a) falling velocity

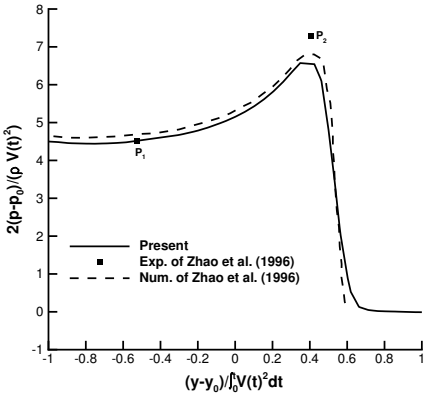
(b) slamming force

FIGURE 6.11: Grid dependence tests for a free falling wedge (a) Time history of falling velocity (b) Time history of slamming force. The present numerical results are compared with experimental and numerical results of Zhao et al. [8].

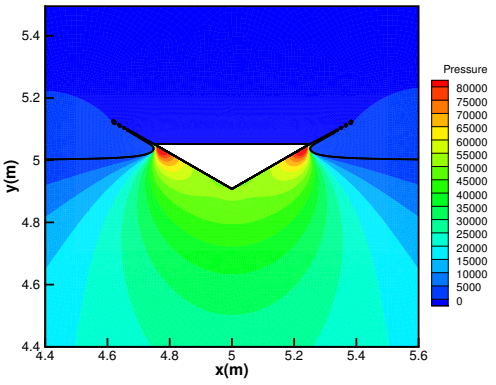
mixed physical quantities. And the interaction of two fluids are formulated by adding the surface tension term in the governing equations, where the interface is captured by THINC-QQ method. Fluid flows are coupled with moving body by solving governing equations of two-phase flow on moving grids. Finally, we verify the numerical model by calculating cases, such as wave generation, water exit of a circular cylinder and free falling of a wedge. The good agreement with reference results indicates the feasibility of present model.



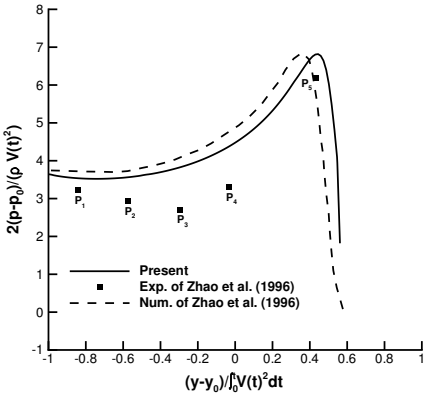

(a) Pressure contour



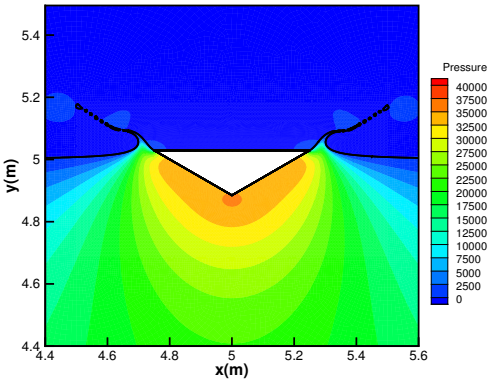

(b) Pressure along wedge boundary



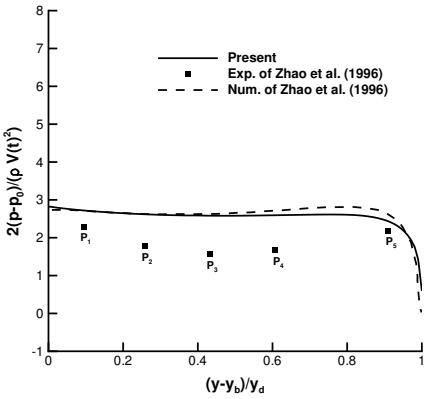

(c) Pressure contour




(d) Pressure along wedge boundary




(e) Pressure contour




(f) Pressure along wedge boundary

FIGURE 6.12: Pressure contour (left panels) and the comparisons of pressure along the wedge boundary (right panels) for different time instants. (a) (b) t=0.00435s; (c) (d) t=0.0158s; (e) (f) t=0.0202s. $V(t)$ denotes the wedge velocity at time $t$. $y$ is the vertical coordinate on the wedge surface. $p_0$ is the initial pressure at the nose of wedge. $y_0$ is the initial vertical coordinate of wedge nose, i.e. $y_0 = 5m$. $y_b$ is the vertical coordinate of wedge nose at time $t$ and $y_d$ is the draft of the body.

# Chapter 7

# Conclusions and future work

## 7.1 Full picture of this thesis

In the numerical simulation of fluid flows with moving interfaces or boundaries, we prefer to use interface tracking technique to achieve accurate solutions around the interfaces. In this study, we proposed a moving mesh framework for fluid flows based on multi-moment finite volume method, where two kinds of moments, VIA and PV, are treated as prognostic computational variables. Moreover, with the proposed fluid model, we constructed a practical and reliable numerical solver for fluid-structure interaction problems of engineering applications.

To describe fluid flows on an arbitrarily moving configuration, Arbitrary Lagrangian Eulerian (ALE) viewpoint is adopted. Thus, governing equations are formulated into ALE integral form and differential form for updating two kinds of moments, VIA and PV, respectively. The VIAs are computed by a finite volume method (FVM) that ensures the rigorous numerical conservativeness, while the PVs are defined at cell vertices and updated efficiently by a finite difference formulation. By employing both VIA and PVs at vertices of each cell as computational variables, multi-moment finite volume method possesses at least following advantages for moving mesh and FSI computations:

- With locally defined DOFs, quadratic or higher-order polynomials can be easily built on unstructured grids.
- PVs are always available at grid points, and can be directly used in moving mesh and FSI computations with efficiency and accuracy.
- The VIA is solved by an FVM formulation and thus rigorously conserved.

This idea is treated as the basis of our study to develop numerical models on moving grids for both compressible and incompressible fluid flows, and finally we want to construct numerical solver for practical FSI computations.

Chapter 3 presents the multi-moment finite volume ALE scheme for Euler equations. Euler equations are cast into integral form with respect to conservative variables for updating VIA, and differential form in terms of primitive variables for solving PVs at cell vertices. We started from one dimensional scheme by using CIP-CSL3 reconstruction and then extended to two dimensional case with multi-moment finite volume reconstruction. In 2D, the edges of computational mesh are always kept in straight-line during the movement, which simplifies the numerical procedure for satisfying GCL but does not degrade the convergence rate for ALE framework. A uniformly 3rd-order accuracy is achieved for both 1D and 2D case. Accuracy test in 1D shows that, with the same mesh resolution, numerical errors of present scheme reduce to $15 \sim 20\%$ of a conventional 3rd-order finite volume scheme no matter mesh moves or not, while the computational cost only increases $20 \sim 25\%$. To explore the robustness of 2D moving mesh computation, we run the case of isentropic vortex until the program stops, and roughly conclude that the present scheme can endure the worst mesh quality of maximum non-orthogonality of around $70^o$. Numerical tests also reveal that the Lagrangian framework tracks the contact discontinuities well in compressible flows.

Chapter 4 extends the multi-moment finite volume method to solve viscous compressible Navier-Stokes equations on moving grids. Numerical formulations for inviscid fluxes are similar with Chapter 3, while the PV moments are calculated from the differential equation in respect of conservative variables. Thus, it is more direct for adding viscous term. With the finite volume form of viscous fluxes for VIA moments, viscous term for PV moments is simply calculated by $\mathcal{TEC}$ formula, which interpolates the viscous effect from the VIA to PV moments. For solving compressible flows involved with force moving body, we implement an efficient and robust moving mesh method, radial basis function (RBF) interpolation, which transfers the boundary movement to internal mesh points. To verify the numerical model, benchmark tests for compressible flows with high Mach number are firstly calculated, and then fluid flows past over a forced oscillating cylinder are investigated. In this phenomenon, the vortex structure is significantly affected by the oscillating amplitude and frequency. We observe the lock-in phenomena by setting cylinder oscillating frequency $f = 0.9$ and $f = 1.1$, and oscillating amplitude $A = 0.3$, which agrees well with published studies. We evaluate the efficiency of RBF interpolation, the RBF code takes only $3.4\%$ of the total computational time by using coarsening technique in single-core calculation. Numerical

results demonstrate the capability of present solver for solving viscous compressible flows involved with forced moving body.

Chapter 5 presents the multi-moment finite volume method for incompressible flows on unstructured moving grids and a numerical solver for fluid-rigid body interaction. Fluid flows are described by incompressible Navier-Stokes equations, which are solved by pressure projection procedure. Both VIA and PV of velocity field are treated as prognostic computational variables and updated separately. With PVs defined at cell vertices, it becomes more accurate and convenient to treat the multi-material interfaces. Thus, we construct a FSI numerical solver, where the structures are simplified to be rigid bodies and treated as a mass-spring-damping system for the sake of verification. We have devised two coupling schemes, i.e. an explicit weak coupling scheme and a semi-implicit strong coupling scheme, to formulate the interactions between fluid and moving solid with a wide range of mass ratios. Various benchmark tests are calculated to verify the present numerical solver. One typical numerical test is about VIV of a circular cylinder. We compute the case with mass ratio $m^* = 1$ by explicit weak coupling scheme for efficiency, while the case with mass ratio $m^* = 0.5$ is calculated by semi-implicit coupling scheme to ensure the numerical stability. In practice, the iterations for fluid-solid implicit coupling rapidly reduce to only $2 \sim 3$ times after short period of calculation, which verifies the efficiency of semi-implicit coupling. A 3D case about a VIV of an elastically mounted sphere with $Re = 300$ is finally computed. The sphere moves almost in a straight-line in transverse plane after periodic vortex formed. And the length of line path of our result is $0.72D$, which is very close to Behara's result $0.70D$.

Chapter 6 presents a preliminary numerical solver for a moving body in free surface flow. In this problem, three kinds of materials (gas, liquid and solid) co-exist and interact. We use moving grids to track the boundary of fluid and solid, and capture the free surface of two-phase flow on a modestly moving grids. Based on one-fluid model, two kinds of fluids are described by incompressible Navier-Stokes equations with a set of averaged physical quantities. Two kinds of fluids are distinguished by VOF function, whose transportation represents the evolution of free surface. The advection equation of VOF function is specially solved by an algebraic interface capturing method, THINC-QQ scheme, which is efficient and can quadratically represents the interfaces on unstructured grids. In this work, the movement of solid body can be directly treated as the boundary conditions of fluid domain. Numerical tests, such as wave generation by moving paddle, water exit of a circular cylinder, and free falling of a wedge, are calculated to show the feasibility of present solver.

## 7.2 Major contribution of current study

The novelty and major contribution of this study can be categorized into following two parts:

1. We present a novel moving mesh framework for fluid flows based on multi-moment finite volume method. By employing both VIA and PV, it not only ensures rigorous numerical conservativeness as same as conventional FVM, but also provides a simple and efficient way for quadratic or higher-order polynomial reconstruction on compact stencils especially for unstructured grids. More importantly, with PVs defined at cell vertices, the solution points can coincide with interfaces between different materials which benefits the accuracy and efficiency of interface tracking. During mesh movement, cell edges (surfaces) always keep in straight-line (plane), which simplifies the numerical procedure for satisfying GCL but does not degrade the convergence rate in ALE framework. A uniformly third order accuracy can be achieved for 2D arbitrarily moving mesh, which is rarely obtained in conventional FVM.

2. We build practical and reliable FSI solvers for engineering applications. An efficient and robust moving mesh technique, RBF interpolation, is included in our numerical solver to transfer the boundary movement to internal mesh points, which is of great significance for practical use. For fluid-solid interaction, we devised not only explicit weak coupling, but also semi-implicit strong coupling where several iterations are performed at each time step to realize the energy equilibrium through the interface. We also extensively solve gas-liquid-solid interaction, by using an algebraic interface capturing method, THINC-QQ scheme, for representing and evolving free surface of two-phase flow, and tracking the fluid-solid interface by numerical model on moving grids. Finally, we can solve a large class of FSI problems and numerical results demonstrate the appealing performance of present solver.

## 7.3 Limitations and future work

In the end, we mention the limitations or deficiencies of present work and point out the directions for future studies. The limitations or deficiencies are firstly listed as follows:

1. The present numerical solver is devised on arbitrarily moving grids, but undoubtedly limited by the mesh quality. As shown in the benchmark test of isentropic vortex, the present numerical model can endure the mesh deformation to the maximum non-orthogonality of around $70^o$.

2. Though we have implemented the RBF interpolation for smoothing the computational mesh, it is hard for handling large body movement, especially for large rotation, since the mesh topology does not change during the movement.

3. We realize that the present solver is hard to solve fluid flows involved with two collision bodies without extra techniques. A possible way to remedy this deficiency may be by employing overset mesh method.

Directions of our future work are summarized from the following four parts:

1. We have presented the one-way coupling solver for compressible flows involved with forced moving body. However, the interaction between fluid and structure, namely, two-way coupling, needs further study in the framework of multi-moment finite volume method. The difficulty is to devise strong coupling schemes, since the compressible Navier-Stokes equations should be solved implicitly which is not straightforward by using multi-moment finite volume method.

2. We are considering including remesh method with the change of mesh topology and remapping procedures to improve the robustness of present solver. Futhermore, we also want to try overset mesh method for practical applications.

3. For gas-liquid-solid interaction, we have just tested the numerical model for a few simple benchmark tests in 2D. 3D applications and efficiency improvement of parallel computing are the next step of our study.

4. In CFD application, especially for two-phase flow, numerical viscosity is generally larger than physical viscosity. Considering the thin boundary layer on the solid surface, we want to try slip boundary condition for two-phase flows with moving bodies in our future study.

# Appendix A

# Eigenstructures of quasilinear form of Euler equation

## A.1 *Form* 1 in 1D

The quasi-linear form of 1D Euler equation in terms of primitive variables $\mathbf{W} = [\rho, u, p]$ can be written as

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} = 0, \tag{A.1}$$

with coefficient matrices

$$\boldsymbol{A}(\mathbf{W}) = \begin{bmatrix} u - u_g & \rho & 0 \\ 0 & u - u_g & \frac{1}{\rho} \\ 0 & \gamma p & u - u_g \end{bmatrix}$$

where all variables or notations are explained in section 3.1.2. The eigenvalues of matrix $\boldsymbol{A}(\mathbf{W})$ are

$$\lambda_1^{\boldsymbol{A}} = u - u_g - c, \lambda_2^{\boldsymbol{A}} = u - u_g, \lambda_3^{\boldsymbol{A}} = u - u_g + c, \tag{A.2}$$

with $c = \sqrt{\frac{\gamma p}{\rho}}$ being sound speed. The corresponding left and right eigenvectors are given by

$$\boldsymbol{L_A} = \begin{bmatrix} 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} \\ 1 & 0 & -\frac{1}{c^2} \\ 0 & \frac{\rho}{2c} & \frac{1}{2c^2} \end{bmatrix}, \boldsymbol{R_A} = \begin{bmatrix} 1 & 1 & 1 \\ -\frac{c}{\rho} & 0 & \frac{c}{\rho} \\ c^2 & 0 & c^2 \end{bmatrix}.$$

## A.2 *Form* 1 in 2D

The quasi-linear form of 2D Euler equation in terms of primitive variables $\mathbf{W} = [\rho, u, v, p]$ can be written as

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} + \boldsymbol{B}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial y} = 0, \tag{A.3}$$

with coefficient matrices

$$\boldsymbol{A}(\mathbf{W}) = \begin{bmatrix} u-u_g & \rho & 0 & 0 \\ 0 & u-u_g & 0 & \frac{1}{\rho} \\ 0 & 0 & u-u_g & 0 \\ 0 & \gamma p & 0 & u-u_g \end{bmatrix}, \boldsymbol{B}(\mathbf{W}) = \begin{bmatrix} v-v_g & 0 & \rho & 0 \\ 0 & v-v_g & 0 & 0 \\ 0 & 0 & v-v_g & \frac{1}{\rho} \\ 0 & 0 & \gamma p & v-v_g \end{bmatrix},$$

where all variables or notations are explained in section 3.1.2. The eigenvalues of matrix $\boldsymbol{A}(\mathbf{W})$ are

$$\lambda_1^{\boldsymbol{A}} = u - u_g - c, \lambda_2^{\boldsymbol{A}} = u - u_g, \lambda_3^{\boldsymbol{A}} = u - u_g, \lambda_4^{\boldsymbol{A}} = u - u_g + c, \tag{A.4}$$

with $c = \sqrt{\frac{\gamma p}{\rho}}$ being sound speed. The corresponding left and right eigenvectors are given by

$$\boldsymbol{L_A} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & -\frac{1}{2\rho c} \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -\frac{1}{c^2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2\rho c} \end{bmatrix}, \boldsymbol{R_A} = \begin{bmatrix} -\frac{\rho}{c} & 0 & 1 & \frac{\rho}{c} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ -\rho c & 0 & 0 & \rho c \end{bmatrix}.$$

The eigenvalues of matrix $\boldsymbol{B}(\mathbf{W})$ are

$$\lambda_1^{\boldsymbol{B}} = v - v_g - c, \lambda_2^{\boldsymbol{B}} = v - v_g, \lambda_3^{\boldsymbol{B}} = v - v_g, \lambda_4^{\boldsymbol{B}} = v - v_g + c. \tag{A.5}$$

The corresponding left and right eigenvectors are given by

$$\boldsymbol{L_B} = \begin{bmatrix} 0 & 0 & \frac{1}{2} & -\frac{1}{2\rho c} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -\frac{1}{c^2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2\rho c} \end{bmatrix}, \boldsymbol{R_B} = \begin{bmatrix} -\frac{\rho}{c} & 0 & 1 & \frac{\rho}{c} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ -\rho c & 0 & 0 & \rho c \end{bmatrix}.$$

## A.3 *Form* 2 in 2D

The quasi-linear form of 2D Euler equation in terms of conservative variables $\mathbf{U} = [\rho, \rho u, \rho v, \rho E]$ can be written as

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\chi} + \boldsymbol{A}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} + \boldsymbol{B}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial y} = 0, \tag{A.6}$$

with coefficient matrices

$$\boldsymbol{A}(\mathbf{U}) = \begin{bmatrix} -u_g & 1 & 0 & 0 \\ -u^2 + \frac{1}{2}(\gamma-1)\boldsymbol{V}^2 & (3-\gamma)u - u_g & (1-\gamma)v & \gamma - 1 \\ -uv & v & u - u_g & 0 \\ u\left[\frac{1}{2}(\gamma-1)\boldsymbol{V}^2 - H\right] & H - (\gamma-1)u^2 & (1-\gamma)uv & \gamma u - u_g \end{bmatrix},$$

$$\boldsymbol{B}(\mathbf{U}) = \begin{bmatrix} -v_g & 0 & 1 & 0 \\ -uv & v - v_g & u & 0 \\ -v^2 + \frac{1}{2}(\gamma-1)\boldsymbol{V}^2 & (1-\gamma)u & (3-\gamma)v - v_g & \gamma - 1 \\ v\left[\frac{1}{2}(\gamma-1)\boldsymbol{V}^2 - H\right] & (1-\gamma)uv & H - (\gamma-1)v^2 & \gamma v - v_g \end{bmatrix},$$

where all variables or notations are explained in section 3.1.2. The eigenvalues of matrix $\boldsymbol{A}(\mathbf{U})$ are

$$\lambda_1^{\boldsymbol{A}} = u - u_g - c, \lambda_2^{\boldsymbol{A}} = u - u_g, \lambda_3^{\boldsymbol{A}} = u - u_g, \lambda_4^{\boldsymbol{A}} = u - u_g + c. \tag{A.7}$$

The corresponding left and right eigenvectors are given by

$$\boldsymbol{L_A} = \begin{bmatrix} (\gamma-1)\frac{V_m}{2c^2} + \frac{u}{2c} & -(\gamma-1)\frac{u}{2c^2} - \frac{1}{2c} & -(\gamma-1)\frac{v}{2c^2} & \frac{\gamma-1}{2c^2} \\ -v & 0 & 1 & 0 \\ 1 - (\gamma-1)\frac{V_m}{c^2} & (\gamma-1)\frac{u}{c^2} & (\gamma-1)\frac{v}{c^2} & -\frac{\gamma-1}{c^2} \\ (\gamma-1)\frac{V_m}{2c^2} - \frac{u}{2c} & -(\gamma-1)\frac{u}{2c^2} + \frac{1}{2c} & -(\gamma-1)\frac{v}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix},$$

$$\boldsymbol{R_A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u - c & 0 & u & u + c \\ v & 1 & v & v \\ H - uc & v & V_m & H + uc \end{bmatrix}.$$

The eigenvalues of matrix $\boldsymbol{B}(\mathbf{U})$ are

$$\lambda_1^{\boldsymbol{B}} = v - v_g - c, \lambda_2^{\boldsymbol{B}} = v - v_g, \lambda_3^{\boldsymbol{B}} = v - v_g, \lambda_4^{\boldsymbol{B}} = v - v_g + c. \tag{A.8}$$

The corresponding left and right eigenvectors are given by

$$\boldsymbol{L_B} = \begin{bmatrix} (\gamma-1)\frac{V_m}{2c^2} + \frac{v}{2c} & -(\gamma-1)\frac{u}{2c^2} & -(\gamma-1)\frac{v}{2c^2} - \frac{1}{2c} & \frac{\gamma-1}{2c^2} \\ -u & 1 & 0 & 0 \\ 1-(\gamma-1)\frac{V_m}{c^2} & (\gamma-1)\frac{u}{c^2} & (\gamma-1)\frac{v}{c^2} & -\frac{\gamma-1}{c^2} \\ (\gamma-1)\frac{V_m}{2c^2} - \frac{v}{2c} & -(\gamma-1)\frac{u}{2c^2} & -(\gamma-1)\frac{v}{2c^2} + \frac{1}{2c} & \frac{\gamma-1}{2c^2} \end{bmatrix},$$

$$\boldsymbol{R_B} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u & 1 & u & u \\ v-c & 0 & v & v+c \\ H-vc & u & V_m & H+vc \end{bmatrix}.$$

with $V_m = \frac{1}{2}\boldsymbol{V}^2$.

# Appendix B

# Multi-moment constraint conditions

## B.1 Triangular element

Given the reconstruction polynomial for triangular element $\Omega_i$ on local coordinates

$$\Phi_i(\xi,\eta) = c_0 + c_1\xi + c_2\eta + c_3\xi\eta + c_4\xi^2 + c_5\eta^2, \tag{B.1}$$

the coefficients can be determined as

$$\begin{cases} \phi_i(0,0) = \phi_{i1} \\ \phi_i(1,0) = \phi_{i2} \\ \phi_i(0,1) = \phi_{i3} \\ \frac{1}{2}\iint_{\Omega} \phi_i(\xi,\eta)d\xi d\eta = \overline{\phi}_i \\ \left.\dfrac{\partial \phi_i(\xi,\eta)}{\partial \xi}\right|_{(\frac{1}{3},\frac{1}{3})} = \phi_{\xi ic} \\ \left.\dfrac{\partial \phi_i(\xi,\eta)}{\partial \eta}\right|_{(\frac{1}{3},\frac{1}{3})} = \phi_{\eta ic} \end{cases} \Longrightarrow \begin{cases} c_0 = \phi_{i1} \\ c_1 = -\frac{10}{3}\phi_{i1} - \frac{4}{3}\phi_{i2} + \frac{2}{3}\phi_{i3} + \phi_{\xi ic} - 2\phi_{\eta ic} + 4\overline{\phi}_i \\ c_2 = -\frac{10}{3}\phi_{i1} + \frac{2}{3}\phi_{i2} - \frac{4}{3}\phi_{i3} + \phi_{\eta ic} - 2\phi_{\xi ic} + 4\overline{\phi}_i \\ c_3 = \frac{16}{3}\phi_{i1} - \frac{2}{3}\phi_{i2} - \frac{2}{3}\phi_{i3} + 2\phi_{\xi ic} + 2\phi_{\eta ic} - 4\overline{\phi}_i \\ c_4 = \frac{7}{3}\phi_{i1} + \frac{7}{3}\phi_{i2} - \frac{2}{3}\phi_{i3} - \phi_{\xi ic} + 2\phi_{\eta ic} - 4\overline{\phi}_i \\ c_5 = \frac{7}{3}\phi_{i1} - \frac{2}{3}\phi_{i2} + \frac{7}{3}\phi_{i3} - \phi_{\eta ic} + 2\phi_{\xi ic} - 4\overline{\phi}_i, \end{cases} \tag{B.2}$$

with $\overline{\phi}_i$ being the volume integrated average of cell $\Omega_i$, $\phi_{ik}(k = 1, 2, 3)$ point values at cell vertices, and $(\phi_{\xi ic}, \phi_{\eta ic})$ first order derivative with respect to local coordinates.

## B.2 Quadrilateral element

Given the reconstruction polynomial for quadrilateral element $\Omega_i$ on local coordinates

$$\Phi_i(\xi,\eta) = c_0 + c_1\xi + c_2\eta + c_3\xi^2 + c_4\eta^2 + c_5\xi\eta + c_6\xi^2\eta + c_7\xi\eta^2 + c_8\xi^2\eta^2. \quad \text{(B.3)}$$

the coefficients can be determined as

$$\begin{cases} \Phi_i(-1,-1) = \phi_{i1} \\ \Phi_i(1,-1) = \phi_{i2} \\ \Phi_i(1,1) = \phi_{i3} \\ \Phi_i(-1,1) = \phi_{i3} \\ \dfrac{1}{4}\displaystyle\iint_{\omega} \Phi_i(\xi,\eta)d\xi d\eta = \overline{\phi}_i \\ \left.\dfrac{\partial\Phi_i(\xi,\eta)}{\partial\xi}\right|_{(0,0} = \phi_{\xi ic} \\ \left.\dfrac{\partial\Phi_i(\xi,\eta)}{\partial\eta}\right|_{(0,0} = \phi_{\eta ic} \\ \left.\dfrac{\partial^2\Phi_i(\xi,\eta)}{\partial\xi^2}\right|_{(0,0} = \phi_{\xi^2 ic} \\ \left.\dfrac{\partial^2\Phi_i(\xi,\eta)}{\partial\eta^2}\right|_{(0,0} = \phi_{\eta^2 ic} \end{cases} \quad \text{(B.4)}$$

$$\Longrightarrow \quad \begin{cases} c_0 = -\frac{1}{32}(\phi_{i1}+\phi_{i2}+\phi_{i3}+\phi_{i4}) + \frac{9}{8}\overline{\phi}_i - \frac{1}{8}(\phi_{\xi^2 ic}+\phi_{\eta^2 ic}) \\ c_1 = \phi_{\xi ic} \\ c_2 = \phi_{\eta ic} \\ c_3 = \frac{1}{2}\phi_{\xi^2 ic} \\ c_4 = \frac{1}{2}\phi_{\eta^2 ic} \\ c_5 = \frac{1}{4}(\phi_{i1}-\phi_{i2}+\phi_{i3}-\phi_{i4}) \\ c_6 = \frac{1}{4}(-\phi_{i1}-\phi_{i2}+\phi_{i3}+\phi_{i4}) - \phi_{\eta ic} \\ c_7 = \frac{1}{4}(-\phi_{i1}+\phi_{i2}+\phi_{i3}-\phi_{i4}) - \phi_{\xi ic} \\ c_8 = \frac{9}{32}(\phi_{i1}+\phi_{i2}+\phi_{i3}+\phi_{i4}) - \frac{9}{8}\overline{\phi}_i - \frac{3}{8}(\phi_{\xi^2 ic}+\phi_{\eta^2 ic}) \end{cases} \quad \text{(B.5)}$$

with $\overline{\phi}_i$ being the volume integrated average of cell $\Omega_i$, $\phi_{ik}(k = 1, 2, 3, 4)$ point values at cell vertices, $(\phi_{\xi ic}, \phi_{\eta ic})$ and $(\phi_{\xi^2 ic}, \phi_{\eta^2 ic})$ first and second order derivatives with respect to local coordinates respectively.

## B.3 Tetrahedral element

Given the reconstruction polynomial for tetrahedral element $\Omega_i$ on local coordinates

$$\Phi_i(\xi,\eta,\zeta) = c_0 + c_1\xi + c_2\eta + c_3\zeta + c_4\xi^2 + c_5\eta^2 + c_6\zeta^2 + c_7(\xi\eta + \eta\zeta + \xi\zeta), \quad \text{(B.6)}$$

the coefficients can be determined as

$$\begin{cases} \Phi_i(0,0,0) = \phi_{i1} \\ \Phi_i(1,0,0) = \phi_{i2} \\ \Phi_i(0,1,0) = \phi_{i3} \\ \Phi_i(0,0,1) = \phi_{i4} \\ 6\iiint_{\omega} \Phi_i(\xi,\eta,\zeta)d\xi d\eta d\zeta = \overline{\phi}_i \\ \left.\dfrac{\partial\Phi_i(\xi,\eta,\zeta)}{\partial\xi}\right|_{(\frac{1}{4},\frac{1}{4})} = \phi_{\xi i} \\ \left.\dfrac{\partial\Phi_i(\xi,\eta,\zeta)}{\partial\eta}\right|_{(\frac{1}{4},\frac{1}{4})} = \phi_{\eta i} \\ \left.\dfrac{\partial\Phi_i(\xi,\eta,\zeta)}{\partial\zeta}\right|_{(\frac{1}{4},\frac{1}{4})} = \phi_{\zeta i} \end{cases}$$

$$\Longrightarrow \quad \begin{cases} c_0 = \overline{\phi}_i \\ c_1 = -\frac{17}{6}\phi_{i1} - \frac{5}{6}\phi_{i2} + \frac{1}{6}\phi_{i3} + \frac{1}{6}\phi_{i4} + \frac{10}{3}\overline{\phi}_i + \phi_{\xi i} - \phi_{\eta i} - \phi_{\zeta i} \\ c_2 = -\frac{17}{6}\phi_{i1} + \frac{1}{6}\phi_{i2} - \frac{5}{6}\phi_{i3} + \frac{1}{6}\phi_{i4} + \frac{10}{3}\overline{\phi}_i - \phi_{\xi i} + \phi_{\eta i} - \phi_{\zeta i} \\ c_3 = -\frac{17}{6}\phi_{i1} + \frac{1}{6}\phi_{i2} + \frac{1}{6}\phi_{i3} - \frac{5}{6}\phi_{i4} + \frac{10}{3}\overline{\phi}_i - \phi_{\xi i} - \phi_{\eta i} + \phi_{\zeta i} \\ c_4 = \frac{11}{6}\phi_{i1} + \frac{11}{6}\phi_{i2} - \frac{1}{6}\phi_{i3} - \frac{1}{6}\phi_{i4} - \frac{10}{3}\overline{\phi}_i - \phi_{\xi i} + \phi_{\eta i} + \phi_{\zeta i} \\ c_5 = \frac{11}{6}\phi_{i1} - \frac{1}{6}\phi_{i2} + \frac{11}{6}\phi_{i3} - \frac{1}{6}\phi_{i4} - \frac{10}{3}\overline{\phi}_i + \phi_{\xi i} - \phi_{\eta i} + \phi_{\zeta i} \\ c_6 = \frac{11}{6}\phi_{i1} - \frac{1}{6}\phi_{i2} - \frac{1}{6}\phi_{i3} + \frac{11}{6}\phi_{i4} - \frac{10}{3}\overline{\phi}_i + \phi_{\xi i} + \phi_{\eta i} - \phi_{\zeta i} \\ c_7 = \frac{23}{6}\phi_{i1} - \frac{1}{6}\phi_{i2} - \frac{1}{6}\phi_{i3} - \frac{1}{6}\phi_{i4} - \frac{10}{3}\overline{\phi}_i + \phi_{\xi i} + \phi_{\eta i} + \phi_{\zeta i} \end{cases} \quad \text{(B.7)}$$

with $\overline{\phi}_i$ being the volume integrated average of cell $\Omega_i$, $\phi_{ik}(k = 1, 2, 3, 4)$ point values at cell vertices, $(\phi_{\xi ic}, \phi_{\eta ic}, \phi_{\zeta ic})$ first order derivatives with respect to local coordinates respectively.

# Appendix C

# Riemann Solvers on moving domain

We use the symbol $\Omega_{ij}$ to denote the surrounding cells adjacent to the target cell $\Omega_i$ with the common boundary segment $\Gamma_{ij}$ $(j = 1, 2, \cdots, \mathbb{J})$. Given the approximated values $\mathbf{U}_i$ and $\mathbf{U}_{ij}$ at Gaussian quadrature points on edge $\Gamma_{ij}$ which are computed from the reconstruction functions over $\Omega_i$ and $\Omega_{ij}$ respectively, Riemann solvers are used to calculate the flux at Gaussian quadrature point on edge $\Gamma_{ij}$, whose outward unit normal vector and length magnitude are $\mathbf{n}_{ij}$ and $|\Gamma_{ij}|$ respectively. As mentioned, the grid velocity varies linearly along edge $\Gamma_{ij}$ and denoted as $\boldsymbol{u}_{gij}$.

We give known vectors of conservative variables and fluxes approximated from left and right reconstructions as below

$$\begin{aligned}
\mathbf{U}_i &= \begin{pmatrix} \rho_i \\ \mathbf{M}_i \\ (\rho E)_i \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}_i) = \begin{pmatrix} (\boldsymbol{u}_i - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} \rho_i \\ (\boldsymbol{u}_i - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} \mathbf{M}_i + p_i \bar{\bar{I}} \\ (\boldsymbol{u}_i - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} (\rho E)_i + p_i \boldsymbol{u}_i \end{pmatrix} \\
\mathbf{U}_{ij} &= \begin{pmatrix} \rho_{ij} \\ \mathbf{M}_{ij} \\ (\rho E)_{ij} \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}_{ij}) = \begin{pmatrix} (\boldsymbol{u}_{ij} - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} \rho_{ij} \\ (\boldsymbol{u}_{ij} - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} \mathbf{M}_{ij} + p_{ij} \bar{\bar{I}} \\ (\boldsymbol{u}_{ij} - \boldsymbol{u}_{gij}) \cdot \mathbf{n}_{ij} (\rho E)_{ij} + p_{ij} \boldsymbol{u}_{ij} \end{pmatrix}
\end{aligned} \tag{C.1}$$

and then present the fluxes by different Riemann solvers in following sections.

## C.1 Lax-Friedrichs flux

Lax-Friedrichs flux on edge $\Gamma_{ij}$ for moving grids is denoted by $\mathbf{F}_{ij}^{LF}$ and formulated as

$$\mathbf{F}_{ij}^{LF} = \frac{1}{2} \left[ \mathbf{F}(\mathbf{U}_i) + \mathbf{F}(\mathbf{U}_{ij}) - \alpha_{ij} \left( \mathbf{U}_{ij} - \mathbf{U}_i \right) \right] \tag{C.2}$$

where $\alpha_{ij}$ is taken as

$$\alpha_{ij} = \max\left(|u_{n_i} - c_i|, |u_{n_i} + c_i|, |u_{n_{ij}} - c_{ij}|, |u_{n_{ij}} + c_{ij}|\right) \tag{C.3}$$

with $u_{n_i} = (\mathbf{u}_i - \mathbf{u}_{gij}) \cdot \mathbf{n}_{ij}, u_{n_{ij}} = (\mathbf{u}_{ij} - \mathbf{u}_{gij}) \cdot \mathbf{n}_{ij}$, $\boldsymbol{u}_i = \frac{\boldsymbol{M}_i}{\rho_i}$, $\boldsymbol{u}_{ij} = \frac{\boldsymbol{M}_{ij}}{\rho_{ij}}$, $c_i = \sqrt{\frac{\gamma p_i}{\rho_i}}$, $c_{ij} = \sqrt{\frac{\gamma p_{ij}}{\rho_{ij}}}$, $p_i = (\gamma - 1)\left[(\rho E)_i - \frac{M_i^2}{2\rho_i}\right]$ and $p_{ij} = (\gamma - 1)\left[(\rho E)_{ij} - \frac{M_{ij}^2}{2\rho_{ij}}\right]$.

## C.2 HLL flux

The HLL flux on edge $\Gamma_{ij}$ for moving grids [83] is denoted by $\mathbf{F}_{ij}^{HLL}$ and formulated as

$$\mathbf{F}_{ij}^{HLL} = \begin{cases} \mathbf{F}(\mathbf{U}_i) & \text{if } 0 \leqslant S_i \\ \frac{S_{ij}\mathbf{F}(\mathbf{U}_i) - S_i\mathbf{F}(\mathbf{U}_{ij}) + S_i S_{ij}(\mathbf{U}_{ij} - \mathbf{U}_i)}{S_{ij} - S_i} & \text{if } S_i \leqslant 0 \leqslant S_{ij} \\ \mathbf{F}(\mathbf{U}_{ij}) & \text{if } 0 \geqslant S_{ij} \end{cases} \tag{C.4}$$

where signal velocities $S_i$ and $S_{ij}$ are defined as

$$S_i = \min\left[u_{n_i} - c_i, (\hat{\mathbf{u}} - \mathbf{u}_{gij}) \cdot \mathbf{n} - \hat{c}\right], S_{ij} = \max\left[u_{n_{ij}} + c_{ij}, (\hat{\mathbf{u}} - \mathbf{u}_{gij}) \cdot \mathbf{n} + \hat{c}\right], \tag{C.5}$$

with $\hat{\mathbf{u}}$ and $\hat{c}$ being the values computed by Roe-averaging for the velocity and the sound speed.

## C.3 HLLC flux

The HLLC flux on edge $\Gamma_{ij}$ for moving grids [83] is denoted by $\mathbf{F}_{ij}^{HLLC}$ and formulated as

$$\mathbf{F}_{ij}^{HLLC} = \begin{cases} \mathbf{F}(\mathbf{U}_i) & S_i > 0 \\ \mathbf{F}(\mathbf{U}_i^\star) & S_i \leqslant 0 < S^\star \\ \mathbf{F}(\mathbf{U}_{ij}^\star) & S^\star \leqslant 0 \leqslant S_{ij} \\ \mathbf{F}(\mathbf{U}_{ij}) & S_{ij} < 0 \end{cases} \tag{C.6}$$

where

$$\mathbf{U}_i^\star = \begin{pmatrix} \rho_i^\star \\ \mathbf{M}_i^\star \\ (\rho E)_i^\star \end{pmatrix} = \frac{1}{S_i - S^\star} \begin{pmatrix} (S_i - u_{n_i})\rho_i \\ (S_i - u_{n_i})\mathbf{M}_i + (p^\star - p_i)\mathbf{n} \\ (S_i - u_{n_i})(\rho E)_i - p_i u_{n_i} + p^\star S^\star \end{pmatrix}, \tag{C.7}$$

$$\mathbf{U}_{ij}^{\star} = \begin{pmatrix} \rho_{ij}^{\star} \\ \mathbf{M}_{ij}^{\star} \\ (\rho E)_{ij}^{\star} \end{pmatrix} = \frac{1}{S_{ij} - S^{\star}} \begin{pmatrix} (S_{ij} - u_{n_{ij}})\rho_{ij} \\ (S_{ij} - u_{n_{ij}})\mathbf{M}_{ij} + (p^{\star} - p_{ij})\mathbf{n} \\ (S_{ij} - u_{n_{ij}})(\rho E)_{ij} - p_{ij}u_{n_{ij}} + p^{\star}S^{\star} \end{pmatrix}, \quad \text{(C.8)}$$

$$\mathbf{F}(\mathbf{U}_{i}^{\star}) = \begin{pmatrix} S^{\star}\rho_{i}^{\star} \\ S^{\star}\mathbf{M}_{i}^{\star} + p^{\star}\mathbf{n} \\ S^{\star}(\rho E)_{i}^{\star} + (S^{\star} + \mathbf{u}_{gij} \cdot \mathbf{n})p^{\star} \end{pmatrix}, \quad \text{(C.9)}$$

$$\mathbf{F}(\mathbf{U}_{ij}^{\star}) = \begin{pmatrix} S^{\star}\rho_{ij}^{\star} \\ S^{\star}\mathbf{M}_{ij}^{\star} + p^{\star}\mathbf{n} \\ S^{\star}(\rho E)_{ij}^{\star} + (S^{\star} + \mathbf{u}_{gij} \cdot \mathbf{n})p^{\star} \end{pmatrix}, \quad \text{(C.10)}$$

$$p^{\star} = \rho_i(u_{n_i} - S_i)(u_{n_i} - S^{\star}) + p_i = \rho_{ij}(u_{n_{ij}} - S_{ij})(u_{n_{ij}} - S^{\star}) + p_{ij}, \quad \text{(C.11)}$$

$$u_{n_i} = (\mathbf{u}_i - \mathbf{u}_{gij}) \cdot \mathbf{n}, \quad u_{n_{ij}} = (\mathbf{u}_{ij} - \mathbf{u}_{gij}) \cdot \mathbf{n}, \quad \text{(C.12)}$$

$$S^{\star} = \frac{\rho_{ij}u_{n_{ij}}(S_{ij} - u_{n_{ij}}) - \rho_i u_{n_i}(S_i - u_{n_i}) + p_i - p_{ij}}{\rho_{ij}(S_{ij} - u_{n_{ij}}) - \rho_i(S_i - u_{n_i})}. \quad \text{(C.13)}$$

Signal velocities $S_i$ and $S_{ij}$ are defined as

$$S_i = \min\left[u_{n_i} - c_i, (\hat{\mathbf{u}} - \mathbf{u}_{gij}) \cdot \mathbf{n} - \hat{c}\right], S_{ij} = \max\left[u_{n_{ij}} + c_{ij}, (\hat{\mathbf{u}} - \mathbf{u}_{gij}) \cdot \mathbf{n} + \hat{c}\right], \quad \text{(C.14)}$$

with $\hat{\mathbf{u}}$ and $\hat{c}$ being the values computed by Roe-averaging for the velocity and the sound speed.

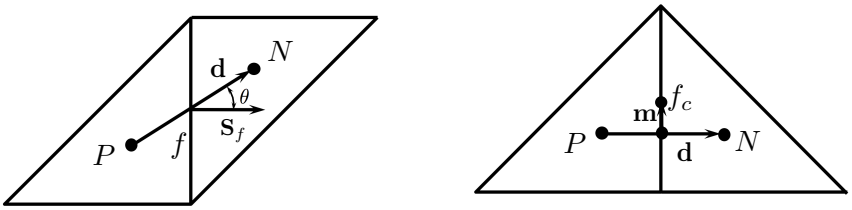
# Appendix D

# Mesh non-orthogonality and skewness

The mesh quality is evaluated based on non-orthogonality and skewness. Following [65], we describe the surface non-orthogonality by the angle $\theta$ between the surface vector $\mathbf{S}_f$ and the vector $\mathbf{d}$ which connects two neighbouring cell centers $P$ and $N$ as shown in Figure D.1 (a). The cell is skewed when the line $\overrightarrow{PN}$ does not go through the surface center $f_c$, thus the degree of skewness is defined as

$$\psi_s = \frac{|\mathbf{m}|}{|\mathbf{d}|}, \tag{D.1}$$

where the vector $\mathbf{m}$ and $\mathbf{d}$ are defined in Figure D.1 (b).




(a) (b)

FIGURE D.1: Schematic diagrams for mesh non-orthogonality (a) and skewness (b).

# Bibliography


[1] G. Koopmann, The vortex wakes of vibrating cylinders at low Reynolds numbers, Journal of Fluid Mechanics 28 (3) (1967) 501–512.

[2] P. Anagnostopoulos, Numerical study of the flow past a cylinder excited transversely to the incident stream. Part 1: Lock-in zone, hydrodynamic forces and wake geometry, Journal of Fluids and Structures 14 (6) (2000) 819–851.

[3] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, J. Fluid Mech. 360 (1998) 249–271.

[4] H. Blackburn, G. Karniadakis, et al., Two-and Three-Dimensional Simulations of Vortex-Induced Vibration Or a Circular Cylinder, in: The Third International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, 1993.

[5] I. Robertson, L. Li, S. Sherwin, P. Bearman, A numerical study of rotational and transverse galloping rectangular bodies, J Fluids Struct 17 (5) (2003) 681–699.

[6] J. Jeong, F. Hussain, On the identification of a vortex, J. Fluid Mech. 285 (1995) 69–94.

[7] M. Greenhow, S. Moyo, Water entry and exit of horizontal circular cylinders, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 355 (1724) (1997) 551–563.

[8] R. Zhao, O. Faltinsen, J. Aarsnes, Water entry of arbitrary two-dimensional sections with and without flow separation, in: Proceedings of the 21st symposium on naval hydrodynamics, Trondheim, Norway, National Academy Press, Washington, DC, USA, 408–423, 1996.

[9] J. H. Ferziger, M. Peric, Computational methods for fluid dynamics, Springer Science & Business Media, 2012.

[10] http://www.claymath.org/millennium-problems.

[11] T. E. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, Encyclopedia of computational mechanics .

[12] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.

[13] C. W. Hirt, A. A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, Journal of computational physics 14 (3) (1974) 227–253.

[14] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodriguez-Ferran, Encyclopedia of Computational Mechanics Vol. 1: Fundamentals., Chapter 14: Arbitrary Lagrangian-Eulerian Methods, 2004.

[15] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[16] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[17] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of computational physics 39 (1) (1981) 201–225.

[18] G. Ryskin, L. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 1. The finite-difference technique, Journal of Fluid Mechanics 148 (1984) 1–17.

[19] G. Ryskin, L. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid, Journal of Fluid Mechanics 148 (1984) 19–35.

[20] T. E. Tezduyar, M. Behr, S. Mittal, A. Johnson, Computation of unsteady incompressible flows with the stabilized finite element methods: space-time formulations, iterative strategies and massively parallel implementations, ASME PRESSURE VESSELS PIPING DIV PUBL PVP., ASME, NEW YORK, NY(USA), 1992, 246 (1992) 7–24.

[21] S. Takagi, Three-dimensional deformation of a rising bubble, in: Proc. German-Japanese Symp. on Multi-Phase Flow, vol. 499, 1994.

[22] J. Donea, S. Giuliani, J.-P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, Computer methods in applied mechanics and engineering 33 (1-3) (1982) 689–723.

[23] T. E. Tezduyar, M. Behr, S. Mittal, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders, Computer methods in applied mechanics and engineering 94 (3) (1992) 353–371.

[24] Y. Bazilevs, K. Takizawa, T. E. Tezduyar, Computational fluid-structure interaction: methods and applications, John Wiley & Sons, 2013.

[25] P. Thomas, C. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA journal 17 (10) (1979) 1030–1037.

[26] T. J. Hughes, G. M. Hulbert, Space-time finite element methods for elastodynamics: formulations and error estimates, Computer methods in applied mechanics and engineering 66 (3) (1988) 339–363.

[27] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains, Computer Methods in Applied Mechanics and Engineering 198 (17-20) (2009) 1585–1595.

[28] W. Boscheri, R. Loubère, M. Dumbser, Direct Arbitrary-Lagrangian–Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws, Journal of Computational Physics 292 (2015) 56–87.

[29] C. Munz, On Godunov-type schemes for Lagrangian gas dynamics, SIAM Journal on Numerical Analysis 31 (1) (1994) 17–42.

[30] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of computational physics 43 (2) (1981) 357–372.

[31] A. Harten, P. D. Lax, B. v. Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM review 25 (1) (1983) 35–61.

[32] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM Journal on Scientific Computing 29 (4) (2007) 1781–1824.

[33] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, Journal of Computational Physics 228 (7) (2009) 2391–2425.

[34] P.-H. Maire, B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, Journal of Computational Physics 228 (3) (2009) 799–821.

[35] J. Cheng, C.-W. Shu, A high order ENO conservative Lagrangian type scheme for the compressible Euler equations, Journal of Computational Physics 227 (2) (2007) 1567–1596.

[36] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, in: Upwind and high-resolution schemes, Springer, 218–290, 1987.

[37] J. Cheng, C.-W. Shu, A third order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations, Communications in Computational Physics 4 (2008) 1008–1024.

[38] M. Dumbser, A. Uuriintsetseg, O. Zanotti, On arbitrary-Lagrangian-Eulerian one-step WENO schemes for stiff hyperbolic balance laws, Communications in Computational Physics 14 (2) (2013) 301–327.

[39] F. Vilar, P.-H. Maire, R. Abgrall, A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids, Journal of Computational Physics 276 (2014) 188–234.

[40] G. Clair, J. M. Ghidaglia, J. P. Perlat, A multi-dimensional finite volume cell-centered direct ALE solver for hydrodynamics, Journal of Computational Physics 326 (2016) 312–333.

[41] W. Boscheri, M. Dumbser, A direct Arbitrary-Lagrangian–Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3D, Journal of Computational Physics 275 (2014) 484–523.

[42] D. J. Mavriplis, C. R. Nastase, On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes, Journal of Computational Physics 230 (11) (2011) 4285–4300.

[43] X. Ren, K. Xu, W. Shyy, A multi-dimensional high-order DG-ALE method based on gas-kinetic theory with application to oscillating bodies, Journal of Computational Physics 316 (2016) 700–720.

[44] B. Van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, Journal of computational physics 23 (3) (1977) 276–299.

[45] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, Journal of computational Physics 32 (1) (1979) 101–136.

[46] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1) (1994) 200–212.

[47] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of computational physics 126 (1) (1996) 202–228.

[48] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, Mathematics of computation 52 (186) (1989) 411–435.

[49] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, Journal of Computational Physics 84 (1) (1989) 90–113.

[50] Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation, Journal of computational physics 178 (1) (2002) 210–251.

[51] Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids: II. Extension to two-dimensional scalar equation, Journal of Computational Physics 179 (2) (2002) 665–697.

[52] Z. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, Progress in Aerospace Sciences 43 (1-3) (2007) 1–41.

[53] E. J. Kubatko, C. Dawson, J. J. Westerink, Time step restrictions for Runge–Kutta discontinuous Galerkin methods on triangular grids, Journal of Computational Physics 227 (23) (2008) 9697–9710.

[54] B. Xie, S. Ii, A. Ikebata, F. Xiao, A multi-moment finite volume method for incompressible Navier–Stokes equations on unstructured grids: volume-average/point-value formulation, Journal of Computational Physics 277 (2014) 138–162.

[55] B. Xie, F. Xiao, Two and three dimensional multi-moment finite volume solver for incompressible Navier–Stokes equations on unstructured grids with arbitrary quadrilateral and hexahedral elements, Computers & Fluids 104 (2014) 40–54.

[56] B. Xie, X. Deng, Z. Sun, F. Xiao, A hybrid pressure–density-based Mach uniform algorithm for 2D Euler equations on unstructured grids by using multi-moment finite volume method, Journal of Computational Physics 335 (2017) 637–663.

[57] X. Deng, B. Xie, F. Xiao, A finite volume multi-moment method with boundary variation diminishing principle for Euler equation on three-dimensional hybrid unstructured grids, Computers & Fluids 153 (2017) 85–101.

[58] C. Farhat, M. Lesoinne, N. Maman, Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution, Int. J. Numer. Methods Fluids 21 (10) (1995) 807–835.

[59] S. Piperno, Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations, Int. J. Numer. Methods Fluids 25 (10) (1997) 1207–1226.

[60] H. T. Ahn, Y. Kallinderis, Strongly coupled flow/structure interactions with a geometrically conservative ALE scheme on general hybrid meshes, Journal of Computational Physics 219 (2) (2006) 671–696.

[61] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (104) (1968) 745–762.

[62] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (2) (1985) 308–323.

[63] A. Naseri, O. Lehmkuhl, I. Gonzalez, A. Oliva, Partitioned semi-implicit methods for simulation of biomechanical fluid-structure interaction problems, in: J. Phys. Conf. Ser., vol. 745, IOP Publishing, 032020, 2016.

[64] A. De Boer, M. Van der Schoot, H. Bijl, Mesh deformation based on radial basis function interpolation, Comput Struct 85 (11-14) (2007) 784–795.

[65] F. M. Bos, Numerical simulations of flapping foil and wing aerodynamics: Mesh deformation using radial basis functions, Ph.D. thesis, Technische Universiteit Delft, http://resolver.tudelft.nl/uuid:edecee10-7db2-443e-8cdf-d41aaddaa1b2, 2010.

[66] O. Estruch, O. Lehmkuhl, R. Borrell, C. P. Segarra, A. Oliva, A parallel radial basis function interpolation method for unstructured dynamic meshes, Computers & Fluids 80 (2013) 44–54.

[67] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley & Sons, 2003.

[68] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, Encyclopedia of Computational Mechanics Second Edition (2017) 1–23.

[69] T. Belytschko, W. K. Liu, B. Moran, K. Elkhodary, Nonlinear finite elements for continua and structures, John wiley & sons, 2013.

[70] P. Wesseling, Principles of computational fluid dynamics, vol. 29, Springer Science & Business Media, 2009.

[71] D. J. Mavriplis, Z. Yang, Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes, Journal of Computational Physics 213 (2) (2006) 557–573.

[72] F. Xiao, T. Yabe, Completely conservative and oscillationless semi-Lagrangian schemes for advection transportation, Journal of computational physics 170 (2) (2001) 498–522.

[73] S. Ii, F. Xiao, CIP/multi-moment finite volume method for Euler equations: A semi-Lagrangian characteristic formulation, Journal of Computational Physics 222 (2) (2007) 849–871.

[74] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, Journal of Computational physics 169 (2) (2001) 556–593.

[75] R. J. LeVeque, Finite volume methods for hyperbolic problems, vol. 31, Cambridge university press, 2002.

[76] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of computational physics 77 (2) (1988) 439–471.

[77] S. Ii, F. Xiao, High order multi-moment constrained finite volume method. Part I: Basic formulation, Journal of Computational Physics 228 (10) (2009) 3669–3707.

[78] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of computational physics 54 (1) (1984) 115–173.

[79] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, in: Upwind and High-Resolution Schemes, Springer, 328–374, 1989.

[80] B. Xie, F. Xiao, A multi-moment constrained finite volume method on arbitrary unstructured grids for incompressible flows, Journal of Computational Physics 327 (2016) 747–778.

[81] B. Xie, Development of numerical model for complex flows on unstructured grids using multi-moment finite volume formulations, Ph.D. thesis, Tokyo Institute of Technology, 2015.

[82] J. S. Park, S.-H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, Journal of Computational Physics 229 (3) (2010) 788–812.

[83] H. Luo, J. D. Baum, R. Löhner, On the computation of multi-material flows using ALE formulation, Journal of Computational Physics 194 (1) (2004) 304–328.

[84] Z. Shen, W. Yan, G. Yuan, A robust and contact resolving Riemann solver on unstructured mesh, Part II, ALE method, Journal of Computational Physics 268 (2014) 456–484.

[85] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced numerical approximation of nonlinear hyperbolic equations, Springer, 325–432, 1998.

[86] J. R. Kamm, F. Timmes, On efficient generation of numerically robust Sedov solutions, Tech. Rep., Technical Report LA-UR-07-2849, Los Alamos National Laboratory, 2007.

[87] F. Xiao, R. Akoh, S. Ii, Unified formulation for compressible and incompressible flows by using multi-integrated moments II: Multi-dimensional version for compressible and incompressible flows, Journal of Computational Physics 213 (1) (2006) 31–56.

[88] J. T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA journal 28 (8) (1990) 1381–1388.

[89] F. J. Blom, Considerations on the spring analogy, International journal for numerical methods in fluids 32 (6) (2000) 647–668.

[90] R. Löhner, C. Yang, Improved ALE mesh velocities for moving bodies, Communications in numerical methods in engineering 12 (10) (1996) 599–608.

[91] R. P. Dwight, Robust mesh deformation using the linear elasticity equations, in: Computational fluid dynamics 2006, Springer, 401–406, 2009.

[92] M. Lombardi, N. Parolini, A. Quarteroni, Radial basis functions for inter-grid interpolation and mesh motion in FSI problems, Comput. Methods Appl. Mech. Engrg. 256 (2013) 117–131.

[93] W. Boscheri, M. Dumbser, Arbitrary-Lagrangian-Eulerian one-step WENO finite volume schemes on unstructured triangular meshes, Communications in Computational Physics 14 (5) (2013) 1174–1206.

[94] P.-H. Maire, R. Loubère, P. Váchal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Communications in Computational Physics 10 (4) (2011) 940–978.

[95] W. Boscheri, M. Dumbser, O. Zanotti, High order cell-centered Lagrangian-type finite volume schemes with time-accurate local time stepping on unstructured triangular meshes, Journal of Computational Physics 291 (2015) 120–150.

[96] J. K. Dukowicz, B. J. Meltz, Vorticity errors in multidimensional Lagrangian codes, Journal of Computational Physics 99 (1) (1992) 115–134.

[97] V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, NRC, Division of Mechanical Engineering, 1962.

[98] J. S. Park, S.-H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, Journal of Computational Physics 229 (3) (2010) 788–812.

[99] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of computational physics 54 (1) (1984) 115–173.

[100] M. Dumbser, I. Peshkov, E. Romenski, O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: viscous heat-conducting fluids and elastic solids, Journal of Computational Physics 314 (2016) 824–862.

[101] J.-R. Carlson, Inflow/outflow boundary conditions with application to FUN3D .

[102] L. Zhang, W. Liu, M. Li, X. He, H. Zhang, A class of DG/FV hybrid schemes for conservation law IV: 2D viscous flows and implicit algorithm for steady cases, Computers & Fluids 97 (2014) 110–125.

[103] B. Müller, High order numerical simulation of aeolian tones, Computers & Fluids 37 (4) (2008) 450–462.

[104] C. Norberg, Fluctuating lift on a circular cylinder: review and new measurements, Journal of Fluids and Structures 17 (1) (2003) 57–96.

[105] A. Placzek, J.-F. Sigrist, A. Hamdouni, Numerical simulation of an oscillating cylinder in a cross-flow at low Reynolds number: Forced and free oscillations, Computers & Fluids 38 (1) (2009) 80–100.

[106] K.-J. Bathe, Finite element method, Wiley encyclopedia of computer science and engineering (2007) 1–12.

[107] K.-J. Bathe, H. Zhang, Y. Yan, The solution of Maxwell's equations in multi-physics, Computers & Structures 132 (2014) 99–112.

[108] K.-J. Bathe, H. Zhang, Finite element developments for general fluid flows with structural interactions, International Journal for Numerical Methods in Engineering 60 (1) (2004) 213–232.

[109] K.-J. Bathe, G. A. Ledezma, Benchmark problems for incompressible fluid flows with structural interactions, Computers & structures 85 (11-14) (2007) 628–644.

[110] K.-J. Bathe, H. Zhang, A mesh adaptivity procedure for CFD and fluid-structure interactions, Computers & Structures 87 (11-12) (2009) 604–617.

[111] B. Xie, P. Jin, F. Xiao, An unstructured-grid numerical model for interfacial multiphase fluids based on multi-moment finite volume formulation and THINC method, Int. J. Multiph. Flow 89 (2017) 375–398.

[112] R. Rossi, Direct numerical simulation of scalar transport using unstructured finite-volume schemes, J. Comput. Phys. 228 (5) (2009) 1639–1657.

[113] C. Farhat, M. Lesoinne, Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems, Comput. Methods Appl. Mech. Engrg. 182 (3-4) (2000) 499–515.

[114] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476.

[115] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, Comput. Methods Appl. Mech. Engrg. 194 (42-44) (2005) 4506–4527.

[116] U. Küttler, W. A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Comput. Mech. 43 (1) (2008) 61–72.

[117] B. Irons, A version of the Aitken accelerator for computer iteration., Internat. J. Numer. Methods Engrg. 1 (3) (1969) 275–277.

[118] B. Xie, X. Deng, S. Liao, High-fidelity solver on polyhedral unstructured grids for low-Mach number compressible viscous flow, submitted to Comput. Methods Appl. Mech. Engrg. .

[119] U. Ghia, K. N. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, Journal of computational physics 48 (3) (1982) 387–411.

[120] X. Su, Y. Cao, Y. Zhao, An unstructured mesh arbitrary Lagrangian-Eulerian unsteady incompressible flow solver and its application to insect flight aerodynamics, Phys. Fluids 28 (6) (2016) 061901.

[121] C.-C. Liao, Y.-W. Chang, C.-A. Lin, J. McDonough, Simulating flows with moving rigid boundary using immersed-boundary method, Computers & Fluids 39 (1) (2010) 152–167.

[122] J.-I. Choi, R. C. Oberoi, J. R. Edwards, J. A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2) (2007) 757–784.

[123] J. Yang, F. Stern, A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions, J. Comput. Phys. 231 (15) (2012) 5029–5061.

[124] C. Liu, C. Hu, Block-based adaptive mesh refinement for fluid structure interactions in incompressible flows, Comput Phys Commun 232 (2018) 104–123.

[125] C. Williamson, R. Govardhan, Vortex-induced vibrations, Annu. Rev. Fluid Mech. 36 (2004) 413–455.

[126] S. Behara, I. Borazjani, F. Sotiropoulos, Vortex-induced vibrations of an elastically mounted sphere with three degrees of freedom at Re= 300: hysteresis and vortex shedding modes, J. Fluid Mech. 686 (2011) 426–450.

[127] A. G. Tomboulides, Direct and large-eddy simulation of wake flows: flow past a sphere, Previews of Heat and Mass Transfer 6 (21) (1995) 563–564.

[128] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, J. Fluid Mech. 378 (1999) 19–70.

[129] F. W. Roos, W. W. Willmarth, Some experimental results on sphere and disk drag, AIAA J. 9 (2) (1971) 285–291.

[130] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.

[131] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, International Journal for Numerical Methods in Fluids 48 (9) (2005) 1023–1040.

[132] B. Xie, F. Xiao, Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids: The THINC method with quadratic surface representation and Gaussian quadrature, Journal of Computational Physics 349 (2017) 415–440.

[133] L. Qian, Y. Wei, F. Xiao, Coupled THINC and level set method: A conservative interface capturing scheme with high-order surface representations, Journal of Computational Physics 373 (2018) 284–303.

[134] R. G. Dean, R. A. Dalrymple, Water wave mechanics for engineers and scientists, vol. 2, World Scientific Publishing Company, 1991.

[135] D. G. Goring, Tsunamis–the propagation of long waves onto a shelf .

[136] P. Lin, A fixed-grid model for simulation of a moving body in free surface flows, Computers & fluids 36 (3) (2007) 549–561.

[137] C. Mnasri, Z. Hafsia, M. Omri, K. Maalel, A moving grid model for simulation of free surface behavior induced by horizontal cylinders exit and entry, Engineering Applications of Computational Fluid Mechanics 4 (2) (2010) 260–275.