Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona, Spain, July 9-13, 2018

ICCFD10-052

Parallel Implicit Hole-cutting Method for Unstructured Chimera Grid

X. H. Chang*,**, R. Ma**, N. H. Wang** and L. P. Zhang*,**
Corresponding authors: cxh_cardc@126.com; zhanglp_cardc@126.com

* State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang Sichuan 621000, China.
** Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang Sichuan 621000, China.

Abstract: As an efficient simulation method for complex configurations and moving boundary problems in computational fluid dynamics, Chimera or overset grid techniques have been widely used in many aspects of aeronautics and astronautics. However, there are still some bottlenecks for present hole-cutting method to handle large scale mesh, such as memory storage limitation, hole-cutting efficiency. In this paper a novel parallel implicit hole-cutting method is presented for handling large scale overset mesh. The method is based on parallel searching of donor cells for all grid nodes. To reduce memory consumption for the searching procedure, the global-to-local searching strategy as well as the background grid approach has been developed. To guarantee the simple connectivity of overset domains, a parallel front advancing method is adopted to automatically distinguish the active regions. Finally, the efficiency is analyzed through some test cases, which demonstrated the efficiency and effectiveness of present Chimera grid method.

Key Word: Implicit hole-cutting, Unstructured chimera grid, Unsteady flow, Large scale mesh.

1 Introduction

Chimera or overset grid method [1] was originally developed to generate structured mesh for complex configurations and has been improved to handle multi-bodies with relative movement in an unsteady condition. At present, Chimera grid method has been widely used in many aspects of aeronautics and astronautics such as the multi-body separation, the rotor wing simulation and so on [2].

With the rapid development of computer hardware and computing method, large-scale computing has become one of the most important research directions in computational fluid dynamics (CFD). Large-scale high performance computing (HPC) is of great significance for both the fundamental research and engineering applications in fluid dynamics. For example, the CFD simulations of future aircrafts need large scale mesh (billions or more cells) to capture more detailed flow structures and more accurate aerodynamic coefficients, especially for the cases with turbulent flow separation. However, handling large scale overlapping grids is still difficult for present hole-cutting method. There are three main issues: (1) Memory consumption limitation. Searching of donor cells is the key process of hole-cutting, and the necessary grid information such as topology, geometry would occupy massive memory storage, particularly for unstructured grids. Therefore, efficient parallel hole-cutting method should be set up to reduce the memory consumption in a distributed parallel environment. (2) Assembling efficiency. It is hard to improve the parallel efficiency for parallel hole-cutting method because of the massive logical operations and message communications between processors. (3) The automation and robustness of hole-cutting methods is also a key issue for large scale overset grid, since the debugging and modification by manual operation would become impossible if very large scale grid is used.

There are two types of hole-cutting methods in literature: the explicit method and the implicit one. In the explicit method, the first step is to blank the cells which locate inside the solid bodies and to designate the interpolated cells. After that, the donor cells for all the interpolated cells should be found to set up the interpolation relationship. Various blanking methods [3]-[6] for cells inside the body have been proposed, and some optimization strategies [7], [8] were developed also to improve the quality of the hole fringe. All of these methods work well, but they have some restrictions such as requiring moderate to high degree of user input. While the implicit hole-cutting (IHC) [9] method is actually a cell-selecting process and no stand-alone hole-cutting takes place. The method loops through all the cells (or nodes) in the grid system to test and select the proper ones, where the cell quality (such as the size or the distance to wall) is usually used as the criterion. The IHC method is highly automatic and robust, but it's not efficient because of the massive searching jobs.

Actually the hole-cutting method mentioned above is suitable for both structured and unstructured mesh. In unstructured overset mesh, fewer grid blocks are used usually to discretize the flow domain. Nakahashi [10] proposed an IHC method based on the minimum distance to wall for unstructured overset mesh. After that, Togashi [11] promoted the method to solve multi-body relative movement problems. Furthermore, in the works of Lohnor [12] and Luo [13], a combined parameter from both the cell-size and the minimum distance to wall was used to improve the results of overset grid assembling.

Parallel computing is the only way to deal with large-scale overset mesh. Zhang et al. [14] developed a parallel explicit hold-cutting method for unstructured mesh. In their method, the cells inside the solid body are sorted by checking if their edges cross the solid boundary faces. Sitaraman et al. [15] developed an automatic and parallel implicit hole-cutting software PUNDIT which was further enhanced by Roget et al. [16]. Now, PUNDIT has become an important module of the Helios system of CREATE-AV [17] in America. TIOGA [18] is another parallel hole-cutting software which is suitable for high-order schemes. Besides, Zagaris et al. [19], Landmann et al. [20] , Liang et al. [21] also developed distinctive parallel hole-cutting methods.

Because of the variety of hole-cutting methods, the corresponding parallel strategies are also different from each other. For the purpose of efficiency, robustness and automation, further research is still required. As mentioned above, the IHC method is of great automation and robustness, but the searching jobs would consume huge CPU time and occupy massive memory storage, so an effective and efficient parallel method is strongly desired.

In this paper, an automatic parallel implicit hole-cutting technique for unstructured overset mesh is presented. The minimum distance to wall is used as the criterion to blank the cells in overlapping regions. A front advancing method is developed to identify all the active cells. An advantage of the front advancing method is that it can guarantee the simple connectivity of active regions, and the cells inside solid bodies can be automatically blanked. The parallel strategy is based on partitioned grid zones. The 'global-to-local' (GTL) method as well as background grid method is proposed to reduce the memory consumption and improve the efficiency. Test cases demonstrate the effectiveness of present parallel method.

2 Parallel IHC method

For unstructured overset mesh, the grid system is composed with several grid blocks around solid bodies. Then each grid block is partitioned into sub-blocks (zones) for parallel computing. In this paper, the cell-centered finite volume discretization on unstructured mesh is adopted, so the purpose of present hole-cutting method is to identify the cell type and to set up the relationship of interpolated/donor cells between grid zones.

As illustrated in Figure 1, the procedure of present method can mainly be divided into three steps: The first step contains the preparatory work such as building the alternating digital tree (ADT) data structure, calculating the minimum distance to wall and constructing some necessary geometric and topology information. This part of job can be easily parallelized, and high parallel efficiency can be achieved because the load balance can be ensured by proper grid partitioning method for unstructured mesh, such as METIS. So the details will not be discussed later.

The second step is to classify the nodes and cells into active and inactive types, in which the traditional IHC method corresponding with a front advancing method is used. The final step is to identify the interpolated cells and to search their donor cells in other grid zones. In the following context, these two steps will be discussed in details.

2.1 Nodes classification in overlap region

The traditional IHC method based on minimum distance to wall is adopted to classify the nodes in the overlapping regions. This method loops through all the nodes to test their attributes. To be specific, Donor cell needs to be found firstly for each node. After that, the node type is identified by comparing between its checking parameter and that of the donor cell. Here, the minimum distance to wall is chosen as the checking parameter. The distance of each node or cell to solid wall is calculated in the block itself. For blocks without solid wall, a user-specified parameter should be given.

For the parallel method here based on partitioned grid zones, the simplest way to find the donor cell is to search in the global mesh. However, this strategy needs massive memory storage because the information of the entire grid system should be stored in each processor.

In previous work of the authors, a 'global-to-local' (GTL) searching strategy [22] described in Figure 2 has been developed to reduce memory consumption. In this GTL strategy, the 'local' node coordinates in each processor are gathered up firstly. Then, the donor cells of the global nodes are searched in the local partitioned grid zones which belong to the present processor. Finally, the donor cell with the smallest distance to wall is chosen amount the results on these processors. This GTL approach needs relatively less storage because only the node coordinates of global mesh need to be saved. It has been successfully used to handle unstructured overset mesh with 123 million cells in our previous work [22]. However, there are still some disadvantages for the GTL strategy. Firstly, it may become invalid when handling larger scale mesh (such as mesh with several billions of cells). Besides, it is not suitable for massive parallel processing because the massive communication between processors is needed to gather up the node coordinates on local zones.

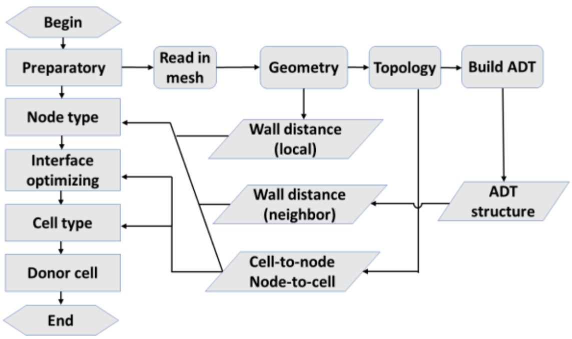


Figure 1: Procedure of the implicit hole-cutting method.

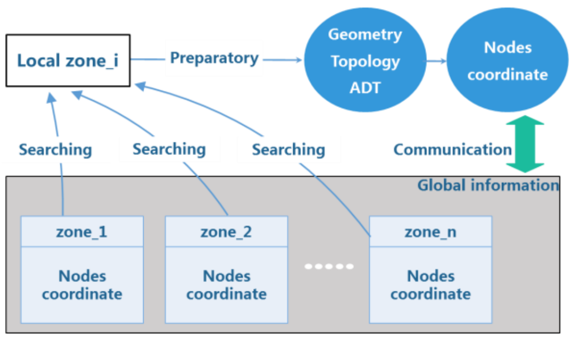


Figure 2: The global-to-local searching strategy.

In this work, a background mesh approach is developed to solve the problem of memory consumption for large scale overset mesh. In this approach the type of nodes is identified by donor cell searching and interpolating in a very coarse background mesh, which can be generated automatically in the preparatory work. The background mesh can only contain triangular cells for 2D cases or tetrahedral cells for 3D cases. Considering the NLR7301 airfoil, for example, Figure 3 shows the computational mesh system before hole-cutting. There are two grid blocks whose total number of cells is about 32,000. Figure 4 shows the corresponding background grids for the two overlapped grid blocks. The total number of background grids is about 300 only.

The ADT-based searching algorithm is used in this work. If the donor cell has been found, the checking parameter (the minimum distance to wall) of the donor cell needs to be interpolated to the nodes from the background grids. Here the interpolating method based on Taylor expansion is used. The variable gradient in a cell can be calculated by the integral on its surfaces with the well-known Green-Gauss formulae:

$$\frac{\partial q}{\partial x}=\frac{1}{V}\sum_{\mathrm{j=1}}^{nf}q_j S_j n_{jx} \qquad \frac{\partial q}{\partial y}=\frac{1}{V}\sum_{\mathrm{j=1}}^{nf}q_j S_j n_{jy} \tag{1}$$

where q denotes the minimum distance to wall. V is the volume of the target cell. nf is the number of surrounding faces of present cell. S is the area of a face, n is the outward normal of a face. The value on a chosen node (red point in Fig.5) then can be calculated by the vector from the node to the cell center with Taylor expansion. Figure 5 shows the sketch of present interpolation method. It is highly recommended that the vertex-centered approach is adopted for the gradient calculation in order to improve the robustness of hole-cutting. In the vertex-centered approach, the value of a node (black point in Fig.5) is calculated by the average of all its neighbor cells. Because of the extended grid stencil in this vertex-centered approach, the gradient in a cell can be calculated more accurately.

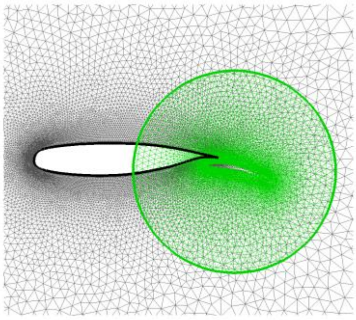
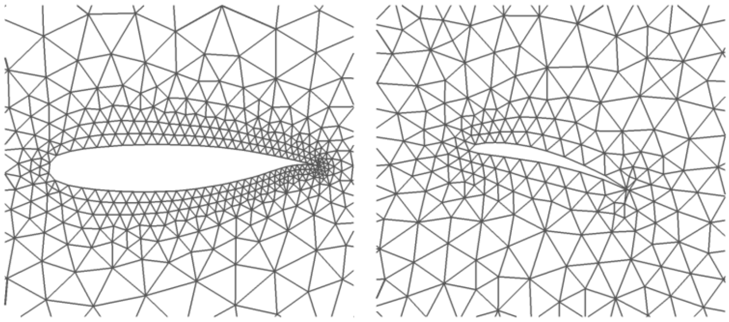


Figure 3: Overset mesh for the NLR7301 airfoil (before hole-cutting)



(a) main foil (b) flap

Figure 4: Background mesh for the NLR7301 airfoil

The determination of the type of a node is indicated in Figure 6, where distance_1 denotes the minimum distance to the solid wall in the present block, and distance_2 is the value calculated by background donor cell in the other block. If distance_1 is larger than distance_2, the node is set to be inactive. In multiple overlapping regions, there may be more than one donor cells for a node. Then the one with the smallest distance is chosen.

By the process above the nodes in the overlapping regions can be classified into active or inactive type. These inactive nodes in overset region form the boundary of active regions.

2.2 Front advancing method

By the searching and interpolating steps above, the types of nodes that do not lie in overlapping regions haven't been identified yet. A feasible way to identify these nodes is to check whether they locate inside any solid body. The nodes inside solid bodies must be inactive, otherwise it must be active.

In this paper, an alternative method, front advancing approach, is proposed. The nodes on physical boundaries (e.g. no-slip wall surface) or other conventional boundaries (e.g. symmetry plane, inflow/outflow) must be active for each grid zone, so these boundaries nodes can be used as the initial front of the active region. By the topology between cells and nodes for unstructured mesh, the front is advancing forward till the inactive nodes which have been identified. Figure 7 shows the sketch of the front advancing method for the flap of the NLR7301 airfoil, in which only the wall surface is used as the initial front. The outer boundary was defined in mesh generation, and the initial boundary of active region was determined in the previous step. This front advancing approach can guarantee the simple connectivity of active regions. And it is also suitable for multi-block problems.

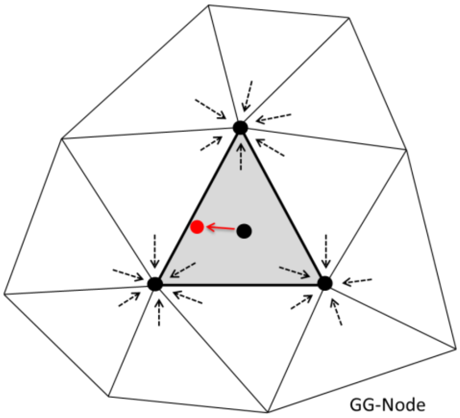


Figure 5: Interpolation of the variable on a node by the cell gradient.

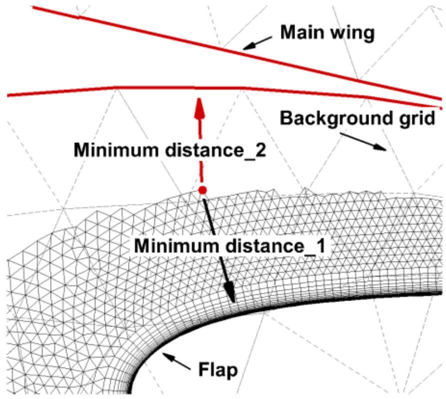


Figure 6: Determination of the type of a node by its minimum distance to wall.

Furthermore, this front advancing method has been parallelized. For each grid zone, the character of nodes on the interface should be communicated to its neighboring zones after each advancing cycle. Two communication approaches can be used here. The first one is to utilize the cell-to-cell connection relationship on the interfaces between grid zones, which has been built up for the cell-centered finite volume scheme. Figure 8 shows the sketch map of this method. The type of cells on the interface which has been determined by their surrounding nodes is communicated firstly. After that, each face on interface boundary is checked. If its ghost cell is active, all the nodes on the face are set to be active. Another method is to use the mapping information between local nodes and global nodes, which has been set up during grid zone partitioning. Using the mapping information, the node type can be determined by uploading the node attribute from local grid to global grid.

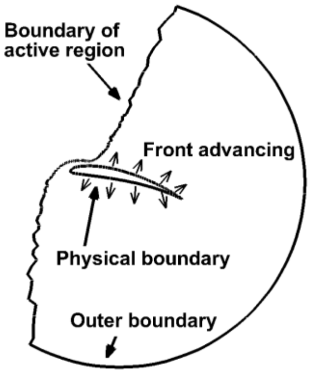


Figure 7: Sketch map for the front advancing method (the flap of the NLR7301 airfoil)

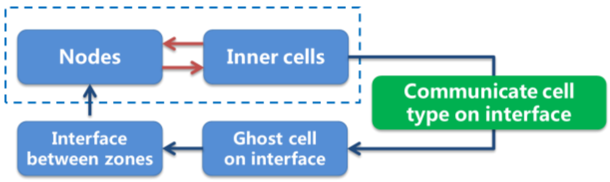


Figure 8: Communication of node type by the cells on the interface between grid zones.

The first method can only ensure that the advancing front goes across the interface of grid zones, while the second method can guarantee the exact communication of node type. Considering the case in Figure 9, for example, the number '1' denotes active, while '-1' indicates inactive. The red node in the left zone should be set to be active by the front advancing step. However, using the first communication method above, the red node is still inactive for the right zone because the type of the gray cell cannot be communicated successfully. This is the reason why the second method can guarantee the exact interface communication of node type.

The active regions should be further optimized to ensure overlapping with each other. The simplest way is to enlarge the active regions by several extra front advancing steps. In these optimizing steps, the second communication method above should be adopted to ensure the exact communication of the node type. Figure 10 shows the mesh before and after optimization for the NLR7301 airfoil. It can be seen that the overlapping region is wider than that without optimization, which ensure a proper donor cell for each interpolated cell.

For complex engineering applications, if solid walls of different bodies are very close to each other, the active region may expand to the inside of solid body by the front advancing method, which will result in failure of hole-cutting. To enhance the robustness of present method, a body-inside checking approach based on auxiliary grids was developed. Considering the NLR7301 airfoil again, for example, the auxiliary inner grids are shown in Figure 11. The coarse inside-body grids are generated only in the adjacent regions of the two foils. Nodes inside these auxiliary grids are identified as inactive ones in the preparatory step, and will be excluded during the following front advancing steps.

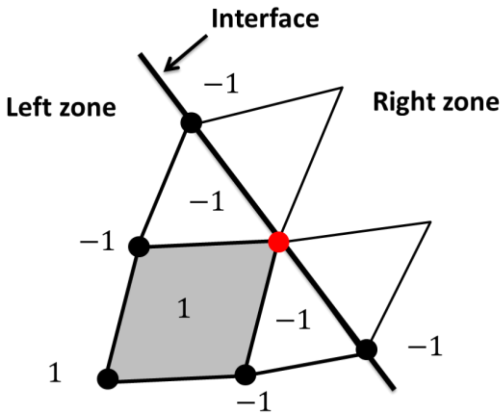
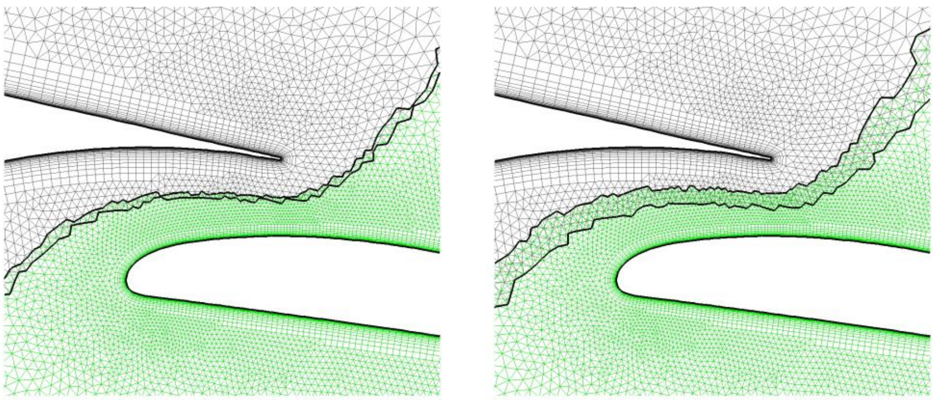


Figure 9: The case that the node's type can't be communicated correctly.



(a) Before optimizing (b) After optimizing

Figure 10: The overset mesh before and after optimizing for NLR7301 airfoil.

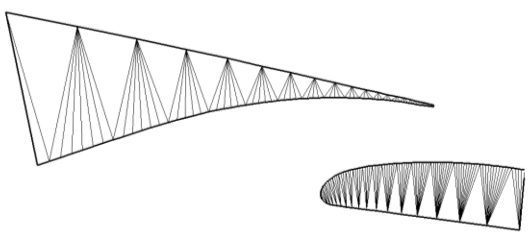


Figure 11: The auxiliary inside-body grids for NLR7301 airfoil.

2.3 Finding the donor cells for interpolated cells

The cell type can be further classified by its surrounding nodes: 1) the cell whose vertexes are all active is set to be active; 2) the cell whose vertexes are all inactive is set to be inactive; 3) the remaining cells must be interpolated cells.

The donor cells of the interpolated cells need to be founded in the neighboring zones. As mentioned above, a simple way is to search directly in the global mesh. However, it would become infeasible for large scale mesh because of the massive memory consumption. So the 'GTL' strategy as shown in Figure 2 can be used here to solve this problem. The procedure can be divided into following three steps:

(1) Gathering up the cell center coordinates of all interpolated cells in each grid zone by communication between processors.
(2) Searching the donor cell in local grid zones on the present processor.
(3) Selecting the best donor cell among processors with communication.

In the multi-body and multi-block cases, there may be several donor cells for one interpolated cell. Then the most appropriate one should be selected from the candidates. Firstly, the inactive donor cells must be excluded from the candidate list. After that, the minimum distance to wall is used as the criterion, and the one with the minimum distance will be chosen as the final donor cell. In the case of parallel assembling, the donor cell is firstly searched and selected in the local zones within their own processor. Then the criterion values are compared between processors by communication to find the most proper one.

Assuming that the number of donor cells is M, the number of total cells is N and the number of processors/zones is L. By ADT-based searching method, the calculation cost for each zone is:

$$M \log_2 (N / L) \tag{2}$$

Therefore, the calculation cost for donor cell searching is not linear with the number of processors in the GTL method. To improve the efficiency of donor cell searching, two strategies can be adopted: 1) Calculating the minimum boxes around the grid zones, and use the boxes to check roughly whether the nodes located inside or not. 2) Roughly determining the overset regions by the geometry of grid blocks. The cells which obviously do not lie in the overset region can be excluded in this searching step.

3 Test and discussion

An overlap mesh for five spheres is used to test the parallel implicit hole-cutting method. The overlap mesh system can be seen in Figure 12. Five grid blocks (colored regions) around the five spheres are embedded in a global unstructured grid (black region) which covers the entire computation region. Each grid block around a sphere contains 5.05 million hybrid cells. The global grid contains only tetrahedral cells whose total number is 10.1 million. The total cell number for the grid system is 35.4 million.

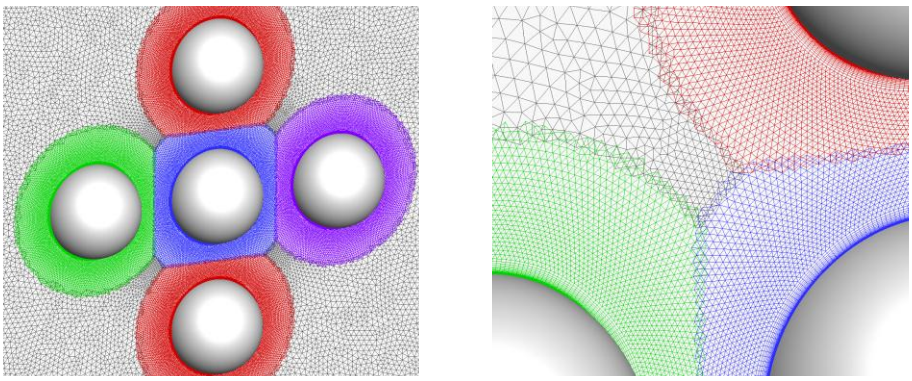


Figure 12: Mesh after hole-cutting for the test case with five spheres.

Table 1: Time of hole-cutting with different number of processors.

Number of processors	Node classifying (seconds)	Front advancing (seconds)	Interpolated cell gathering up (seconds)	Donor cell searching (seconds)	Total (seconds)
28	41.33	1.07	1.38	106.24	150.32
56	22.77	0.64	1.71	43.44	68.7
112	15.07	0.32	2.05	20.53	38.04
224	9.25	0.19	2.92	8.96	21.36
448	6.71	0.11	4.67	3.56	15.07
896	4.66	0.06	6.17	2.15	13.05
1792	3.41	0.12	6.18	3.18	12.9

This case is run on a HPC cluster in which the 1.5GHz FT1500 processor is equipped. Figure 12 shows the mesh after hole-cutting. It demonstrates the ability of present method to handle multi-body and multi-block problems. Using the criterion of minimum distance to wall, the interpolated cells are almost at the median lines of sphere surfaces, and all the active regions are simply connected domains.

Table 1 shows the CPU time of the hole-cutting process by different number of processors. Four parts (node classifying, front advancing, interpolated cell gathering up, donor cell searching) are taken into consideration. Figure 13 shows the CPU time proportion of each part, while Figure 14 plots the speedup performance.

Part one ('node classifying' in Fig.13) is the time to classify the nodes in the overlapping region, in which the donor cell of each node is needed to be determined. This part of time decreases monotonously with number of processors. However, the speedup performance increases slowly with the parallel scale. There are two main reasons: the first is the unbalance of the loading. In preprocessing, the overlap regions have been sorted roughly by the minimum box of each grid block in order to improve the efficiency. The nodes that located obviously out of the overlapping regions (such as the nodes at far-field) don't need further checking, so the loads between processors are unbalance. Another reason is the necessary computational tasks for the background mesh, including the calculation of geometry information and the minimum distance to wall, the construction of ADT data structure. These tasks have not been parallelized yet. However, the proportion of this part becomes less at larger parallel scale.

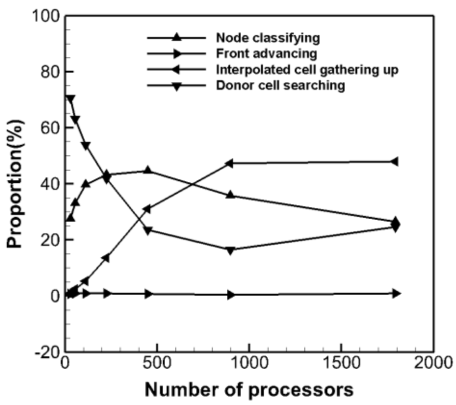


Figure 13: Proportion of the four main processes in hole-cutting.

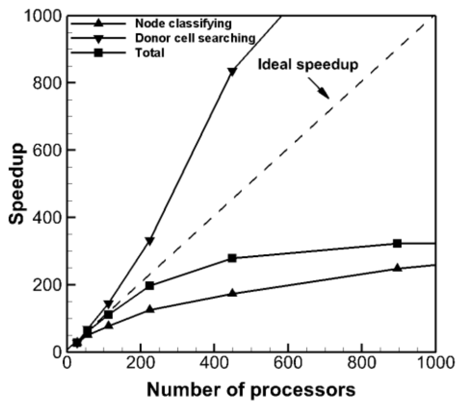


Figure 14: Parallel speedup for present test case.

The second part ('front advancing' in Fig.13) is the time to identify the node type in the active regions by front advancing. In each front advancing step all the cells are checked one by one. If any node of a cell is active, then all of the cell's nodes are set to be active. And the new attributes of nodes will be immediately and recursively used to check the neighboring cells. Therefore, the active region can be filled within only several advancing steps by this method. From Table 1 and Figure 13 we can see that the time proportion of this part is very small for different number of processors.

The third part ('interpolated cell gathering up' in Fig.13) is the time to gather up all the interpolated cells. This part of time increases monotonously with number of processors because of the communication task. In this communication process, each processor uploads its interpolated cells' coordinates to the server processor firstly, and then all the global interpolated cells' coordinates are downloaded to each processor from the server. So the communication task will become heavier gradually with the increasing parallel scale. The time proportion of this part ultimately becomes the largest one at large scale parallel hole-cutting.

The forth part ('donor cell searching' in Fig.13) is the time to search the donor cells for the interpolated cells. In this step the GTL strategy is adopted, the donor cell of each interpolated cell is searched firstly in the local grid zones within its own processor and then is checked by neighboring zone communication. This part of time decreases quickly with the increasing number of processors. However, finally this part of time increases slightly with the further increasing of parallel scale. The reason is the communication task in the logical judgment of donor cell selecting. From Figure 14 it can be seen that the speedup performance of donor cell searching is even larger than the ideal speedup value. It is due to the characteristics of the GTL strategy. As mentioned in equation (2), this part of time is not linear with the increasing of number of processors. And several preprocess strategies have been used to improve the efficiency. Before searching in the grid zone, it is firstly checked whether the node is located in the minimum box around the grid zone. Therefore, the searching time depends not only on the number of cells, but also the volume of zone. Here the number of partitioned grid zones equals to the number of processors, so both the volume and the number of cells would decreases with the increasing number of processors. The detailed analysis of GTL method can be seen in Ref. [22]. The time proportion of this part is the largest one when number of processors is relatively small. However, the proportion gradually becomes equivalent to that of part one.

The advantage of present parallel hole-cutting method is the less storage consumption. For large scale overset grids, the background grids can still be very coarse, so the major storage consumption is the coordinates of the global interpolated cells. In this case, the total number of interpolated cells is 0.59 million, which only occupy less than 15MB memory for double precision. Furthermore, the interpolated regions are thin layers, so the relationship between the number of interpolated cells and the number of computation cells can be written as:

$$m = kN^{2/3} \tag{3}$$

where the constant k is related to the topology of the mesh. So it can be estimated that the number of interpolated cells may be 5.6 million if the number of computation cells increases to 1.0 billion with the same topology. In this case, the memory storage for the interpolated cells' coordinates will be 128MB only. Then we can handle very large scale problems without the memory limitation using this parallel hole-cutting approach.

4 Applications

4.1 Store-separation

Based on the present implicit hole cutting method, a coupled solver of Navier-Stokes (NS) equation and six-DOF (degree of freedom, DOF) equation is developed to solve multi-body separation problems. The second order cell-centered finite volume scheme and the dual-time stepping method are used for the NS solver, while the loose coupling approach between the NS solver and the six-DOF solver is adopted for the coupled system.

The overset mesh for the two blocks before hole-cutting is shown in Figure 15. Hybrid cells are used to discretize the computation domain. The total number of cells is 6.8 million. Figure 16 shows the grids during the store separation, where the green region denotes the boundary of active cells over the store. The parameters for the store-separation can be seen in Ref. [23]. Figure 17 shows the kinematic parameters of the store during separation, in which *dx, dy, dz* represent the displacements in *x*, *y*, *z* directions respectively, and phi, theta, psi represent the rolling angle, pitching angle and yawing angle respectively. The present simulation results agree well with the test data.

The GTL strategy is used for the node classifying process. The total time for hole-cutting is about 7 seconds with 256 processors (1.5GHz FT1500), which is much less than the time of unsteady flow solver iteration.

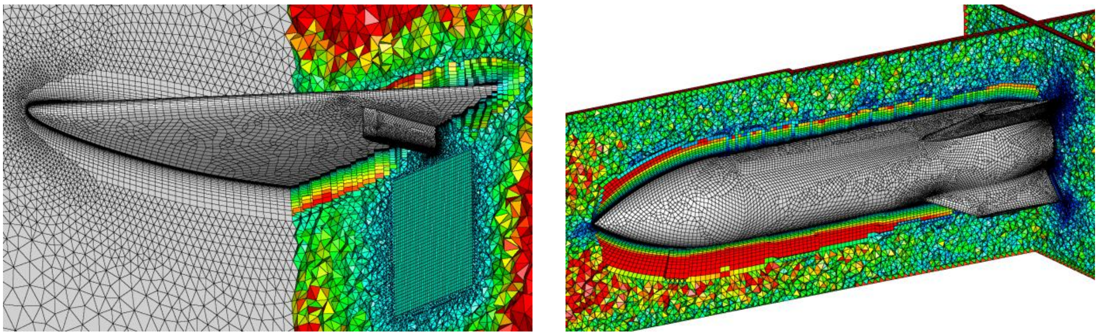


Figure 15: Initial grids around the wing/store configurations.

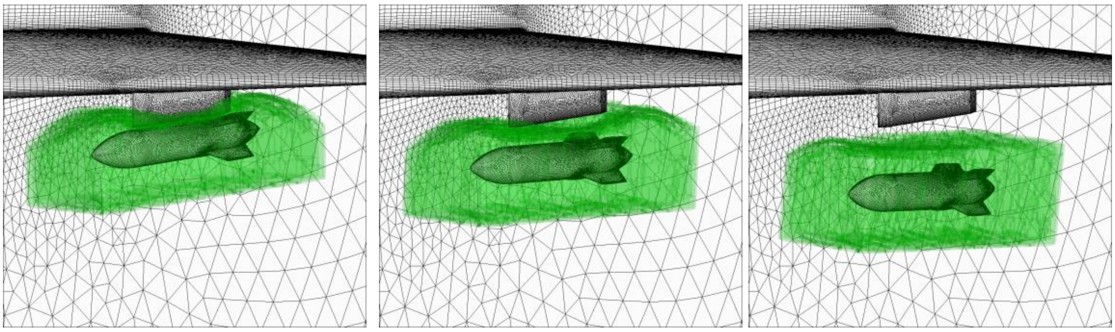
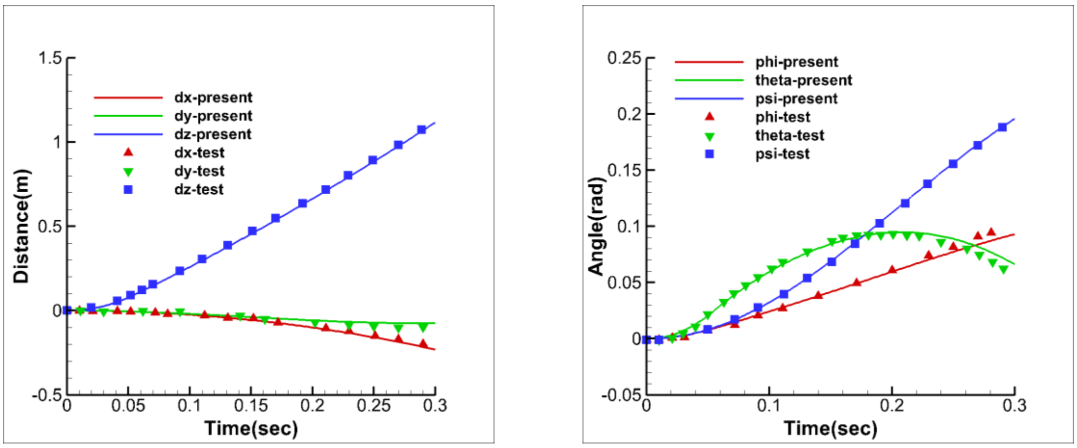


Figure 16: Overset grids over the store during separation.



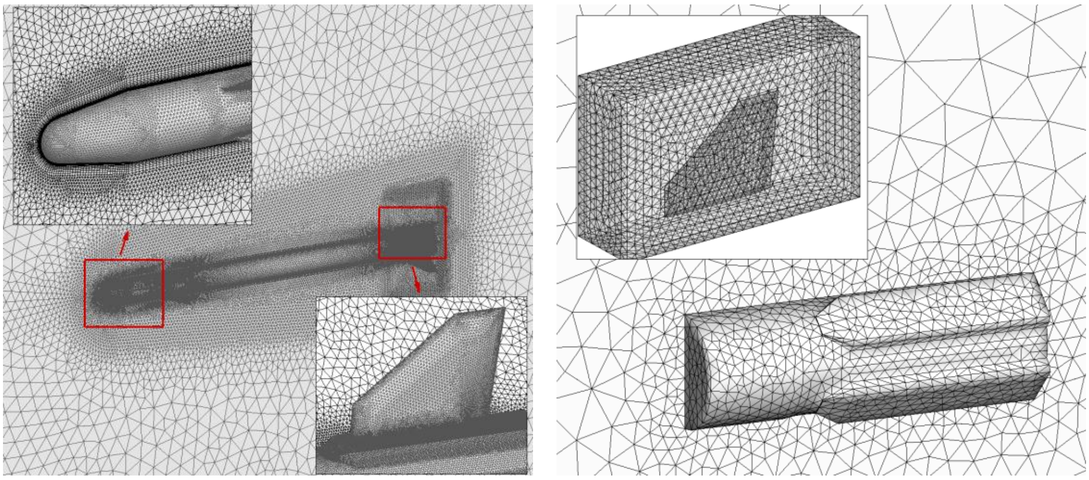
(a) Displacement (b) Euler Angle

Figure 17: Kinematic parameters of missile during separation

4.2 The 'numerical virtual flight' of a highly maneuverable missile

In this case the pitching process with flight control system of a highly maneuverable missile model is presented. The model contains four rotatable after-body rudders which control the body rolling, pitching and yawing. Figure 18(a) shows the initial mesh. Hybrid grids are used and the total number of cells is 8.5 million. Here only half of the model is considered since only the pitching motion is simulated. Figure 18(b) shows the auxiliary background mesh. In the background mesh, the domain far away from the rudders can be ignored because it is not located in the overlap region. The total cell number of background mesh is 0.13 million. The hole-cutting job can be done within 8 seconds for each rotation step of the rudders with 280 processors (1.5GHz FT1500).

Using the coupled solver of NS equation, six-DOF equation and flight control law, the pitching process of the model from 0° to 30° angle of attack is simulated. Figure 19 shows the missile's pitching angle and rudder's rotation angle during the flight control process. Figure 20 shows the instantaneous pressure contour and streamlines at a typical time step.



(a) Initial overset mesh (b) The background mesh

Figure 18: The overset mesh and background mesh for the highly maneuverable missile.

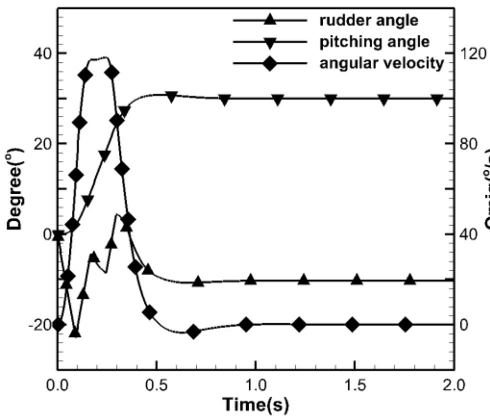


Figure 19: The missile's pitching angle and rudder's rotation angle during the flight control process

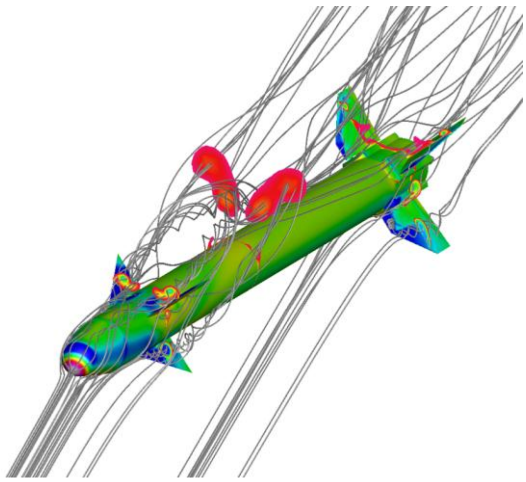


Figure 20: Pressure contour and streamlines at an instantaneous status.

4.3 Two dimensional fish schooling

One disadvantage of overset mesh method is that it cannot handle morphing boundary problems, which is very common in engineer applications such as wing aero-elastics, bio-fluid dynamics. Coupling with the deforming mesh method is a feasible way to simulate these cases with both boundary morphing and rigid movement.

The present implicit hole-cutting method has been coupled with the deforming mesh method based on radius basis function (RBF) strategy [24]. To improve the efficiency of RBF method, the surface node selecting technique based on maximum surface error is used to reduce the number of reference nodes for interpolation.

In the coupled dynamic mesh method here, the moving of a fish body is classified into two parts:

the morphing in its body coordinate system and the rigid moving in the computational coordinate system. Accordingly, the displacements in x and y directions (2D) of the volume nodes need to be calculated firstly by RBF method in each dynamic mesh generation step. After that, the entire grid block around the fish body moves by its own six degree of freedom (6DOF) parameters. Finally the implicit hole-cutting method is used to set up the relationship between grid zones.

The S-type starting process of four fish schooling (2D) in a diamond array was simulated. Here the strong coupled method [25] is used to solver the coupled problem of NS equation and kinetic equation of fish, due to the stability condition, since the flow density and the fish body density is very close. The swimming control law is not considered in this case. The active morphing of a fish body can be written as:

$$
\begin{aligned}
&dy = k(t)A(x)\sin\left(2\pi\left(\frac{t}{T}+\frac{x}{\lambda}\right)\right) \\
&A(x) = 0.02\text{-}0.0825x + 0.1625x^2 \\
&k(t) = \min\left(1.0, \frac{t}{T}\right)
\end{aligned} \tag{4}
$$

where x denotes the distance of nodes away from the fish head. dy is the displacement in the side direction of the fish. T is the morphing period and is set to $1s$ here. The wave length λ is set to $0.1m$ and is equal to the fish length.

Figure 21 shows the dynamic meshes during the schooling. It can be seen that it is very easy for the coupling approach to deal with these kinds of problems with both boundary morphing and rigid movement.

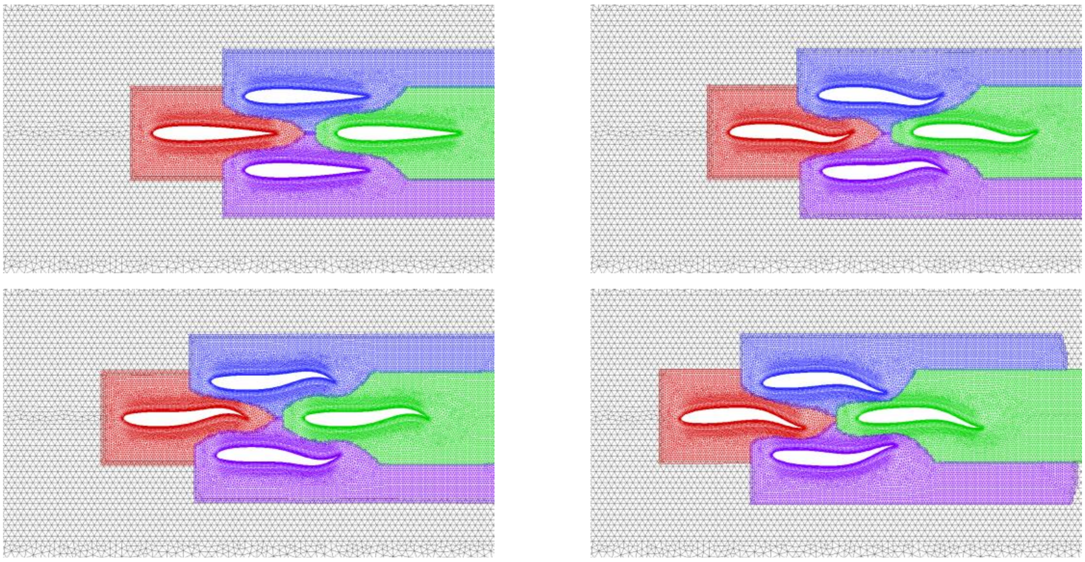


Figure 21: Dynamic meshes during fish schooling.

5 Conclusions

In this paper, a parallel IHC method for unstructured overset mesh was developed. The traditional IHC strategy based on minimum distance to wall is used to classify the type of nodes located in overlapping regions. Then a novel front advancing method is proposed to sort the active region. This front advancing method can maintain the simple connectivity of active regions. Test case shows that this implicit hole-cutting method is of highly automation and can handle multi-body or multi-block problems.

The method has been parallelized to deal with large scale overset mesh. To solve the problem of memory consumption limitation and lower parallel efficiency for large scale mesh, the global-to-local strategy and the background mesh method was developed. The test cases show the robustness and efficiency of present parallel method. However, the communication task between processors gradually becomes the major part of hole-cutting with the increasing of number of processors. It is mainly due to the gathering up procedure of the interpolated cells. The communication strategy need to be further optimized in our future work to improve the parallel efficiency.

Acknowledgements

This work is supported partially by National Key Research and Development Program (No.2016YFB0200700) and National Natural Science Foundation of China (Grant No. 11532016, and No. 11672324).

References

[1] Steger J L, Dougherty F C and Benek J A. A chimera grid scheme. Presented at Applied mechanics, bioengineering and fluids engineering conference, Houston, Texas, Jun.20-22, 1983.

[2] Noack R W, Slotnick J P. A summary of the 2004 overset symposium on composite grids and solution technology. AIAA paper 2005-921, 1995.

[3] Benek J A, Steger J L, Dougherty F A. A flexible grid embedding technique with application to the Euler equations. AIAA paper 83-1944, 1983.

[4] LaBozzetta W F, Gatzke T D. MACGS-towards the complete grid generation system. AIAA paper 94-1923, 1994.

[5] Chiu I T, Meakin R. On automating domain connectivity for overset grids. AIAA Paper 95-0854, 1995.

[6] Meakin R L. Object X-rays for cutting holes in composite overset structured grid. AIAA paper 2001-2537, 2001.

[7] Cho K W, Kwon J H. Development of a fully systemized chimera methodology for steady/unsteady problems. Journal of Aircraft, 1999, 36(6): 973-980.

[8] Li T H, Yan C and Li Y J. Investigation and enhancement of cut-paste algorithm in overlapping grid. Journal of Beijing University of Aeronautics and Astronautics. 2005, 31(4): 402-406 (in Chinese).

[9] Lee Y L, Baeder J D. Implicit hole cutting - a new approach for overset grid connectivity [R]. AIAA paper 2003-4128, 2003.

[10] Nakahashi K, Togashi F and Sharov D. An intergrid-boundary definition method for overset unstructured grid approach AIAA paper 99-3304, 1999.

[11] Togashi F, Ito Y, Nakahashi K and Obayashi S. Overset unstructured grids method for viscous flow computations. AIAA Journal, 2006, 44(7): 1617-1623.

[12] Lohner R, Sharov D, Luo H. Overlapping unstructured grids. AIAA paper 2001-0439, 2001.

[13] Luo H. An overlapping unstructured grid method for viscous flows. AIAA paper 2001-2603, 2001.

[14] Zhang S J, Owens S F. A parallel unstructured chimera grid method. AIAA paper 2005-4877, 2005.

[15] Sitaraman J, Floros M, Wissink A and Potsdam M. Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids. Journal of Computational Physics, 2010, 229(12): 4703-4723.

[16] Roget B, Sitaraman J. Robust and efficient overset grid assembly for partitioned unstructured meshes. Journal of Computational Physics, 2014, 260(1): 1-24.

[17] Post D E, ATWOOD C, et al. A new DoD initiative: The computational research and engineering

acquisition tools and environments (CREATE) program. Journal of Computational Physics, Conference Series 125, 2008.
[18] Brazell M, Sitaraman J and Mavriplis D. An overset mesh approach for 3D mixed element high order discretization. 53rd Aerospace Sciences Meeting, AIAA SciTech, January 5-9, 2015.
[19] Zagaris G, Campbell M T, Bodony, Shaffer E, Brandyberry MD. A toolkit for parallel overset grid assembly targeting large-scale moving body aerodynamic simulations. Proceedings of the 19th International Meshing Roundtable, 2010. 385-401.
[20] Landmann B and Montagnac M. A highly automated parallel Chimera method for overset grids based on the implicit hole cutting technique. International Journal for Numerical Methods in Fluids, 2011, 66(6): 778-804.
[21] Liang S, Zhang J, Lu Z H. Parallel overset grid assembly in large scale aerodynamic simulation. E-science Technology & Application, 2016, 7(3): 66-76 (in Chinese).
[22] Chang X H, Ma R, Zhang L P. Parallel implicit hole-cutting method for unstructured overset grid. Acta Aeronautica et Astronautica Sinica, 2018, 39(6): 121780 (in Chinese).
[23] Hall LH, Parthasarathy V. Validation of an automated Chimera/6-DOF methodology for multiple moving body problems [C]. The 6th AIAA Aerospace Sciences Meeting and Exhibit, January 12-15, 1998.
[24] Rendall T C S. Allen C B. Reduced surface point selection options for efficient mesh deformation using radial basis functions. Journal of Computational Physics, 2010, 229: 2810-2820.
[25] Chang X H, Ma R, Zhang L P, He X. Fully-implicit approach for flow/kinetic coupled problems based on dynamic hybrid mesh. The Eighth International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, Sichuan, China, July 14-18, 2014.