32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

An Intelligent Structured Chimera Grid Approach for Moving Objectives

LI Li[1, 2], LIANG Yihua[2], HUAI Yang[2]

[1]School of Computer Science, Northwestern Polytechnical University
[2]AVIC Aeronautics Computing Technique Research Institute

Abstract

There are many demands to establish the capability to simulate problems with moving objectives, e.g., safety evaluation of multi-bodies separation, maneuver flight simulation, landing process simulation, and so on. The chimera grid approach is an efficient and flexible way to solve these problems. Based on authors' previous work on the static automatic chimera grid approach, this paper presents an intelligent structured chimera grid approach for problems with moving objectives. The idea is to introduce a thought on the modular and application programming interface (API) based software development strategy into the procedure to develop a scalable and reusable software library, SOGA, for chimera grid applications. The approach is implemented into the CFL3D solver (which is an open source CFD solver from NASA, and doesn't have chimera capability for moving objectives) to demonstrated efficiency of the procedure, and results from a pitching NACA0012 airfoil (which is known as the AGARD TC3 case at Mach 0.6, angle of attack 2.89), and flow past two moving cylinders at Mach 0.2, and Reynolds number 1.e+5 are shown to validate the algorithm. It is concluded that the present approach is feasible and effective.

Keywords: Automatic chimera grid; Moving objectives; ADT trees; Validation; Computational Fluid Dynamics

1. Introduction

In the last decades, Computational Fluid Dynamics (CFD) has undergone a strong development and has become a powerful tool both for the analysis and understanding of fluid dynamics phenomena, and for the design and optimization of aerodynamic performance of aircrafts or aerospace vehicles. This progress has been made possible with the advent, in the meantime, of faster and faster supercomputers with increasing memory capabilities, and with the rapid progress of modern numerical computing technology. CFD technology is becoming one of the basic and the keynote enabling technologies to support the future 'digital aeronautics' via virtual design and virtual manufacture [1].

However, during the use of the CFD technology, grid generation is always a bottle-neck problem. There is a famous statement that one has to put more than 70 percent work to prepare the computational grid to perform a simulation for a real and complex aircraft configuration. With remarkable advantages to economic the procedure to establish a complex grid system for a real and complex problem, the chimera grid approach had been paid much attention during the past years since it was proposed in 1980s [2]. The main idea of the chimera grid approach is to divide the complex configure or the complex computational regions into simple sub ones according to the geometry complexity or the flow complexity. In this way, one can easily generate the computational mesh for each sub region, and further establish the grid system used for the whole problem. Such a strategy is particularly suitable for modeling the CFD problem with moving objectives [3,4]. The big reason is that the chimera grid approach can greatly simplify the procedure to generate the dynamical mesh for modeling the problem. When it is used, it only needs to re-assemble the meshes in sub region to establish a new chimera gridding system, but not to generate the dynamical computational mesh repeatedly.

Nevertheless, there are still many specific problems when one wants to establish an engineering-oriented capability for easy use of the chimera grid technology for large and complex applications. In the past few years, the authors had been devoting to establish such a capability by developing an automatic and intelligent chimera grid approach [4,5] within in-house code. It is a similar work as NASA to develop a general software package, PEGSUS, for the chimera grid applications [6]. There are many demands to establish such capabilities since there are more and more practical needs for virtual design and testing by simulating the problems with many moving objectives such as safety evaluation of multi-bodies separation, maneuver flight simulation, landing process simulation, and so on. For such problems, the automatic chimera grid approach can provide an elegant and efficient way.

In this work, based on authors' previous work for static problem [4,5], we present an intelligent chimera grid approach for multi-block structured grid to solve the problems with moving objectives. The main idea is to introduce a thought on the modular and application programming interface (API) based software development strategy into the procedure to develop a scalable and reusable software library, SOGA, for chimera grid applications. The full name of SOGA is read as Structured based Overset Gridding Assembler, which consists of several APIs with different functions that include definition of basic properties, grid registration, preprocessing for hole-mapping and alternating digital tree (ADT) construction, connectivity establishing, and I/O management. The feasibility and efficiency of the procedure is demonstrated by coupling the SOGA package with a well-known open source CFD solver, CFL3D, which doesn't have chimera grid capability for moving objectives. Numerical results from a pitching NACA0012 airfoil, which is known as the AGARD TC3 case at Mach 0.6, angle of attack 2.89, and flow past two moving cylinders at Mach 0.2, and Reynolds number 1.e+5 are shown further to validate the algorithm.

The full paper is organized as followed. In first Section, strategies used in this paper for generating the chimera grid for large complex engineering problems is outlined. Subsequently, SOGA APIs and its key algorithms to establish a real chimera gridding system for computation is briefly introduced followed by the implementation into the CFL3D code. Finally, in last Section, typical numerical results are shown to validate the procedure and the algorithm.

2. Grid Generation Strategies

In our approach, the fundamental strategy to generate the chimera grid is to divide the whole grid generation procedures into the near-body grid generation, and the background grid generation similar to PEGSUS [6]. But unlike the strategy used in PEGSUS that one has to generate a single block mesh for the near-body or the background grid, we have adopted a flexible way. For each type of grid, near-body or background, one can generate a single block mesh or a multi-block mesh in order to ensure the feasibility and the practicability of the approach. After that, all grids can be assembled as a complex overlapping mesh system, i.e., the chimera gridding system for computation. In this way, one can easily keep all best practices in generating a practical grid for complex application, and choose different grid generation software, e.g., the commercial ICEM-CFD software, Pointwise software, or some in-house grid generation tools as requirements.

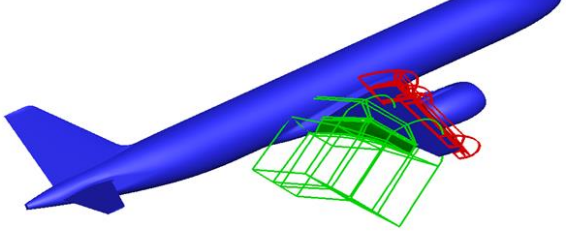


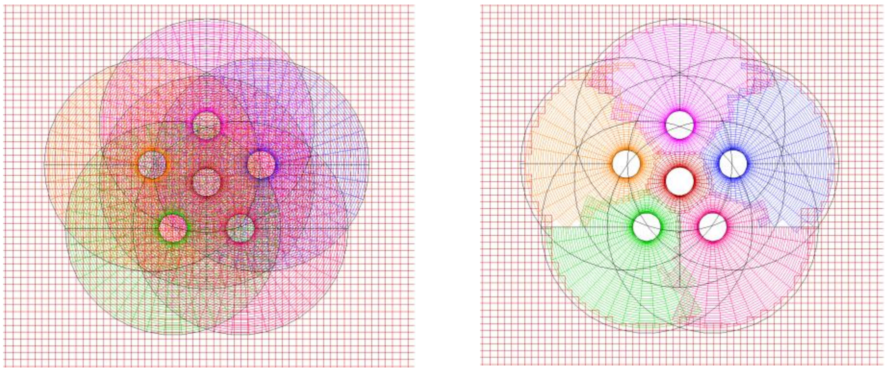
Figure 1 – Topology to generate the chimera grid for a landing configure of real civil aircraft

To illustrate the flexibility of the presented approach, in Figure 1, a topology to generate the chimera grid for a landing configure of real civil aircraft is shown. In this example, the background grid is defined as the gird for the whole wing-body-nacelle-tail configure of the aircraft except high lift devices, and the near-body grid consists of two grids, in which one is for the slat, and another is for the flat. For an experienced engineer, one can easily generate a multi-block mesh that is very suitable for the practical complex application both for the background grid, and the near-body grid.

3. SOGA APIs and Keynote Algorithms

In order to establish a general software library that can be coupled with arbitrary structured based CFD solvers, in particularly, our in-house code, we have introduced a thought to develop some necessary application programming interfaces (APIs) with modular development method. The main objective to develop such a library or software package is to hide behind the algorithm and the programming details for establishing relationships between the field points and the interpolated points needed by the chimera grid computation, and to pride a standard interfaces called by other programs.

It's known that in a chimera grid approach, both the field points and the interpolated points are those should be computed. The difference between them is that values on the interpolated points are obtained by a suitable interpolating algorithm, e.g., the trilinear interpolation, according to values on surrounding field points. The main task herein is that we firstly have to identify all points in the whole overlapping mesh system, at which points the flow solution should be computed (the field points), interpolated (the receptor points), and not computed at all (the hole points). The procedure to identify these different points can be graphically called the procedure for 'digging holes', or be called hole profiling [7]. In Figure 2, an example for 'digging holes' for an overlapping grid system consisting of five cylinder-grids and a background grid is shown to further illustrate the above terminologies, in which the result is obtained via SOGA.



(a) Overlapping grids: all points (b) Field points (solution computed)

Figure 2 – Example for hole profiling for an overlapping grid system

Besides hole profiling, in order to establish a complete procedure for a chimera grid computation, one should also have to do some other operations including query point identification, pre-processed mesh-Block profiling, donor search, decision logic, and interpolation as described and illustrated in the literature [7].

Kept these procedures in mind, now we can try to establish proper APIs suitable for adapting arbitrary CFD solvers. In our approach we have defined five types of different APIs to implement these functions. These APIs include:

1) API for register basic properties and for dynamic memory management:

SUBROUTINE SOGA_register_basic(...)

SUBROUTINE SOGA_delete()

2) APIs for registering volume grids and boundaries:

SUBROUTINE SOGA_register_grid_data(...)

SUBROUTINE SOGA_register_bc_global(...)

SUBROUTINE SOGA_register_block_bc(...)

3) APIs for hole profiling, and query point identification:

SUBROUTINE SOGA_preprocess_grids()

4) APIs for donor search and finding interpolation coefficients:

SUBROUTINE SOGA_perform_connectivity()

5) APIs for I/O, in particularly for output of the overlapping and interpolating information:

SUBROUTINE SOGA_printout_files()

After that, we can use the above APIs to establish a typical process to obtain the complete overlapping and interpolating information for a chimera grid application, which can be clearly illustrated in Figure 3. There are five key steps: a) registering basic and global information into SOGA for allocating dynamic memory; b) registering grid and boundary condition data for real application; c) preprocessing for hole profiling, and hole optimization; d) preprocessing for donor search, and for interpolating; e) releasing allocated dynamic memory.

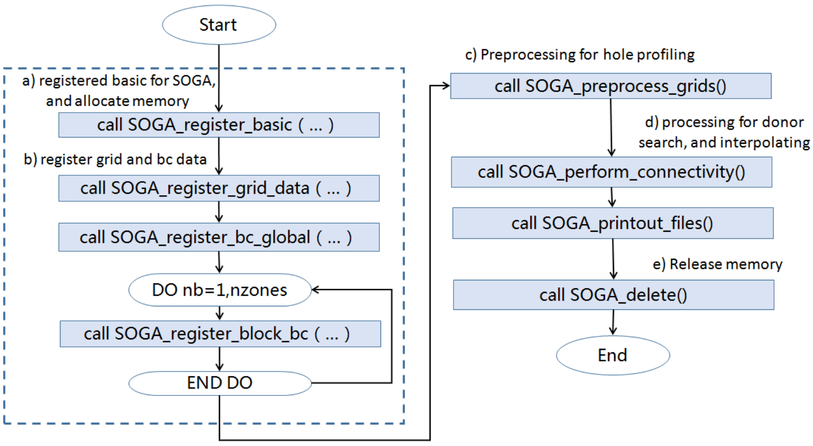


Figure 3 – A typical process for calling SOGA APIs for the chimera grid application

Through above APIs, it can be found that one had hide all the key operations and algorithms for preprocessing of chimera grid behind so that it can be easily to implement different algorithms in the same framework. In our approach, we have adopted a modified hole-mapping method through inverse marked method and vector ray method for hole profiling [4], and an Alternating Digital Tree (ADT) based method for grid assembly [8]. The advantage of the present method is that it is valid for complex and half-closed geometry, while the traditional method not. For details one can refer to [4] and [9].

To demonstrate the feasibility and efficiency of present approach, we had implemented the above procedure into the well-known open source CFD solver, CFL3D [10]. CFL3D is a Reynolds-Averaged Navier-Stokes flow solver for structured grids, whose original version was developed in the early1980's in the Computational Fluids Laboratory at NASA Langley Research Center, and had been used widely for different applications. The origin CFL3D solver has the solution capability for static chimera and overlapping grid, but doesn't have dynamic chimera grid capability required for moving objectives. Even for the static chimera grid cases, CFL3D has to introduce other tools outside, e.g., MAGGIE, PEGSUS [6], to establish interpolating relationships between different overlapping grids.

The key points to couple the SOGA library into CFL3D include: a) one has to introduce a new data structure for recording global overset data (namely, global_overset_m, in our approach); b) one has to perform the whole operations to find interpolating relationships between grids, if some of grids in computations are updated; c) one has to update the overlapping information via external files or memory repeatedly by time.

4. Numerical Examples and Results

In order to validate the approach presented in this paper, numerical results for several typical benchmark test cases have been obtained. These test cases include the ARA-M100 wing-body configuration, the AGRAD TC3 pitching NACA0012 airfoil case and the two moving cylinders case. The former ARA-M100 case is used to validate the algorithm that would be used for complex configuration in static case.

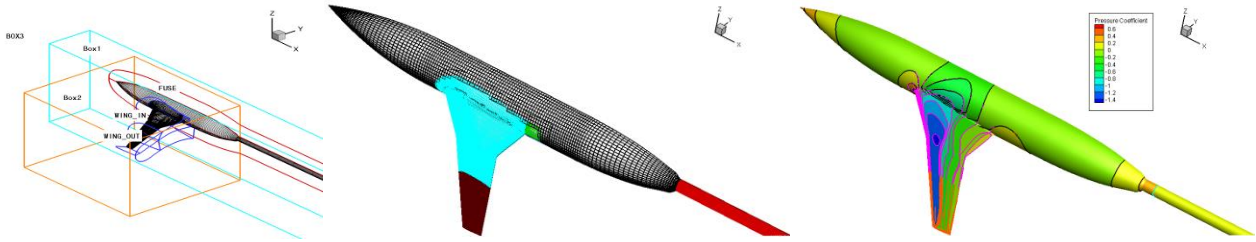
4.1 ARA-M100 Wing-body Configuration

ARA-M100 wing-body configuration is a benchmark test case provided on CFL3D website [10]. Results obtained via CFL3D and PEGSUS 4.1 are also provided by the website so that we can utilize it to perform comparison between PEGSUS and SOGA libraries. The flow condition for the case is at:

$$M_{\infty} = 0.8027\,, \alpha = 2.873^{0}\,, \mathrm{Re} = 1.31 \times 10^{7}$$

The chimera grid utilized in this case can be also found on the website, which consists of six different components (see Figure 4(a)). These component grids include Box1, Box2, Box3, WING_IN, WING_OUT, and FUSE.

In Figure 4(b)-Figure 4(c), typical results calculated by our approach for the wing-body configuration are shown. It can be seen that the surface pressure coefficients are very smooth at different interpolating interfaces to ensure to obtain a good result. In Table 1, comparison of total forces between CFL3D+SOGA and CFL3D+PEGSUS (version 4.1) is drawn. Good agreement between the different approaches is obtained, which validates our approach. Compared with PEGSUS, our approach is an unmanned and automatic one, where people don't need to prepare a complex input file to describe overlapping relationship between grids for the preprocessor.



(a)Topology for the chimera gridding system

(b) Surface grids: field points

(c) Caculated surface pressure coefficients

Figure 4 – Chimera grid sytem and typical results for the ARA-M100 wing-body configuration

Table 1 – comparison of total forces between SOGA and PEGSUS

methods	Cl	Cd	Cm
CFL3D+SOGA	0.6469	0.0465	-0.1724
CFL3D+PEGSUS[10]	0.6586	0.0468	-0.1698
Difference	-1.77%	-0.64%	1.53%

4.2 AGARD TC3 Pitching NACA0012 Airfoil

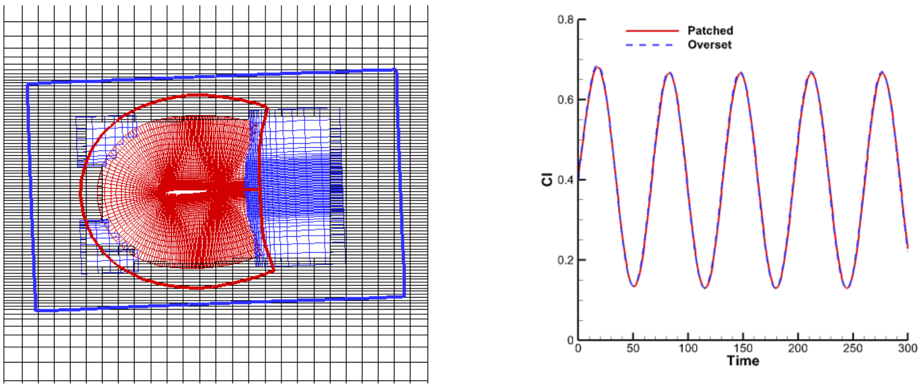
The AGARD pitching NACA0012 airfoil had been used widely worldwide as a validation case for unsteady and moving grid applications. In origin AGARD wind-tunnel experiments, there are three different cases, namely, TC1, TC2, and TC3. In our numerical experiments, we have chosen the TC3 case as the test case. The flow condition of TC3 is at

$$M_{\infty} = 0.6\,, \alpha_0 = 4.86^{0}\,, \mathrm{Re} = 4800000$$

The pitching condition is

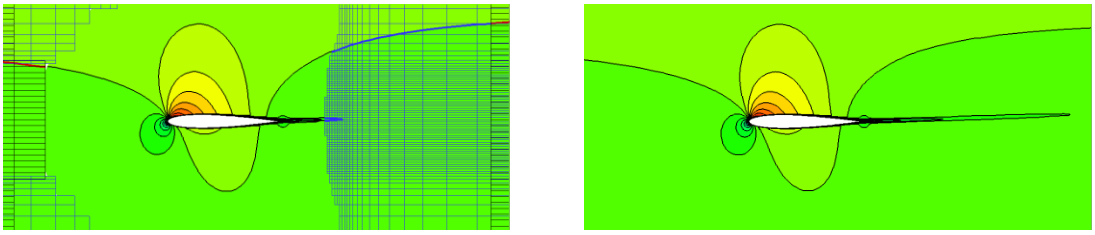
$$\alpha = \alpha_0 + \alpha_1 \sin(2\pi f t)$$

where $\alpha_1 = 2.44^{o}$, $f = 50.32Hz$. In Figure 5, the chimera gridding system and the predicted transient lift coefficient is shown, in which the predicted lift coefficient is compared with those obtained via a traditional patched grid approach. It can be shown that these two different approaches had given almost identical lift results, which confirms the feasibility and efficiency of the present dynamic chimera grid approach. In Figure 6, comparison of the typical transient pressure contour with different approaches for the pitching NACA0012 airfoil is further drawn. Again, almost identical results from different approaches are obtained.



(a) Chimera gridding system (b) Predicting lift coefficient

Figure 5 – Typical result for the pitching NACA0012 airfoil at Mach 0.8



(a) Chimera grid based approach (b) Patched grid based approach

Figure 6 – Comparison of typical transient pressure contour with different approaches for NACA0012 airfoil

4.3 Two moving cylinders

Flow past two moving cylinders at flow condition, is considered here. This test case is used to validate the robustness of the present approach. In the case, the initial distance between the two cylinders is 4 times of the radius for each cylinder, then one cylinder starts to move at a constant speed at 34.0m/s. In Figure 7, typical total pressure ratio contour results at different time are shown. It can be seen that reasonable results have been obtained using our approach. From the result, we can also conclude that our approach is quite robust since it can deal with the very closed, even collapsed geometry cases from the numerical experiment.

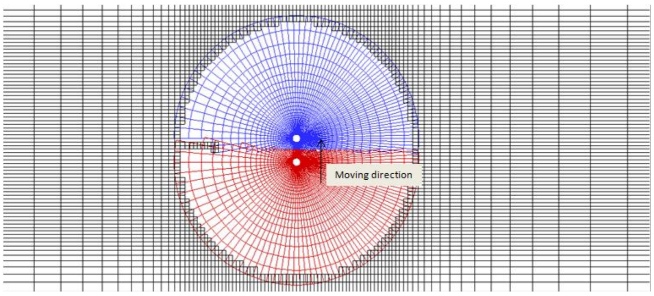


Figure 7 – The initial Chimera gridding system for the two moving cylinders case

5. Conclusions

In order to establish the capability of chimera grid approach for moving objectives, we present an intelligent approach by introducing the modular and application programming interface (API) based software development strategy into the procedure to develop a scalable and reusable software library for chimera grid applications. This results in developing a so-called SOGA library, which provides functions for definition of basic properties, grid registration, preprocessing for hole-mapping and alternating digital tree (ADT) construction, connectivity establishing, and I/O management to establish a complete procedure for static and dynamic chimera grid application. The library is coupled with CFL3D, a well-known open source and wide-used code, to validate the procedure. Numerical results on three typical test cases including the ARA-M100 wing-body configuration, the AGRAD

TC3 pitching NACA0012 airfoil case and the two moving cylinders case validate the efficiency and robustness of the present approach.

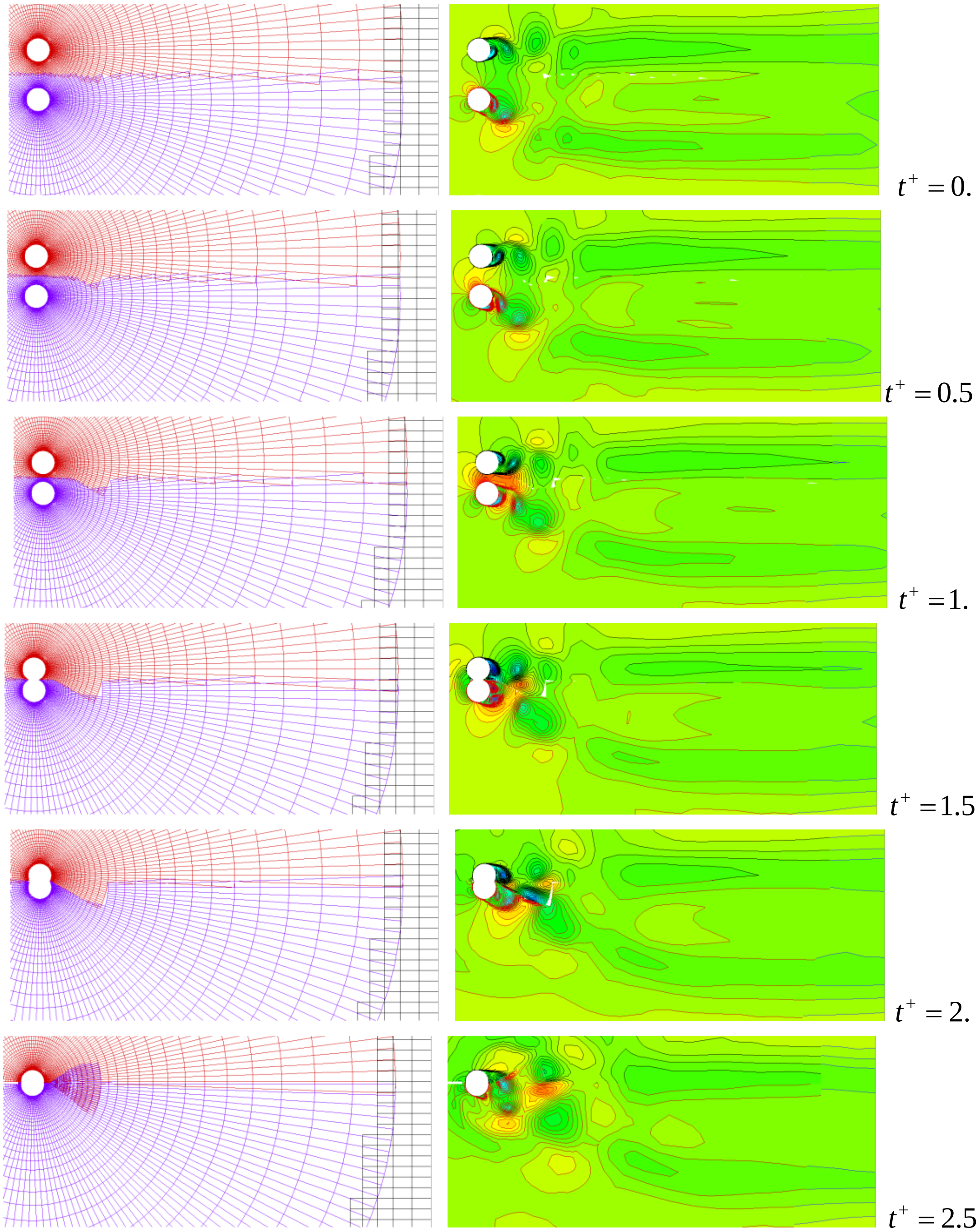


Figure 8 – Typical dynamic chimera gridding system and corresponding total pressure ratio contour for the moving cylinders case at Mach 0.2, Re=1.e+5

6. Contact Author Email Address

mailto: westlili@163.com

7. Copyright Statement

References

[1] CHEN Yingchun, ZHANG Meihong, ZHANG Miao, et al. Review of Large Civil Aircraft Aerodynamic Design. Acta Aeronautica Aeronautic et Astronautic Sinica, 2019, 40(1): 522759 (in Chinese).

[2] Benek J A，Buning P G，Steger J L．A 3-D chimera grid embedding technique. Proceedings of the 7th AIAA Computational Fluid Dynamics Conference, Cincinnati, OH：AIAA, 1985．

[3] Du Chao, Li Xiaowei. Numerical analysis of viscous flow around oscillatoring wing using dynamic chimera grid method. Journal of Shanghai University: Natural Science Edition, 2007, 13(3): 304-307 (in Chinese)).

[4] Huai Yang, Zhang Yifan, Hao Haibing, et al. Numerical simulation on multi-body separation problems based on structured chimera grid. Proceedings of the 2015 Workshop on Dynamic Aerodynamics in Aeronautics. Shenyang: Shenyang Aircraft Design Research Institute, 2015: 309-317 (in Chinese)).

[5] LI Li, MA Rong, HAO Haibing, etc. A practical method for efficient creation of initial hole boundary in chimera grid computation. Chinese Journal of Applied Mechanics, 2016, 33(2): 262-267(in Chinese)).

[6] Rogers S E, Suhs E, Dietz W E. PEGASUS 5: an automated preprocessor for computational fluid dynamics. AIAA Journal, 2003, 41(6): 1037-1045.

[7] Roget_B and Sitaraman J. Robust and Scalable Overset Grid Assembly for Partitioned Unstructured Meshes. AIAA 2013-0797, 2013.

[8] Bonet J and Peraire J. An Alternating Digital Tree (ADT) Algorithm for 3D geometric searching and intersection problems. International Journal of Numerical Methods in Engineering,1991, 31: 1-17.

[9] HUAI Yang. Investigations on automatic overlapping grid technologies. Thesis for Master degree in Science, Chinese Aeronautics Establishment, 2015(in Chinese)).

[10]CFL3D website: http://cfl3d.larc.nasa.gov/Cfl3dv6/ cfl3dv6.html.