INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **33**: 919–938

A fully automated Chimera methodology for multiple moving body problems

Z. J. Wang and V. Parthasarathy*

CFD Research Corporation, 215 Wynn Drive, Huntsville, AL 35803, U.S.A.

SUMMARY

A fully automated Chimera methodology has been developed in this study to provide geometric or stencil information required to facilitate inter-grid data communications. Chimera holes are cut automatically in each grid of an overset grid system based on whether the grid overlaps with non-penetrable surfaces (NPS) and/or blocked regions. The efficiency of the hole-cutting algorithm is boosted with search algorithms based on the state-of-the-art alternating digital tree (ADT) data structures. The automated nature of the hole-cutting algorithm is ideally suited for handling multiple moving body problems. Several cases, both steady and unsteady, are used to demonstrate the effectiveness of the methodology. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Chimera methodology; multiple moving body problem; non-penetrable surfaces

1. INTRODUCTION

A multi-zonal overlapped grid approach (Chimera) provides a very powerful means of handling complex geometries in computational fluid dynamics (CFD). With Chimera, different zones of a multi-domain grid can intersect each other in an arbitrary manner. Independent/modular grid generation and a parts library of frequently used component grids become possible, drastically reducing the degree of difficulty and user time associated with grid generation. With its first introduction into CFD about a decade ago by Steger *et al.* [1] and Benek *et al.* [2,3], Chimera has been extended to solve both steady and unsteady Navier–Stokes equations for complex geometries [4,5]. For example, transonic flow around the complete space shuttle vehicle with over 16 million grid points was simulated successfully with Chimera [6], and an unsteady flow around a tiltrotor helicopter under forward flight was solved with moving overlapped grids [7].

For modeling complex problems that involve multiple bodies, it is more convenient to generate independent multi-block grids and composite them together into a system of overset

* Correspondence to: CFD Research Corporation, 215 Wynn Drive, Huntsville, AL 35803, U.S.A.

Received 24 July 1998
Revised 1 March 1999

grids that represents the computational grid. In a Chimera grid system, inter-grid communications are facilitated through data interpolations. Although the time taken to generate overset grids for a flow analysis is significantly reduced, establishing valid inter-grid communications between the overset grids is not trivial. First, valid Chimera holes need to be cut in each grid in regions that overlap with solid bodies or any other non-flow regions that belong to the other grids of the overset grid system. Next, interpolation stencils have to be established for cells that lie along the hole–boundary as well as other interpolation boundaries identified in each grid. Currently, most Chimera grid users utilize PEGSUS [8] or DCF [5] to generate interpolation stencils. Although these codes are very powerful, they require tremendous technical expertise from the user, especially in the task of hole cutting. For complex geometries, hole cutting is performed through trial and error and thus can be very expensive. For example, cutting valid holes for a complete aircraft flow analysis can take up to one man-month. This cost is difficult to justify in a design process. The difficulty associated with hole cutting in a Chimera scheme has undoubtedly hindered its wider applications in the industry. This is probably why there has not been a commercial Chimera CFD code in the market.

In this paper, the authors present a new methodology that automates the hole cutting and stencil identification algorithm in a Chimera scheme. The new approach has been tested with problems that involve very complex geometries for both steady and unsteady state computations, which involve multiple moving bodies. State-of-the-art data structures and search algorithms are developed to minimize CPU time taken in the search operations. Several demonstration problems are presented to demonstrate the capability of the approach.

2. AN AUTOMATED CHIMERA METHODOLOGY

There are two major steps in the traditional Chimera scheme to establish inter-grid communications:

- Hole cutting, which essentially involves blanking cells of a grid in regions that overlap with non-penetrable surfaces (NPS) in the other grids of an overset mesh and identifying the Chimera boundary cells that lie along the hole or fringe boundary surface (FBS) as well as interpolation boundary surface (IBS).
- Identification of interpolation stencils, which involves finding a donor cell for the Chimera boundary cells in each grid and interpolating the solution from the donor cells to the boundary cells.

With PEGSUS, the user needs to define the hole cutting boundaries and donor zones for each grid manually. For complex three-dimensional geometries, defining proper hole cutting boundaries can be extremely challenging. The user needs to know all the details of the computational grids and make trial and error decisions on where to cut holes in the grids. Furthermore, specifying the donor grids for boundary cells on the hole boundary and interpolation boundary is not a trivial task either. All the aforementioned drawbacks are further compounded while performing simulations that involve multiple moving bodies. In

general, given a system of overset grids with arbitrary grid overlap, hole cutting based on user-defined input demands tremendous technical expertise from the user.

The automated Chimera methodology presented by the authors circumvents all these drawbacks and simplifies the task, in most of the cases, to that of just defining appropriate boundary conditions for each grid. Another powerful feature of the present work is that it is applicable to both structured and unstructured grids. The cases studies presented here include only structured grids. However, it should be mentioned that the data structures employed for the Chimera scheme treat the structured grid akin to an unstructured, hexahedral grid and the methodology is being extended for unstructured grids with other topologies, such as tetrahedra and prisms.

2.1. Grid relations and user input

In earlier Chimera algorithms, some limitations may be placed on the way the overset grids interact with each other. In the present work, in order to allow flexible grid interactions, the overset grids are allowed to overlap each other in an arbitrary manner. The major–minor grid relationship that is conventionally used in traditional Chimera schemes is completely removed. Instead, each grid is considered an independent entity, which can have hole boundary cells or fringe boundary cells (FBC), and/or interpolation boundary cells (IBC). Each grid communicates with one or more grids in the overset mesh system via data interpolation to FBCs and IBCs. Once the inter-grid communication is completed, from the standpoint of flow solver, there is no distinction in how these different grids of the overset grid system are treated. Since automation is the primary goal, the present methodology demands only minimal input from the user. The user is required to provide only the following information:

1. NPSs, usually the solid wall boundaries in each grid.
2. IBSs, usually the outer boundary of a grid.
3. For a given grid, the user may also define a set of donor grids that would contain the interpolation stencils for its IBCs/FBCs. This option is provided in order to resolve ambiguities in cases where the IBCs/FBCs of a grid might have more than one donor. If this information is not provided in the user input, it is found automatically by the interpolation algorithm.

2.2. Automated hole cutting

A Chimera hole needs to be cut in a grid only when it overlaps with an NPS that belongs to another grid in the overset mesh. If there are blockages in a grid, outer surfaces of the blocked region are automatically treated as an NPS because no flow is allowed within the blocked regions. With the NPS being the only determining factor in cutting a hole in any grid, the first step in the automated hole-cutting algorithm is to build a list of NPS faces for the overset grid system. The normal vector of all NPS faces is oriented towards the flow domain, i.e. the flow domain is considered to be outside the NPS. All the cells of a grid that lie inside the NPS must be blanked in order to create the Chimera hole. Consider an illustrative overset grid shown in Figure 1. The steps involved in generating a Chimera hole and identifying the FBCs/IBCs are outlined as follows:

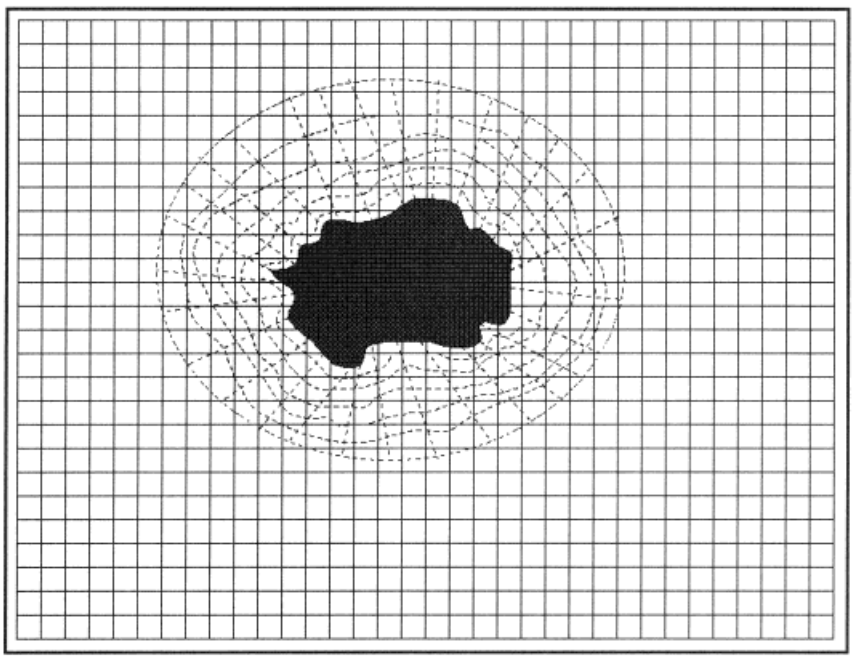

Figure 1. Illustration of a grid overlapping with an NPS represented by the outer boundary of the solid wall region. The dotted line refers to the grid that the NPS belongs to.

Step (1) Determine the bounding boxes ($\{x_{\min}, y_{\min}, z_{\min}\}$, $\{x_{\max}, y_{\max}, z_{\max}\}$) of all the cells in each grid of the overset mesh system as well as those of the faces on the NPS.

Step (2) In order to cut a Chimera hole in a grid, the algorithm has to first identify the cut-cells that intersect the NPS. The number of cut-cells is a very small percentage of the total number of cells in a grid. Hence, identifying the cut-cells could involve a very exhaustive search. However, this search is highly optimized using the alternating digital tree (ADT) scheme. For every face in the NPS, the algorithm searches the ADT of the given grid and creates a list of cells whose bounding boxes overlap with that of the NPS face. This shortlist of cells is examined further in order to identify the cut-cells.
Referring to Figure 2, NPS face ***ABC*** overlaps with that of the cell ***1-2-3-4-5-6-7-8*** if the respective bounding boxes represented by the six-dimensional points

$$(x_{\min} \quad y_{\min} \quad z_{\min} \quad x_{\max} \quad y_{\max} \quad z_{\max})_f$$

$$(X_{\min} \quad Y_{\min} \quad Z_{\min} \quad X_{\max} \quad Y_{\max} \quad Z_{\max})_c$$

satisfy the following inequality:

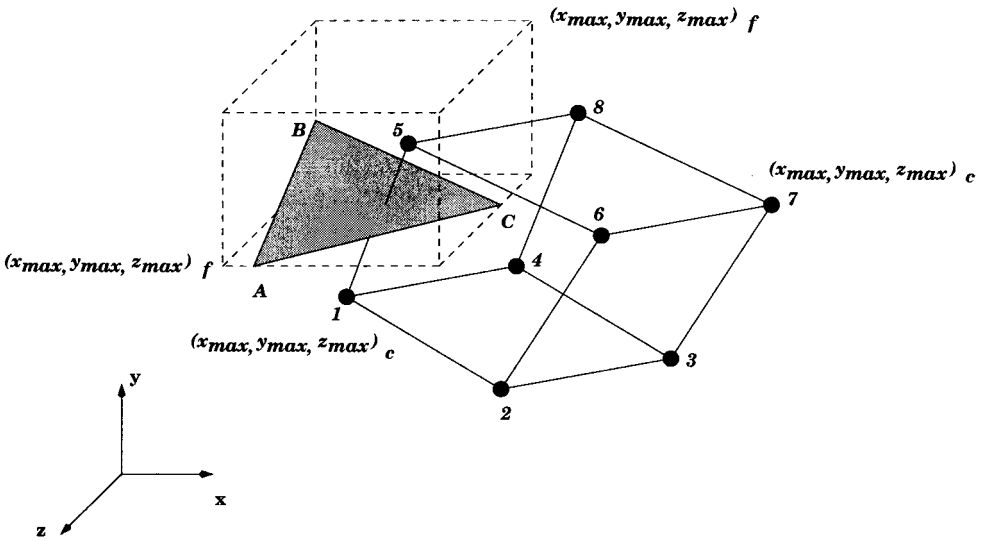

Figure 2. Overlapping bounding boxes of an NPS face and a cut-cell.

$$\begin{bmatrix} X_{\min} \\ Y_{\min} \\ Z_{\min} \\ x_{\min} \\ y_{\min} \\ z_{\min} \end{bmatrix} \leq \begin{bmatrix} X_{\min} \\ Y_{\min} \\ Z_{\min} \\ X_{\max} \\ Y_{\max} \\ Z_{\max} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ y_{\max} \\ z_{\max} \\ X_{\max} \\ Y_{\max} \\ Z_{\max} \end{bmatrix}$$

The inequality implies a component-by-component comparison of the six-dimensional vector.

Step (3) To determine if a cell is a *cut-cell*, the algorithm examines if one or more edges that belong to the cell intersect the NPS face. Consider an NPS face ***ABC*** and an edge $\vec{r}_{ab}$, as shown in Figure 3. The edge is first tested for intersection with the plane ***P***, which the NPS face belongs to. This it true if the following condition is satisfied:

$$(\vec{r}_{Aa} \cdot \hat{n}) \times (\vec{r}_{Ab} \cdot \hat{n}) < 0 \tag{1}$$

where $\hat{n}$ is the normal vector of the NPS face. This normal vector does not rely on any canonical local node-ordering scheme. The normal is always oriented away from the NPS and into the computational domain. Equation (1) determines if a given line segment intersects a given plane. The edge does not intersect the plane ***P*** if

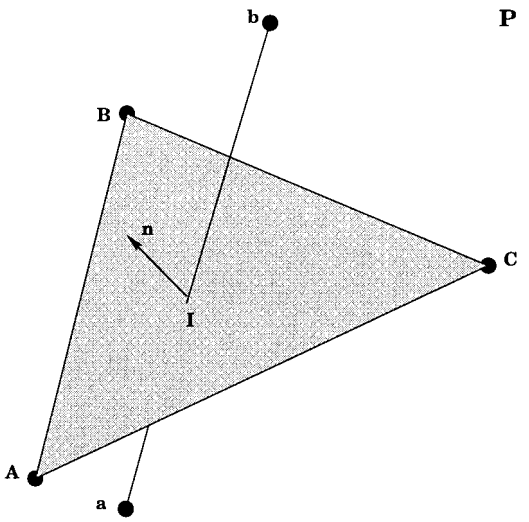

Figure 3. Intersection of an edge with a face.

$$(\vec{r}_{Aa} \cdot \hat{n}) \times (\vec{r}_{Ab} \cdot \hat{n}) \geq 0$$

Step (4) If Equation (1) is satisfied, the point ***I*** at which the edge intersects the plane ***P*** is determined as follows:

$$x_I = x_a + \chi(x_b - x_a)$$

$$y_I = y_a + \chi(y_b - y_a)$$

$$z_I = z_a + (z_b - z_a)$$

where $0 \leq \chi \leq 1$ and $\chi = |(\vec{r}_{Aa} \cdot \hat{n})/(\vec{r}_{ab} \cdot \hat{n})|$. This assumes that all the NPSs are planar. Any polygonal NPS face is split into planar triangular faces to perform the above test. Higher-order surfaces are not considered in the present work. It may be expensive to compute the intersection between an edge and a high-order surface.

Step (5) Once the co-ordinates of ***I*** are determined, the next step is to examine if this point lies inside the NPS face ***ABC***. This is true if the condition

$$\Delta ABI + \Delta BCI + \Delta CAI = \Delta ABC \tag{2}$$

is satisfied. Δ in the above equation represents the area of the respective triangles.

Step (6) If an edge $\vec{r}_{ab}$ satisfies Equation (2), the next step is to distinguish the status of its two end points as either IN or OUT, with respect to the NPS face. Node *a* lies OUT with respect to the NPS face if

$$\vec{r}_{Ia} \cdot \hat{n} \geq 0$$

where $\hat{n}$ is the outward normal vector of ***ABC***. Node *a* lies IN with respect to the NPS face if

$$\vec{r}_{Ia} \cdot \hat{n} \geq 0$$

If an edge intersects an NPS face, then the status of its nodes should conform to the following rules, as illustrated in Figure 4:

- If an edge intersects an NPS face an *odd* number of times, then one of its nodes must lie IN with respect to the NPS face and the other must lie OUT.
- If an edge intersects an NPS face an *even* number of times, then both its nodes must lie either IN or OUT with respect to the NPS face.

Note that if a line segment lies in the plane of a face, they are not considered intersected.

Step (7) Based on Steps (1)–(6), all edges in the given grid that intersect the NPS are detected and are assigned the proper status, as shown in Figure 5. The algorithm then propagates the status recursively through the other edges until all grid points that are IN are identified, as shown in Figure 6.

Step (8) Next, the cells that contain grid points classified as IN are blanked, as shown in Figure 7, thus generating the Chimera hole. The limiting boundary of the Chimera hole is constituted by the *cut-cells*. The cells that lie outside the NPS, adjacent to the Chimera hole are identified as FBCs, as shown in Figure 8.

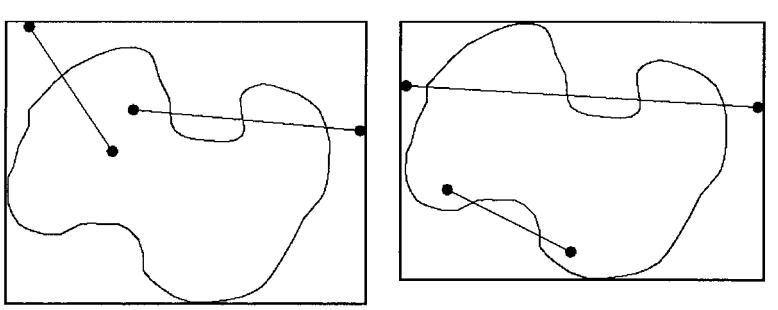

Figure 4. Classifying the status of end points of an edge that intersects the NPS as either IN or OUT: (a) edge with an *even* number of intersections; (b) edge with an *odd* number of intersections.

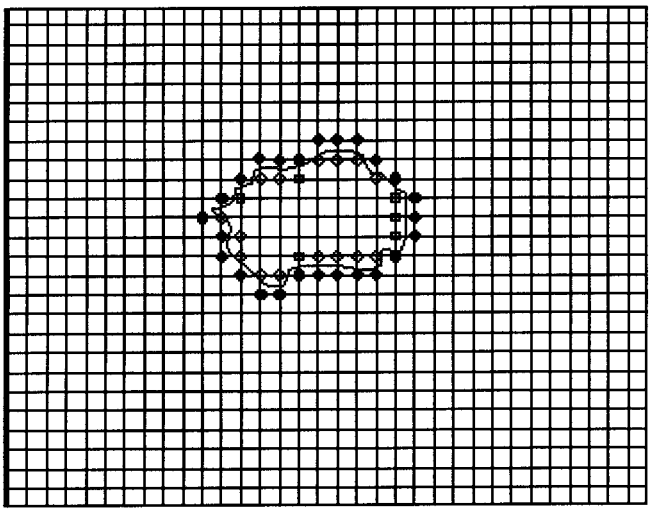

Figure 5. Assignment of IN/OUT status to end points of all edges of the grid that intersect the NPS faces.

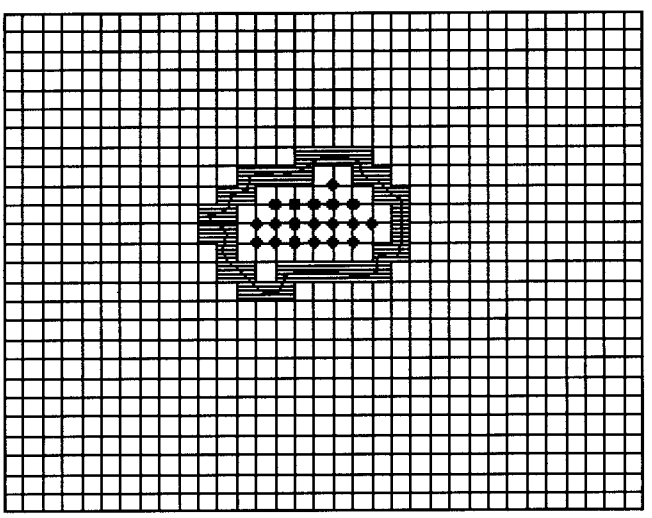

Figure 6. Cut-cells of the grid blanketing all the grid points that are identified as IN with respect to the NPS.

Step (9) If the grid has an interpolation boundary, the cells that lie on that boundary are identified as IBCs.

Steps (2)–(9) are repeated for all the grids in the overset grid system. The next step is to identify an interpolation stencil for the FBCs/IBCs in all the grids.

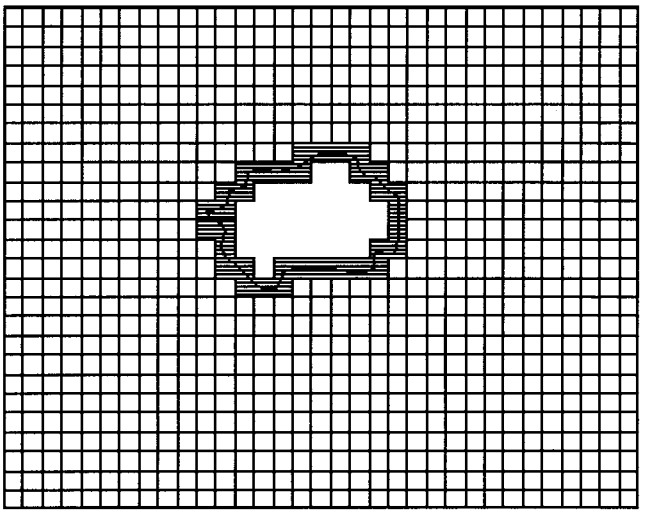

Figure 7. Chimera hole cut in the grid with respect to the given NPS. Cut-cells represent the outer limit of the hole boundary.

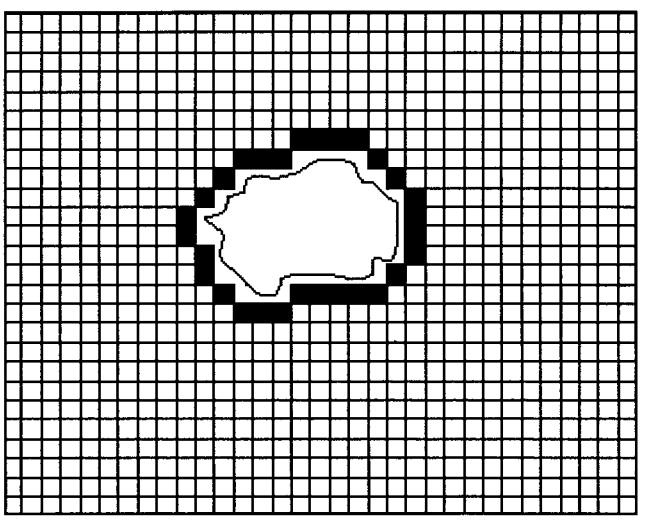

Figure 8. FBCs adjacent to the Chimera hole boundary.

2.3. Stencil identification and interpolation

Each grid in an overset grid system may contain IBCs and/or FBCs. They represent the linkage between the grids in an overset mesh. Inter-grid communication occurs by virtue of interpolating data to IBCs/FBCs in each grid from the respective *donors* in another grid. A donor is a

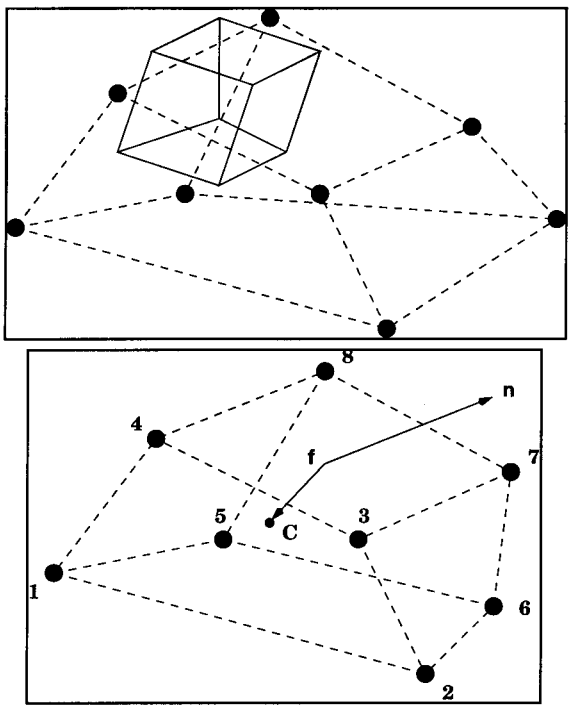

Figure 9. Chimera boundary cell is shown in solid lines. The dashed line represents a potential donor cell whose bounding box overlaps with that of the Chimera boundary cell.

cell that overlaps with the given IBC/FBC and provides an interpolation stencil. If the user does not specify the candidate grids that could contain a donor, the algorithm searches all the grids of the overset mesh to find the best donor. The following algorithm is used to find a donor for a given IBC/FBC.

1. Using the ADT, search all the other grids and identify the cells whose bounding boxes overlap with that of the IBC/FBC. Using these candidates, identify the cell that contains the centroid of the IBC/FBC. Referring to Figure 9, centroid ***C*** of the Chimera boundary cell lies inside the donor cell ***1-2-3-4-5-6-7-8*** if the inequality

$$\vec{r}_{fC} \cdot \hat{n} \geq 0$$

   is satisfied for each face of the cell. In the above equation, $\hat{n}$ represents the normal vector to the face and f denotes the centroid of the face.
2. Once a donor is identified, trilinearly interpolate the solution at the nodes of the donor to the centroid ***C*** of the IBC/FBC as follows:

$$q_C = q_1 \cdot s_2 s_4 s_6 + q_2 \cdot s_1 s_4 s_6 + q_3 \cdot s_1 s_3 s_6 + q_4 \cdot s_2 s_3 s_6 + q_5 \cdot s_2 s_4 s_5 + q_6 \cdot s_1 s_4 s_5 + q_7 \cdot s_1 s_3 s_5 + q_8 \cdot s_2 s_3 s_5$$

where q_i represents the solution at the corner nodes of the hexahedral cell and s_f represents the normalized distance from the centroid of the donor cell to the face f. Similar trilinear interpolations are formulated for other grid topologies, such as tetrahedra.

3. In the event of finding more than one valid donor, select the donor with the smallest volume (best grid resolution).

When the donor cell is identified for a given IBC/FBC, proper interpolation of the solution from the donor to the Chimera boundary cell is important in order to have an accurate transfer of data between the two grids. Trilinear interpolations are chosen because they are fully compatible with a second-order finite volume flow solver since a cellwise linear reconstruction algorithm is used in each cell. To facilitate data interpolation, the cell-centered solution is converted to node-centered in the donor grid. Furthermore, use of the weight functions s ensures that the solution at nodes that are closest to the centroid of the IBC/FBC get the maximum weight in the transferred data.

2.4. Data structures

The ADT [9] is the corner stone of the present Chimera methodology. The hole-cutting algorithm, as described above, involves identifying edges in each grid that intersect the NPS. This requires the algorithm to first identify the cells that overlap with the NPS faces. Similarly, the interpolation algorithm has to identify the cells that overlap with the FBCs/IBCs of each grid in order to find their donor. These searches could be very expensive for large grids. In a steady state simulation, wherein the grid geometries remain invariant, hole cutting and interpolation stencil identification both need to be done only once. However, for a simulation that involves moving bodies, these procedures have to be done repetitively as the orientation of the overset grids relative to each other changes with time. Hence, it is imperative to expedite the search processes. The ADT is an extension of a one-dimensional binary tree. It provides a general method for building a tree with a given set of data in n-dimensional space. In the present work, the dataset used for building the ADT comprises the bounding box co-ordinates of the grid cells. The ADT optimizes the search operations to $O(\mathrm{Log}_2 N)$ as opposed to $O(N)$.

3. EXTENSION TO MOVING BODY PROBLEMS

The automation of hole cutting and interpolation procedures makes the present Chimera methodology a powerful tool for simulation of flows with multiple moving bodies. The presence of moving bodies changes the relative orientation of the overset grids continually during the flow simulation. This entails Chimera holes to be cut anew and interpolation coefficients to be recomputed, as the grid (around a moving body) traverses through the computational domain. Without any automation, such problems require the

user to provide *a priori* all the input that would enable hole cutting and stencil 1identification for any given configuration of the overset grids. This has been one of the drawbacks of codes such as PEGSUS that are not closely coupled with the flow solver. In the present work, the Chimera scheme is tightly integrated with the flow solver, unlike PEGSUS. The steps involved in performing a moving body simulation may be outlined as follows:

1. Define initial grid geometries.
2. Compute a global time step Δt for all the overset grids.
3. Implement the Chimera scheme, which involves:
   - building an ADT for each grid if it is identified to have an FBS;
   - cutting a Chimera hole in each grid, if necessary, and identifying the FBCs/IBCs;
   - finding donors and establishing interpolation stencils for IBCs/FBCs in each grid;
   - enabling inter-grid communications by interpolating data to the IBCs/FBCs;
   - removing the existing ADTs as they have to be rebuilt due to the fact that the grid geometries would change in subsequent steps.
4. Advance the solution by Δt in all grids using the flow solver.
5. Redefine the geometries of grids that belong to the moving bodies. This procedure is governed by the dynamics and kinematics of the bodies in motion.
6. Go to step 2.

The use of efficient tree-based data structures reduces the CPU time required for hole cutting and solution interpolation to a fraction of the CPU time required for the flow solver. However, it is not necessary, in most simulations, to rebuild the Chimera hole and recompute the interpolation coefficients in each time step. This is due to the fact that the time step Δt, computed based on stability criteria, is usually small enough so as not to affect the grid geometries drastically in subsequent steps. Hence, it would suffice to implement step 3 with a user specified frequency in order to reduce further the computational cost of the simulation.

4. DEMONSTRATION CASES

The study cases presented in this section demonstrate the capability of the automated Chimera scheme developed in the present work. The discussion places emphasis on bringing forth the generality of the automated approach by employing it for different geometries. Both steady and unsteady simulations are included.

4.1. Flow around a three-element airfoil

The grid employed for this problem is shown in Figure 10. The boundary conditions, with regard to the Chimera scheme, that are required to be specified by the user are as follows.

- Solid wall boundaries on the surface of the three airfoil elements that constitute the NPS for the overset grid system.
- Interpolation boundaries on the outer surface of the grid around the slat and the flap.

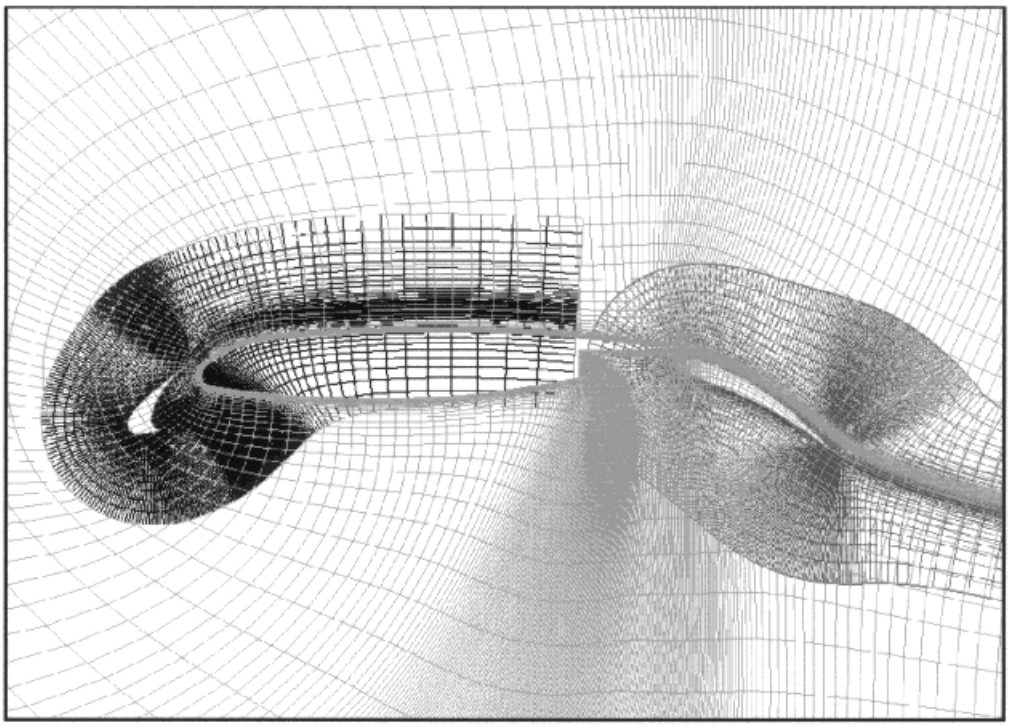

Figure 10. Overset grid around a three-element airfoil configuration comprised of grids around the slat, the main airfoil and the flap.

It is seen from the figure that each grid in the overset grid system overlaps with solid wall boundaries of one or more of the other grids in the system. The automated Chimera approach can handle these complications without any user intervention. This approach requires fringe boundaries to be created in each grid by cutting holes in the regions that overlap with a NPS. Chimera holes cut in the main airfoil grid, the slat grid and the flap grid are shown in Figures 11–13. It is also seen from the figures that the Chimera holes in each grid have receded further away from the respective NPS. This is done so that the FBCs on an FBS would be associated with a donor that is not located immediately next to a solid wall boundary. When data is interpolated to the FBCs from such donors, the overall solution accuracy was observed to improve. This feature of the Chimera scheme is easily controlled by prescribing a preferred number of *buffer layers* by which to recede an FBS from its associated NPS.

4.2. Unsteady simulation of a store separating from a wing

This case demonstrates a two-dimensional study of a store separating from a wing. The airfoil cross-section of the wing and ogive–cylindrical cross-section of the store are shown in Figure 14. The grid around the airfoil is a multi-block structured grid and is overset on the grid around the store. The user needs to identify the following boundaries in the input for this case.

- Solid wall boundaries on the surface of the wing and the store. These two boundaries together constitute the NPS for this overset grid system.

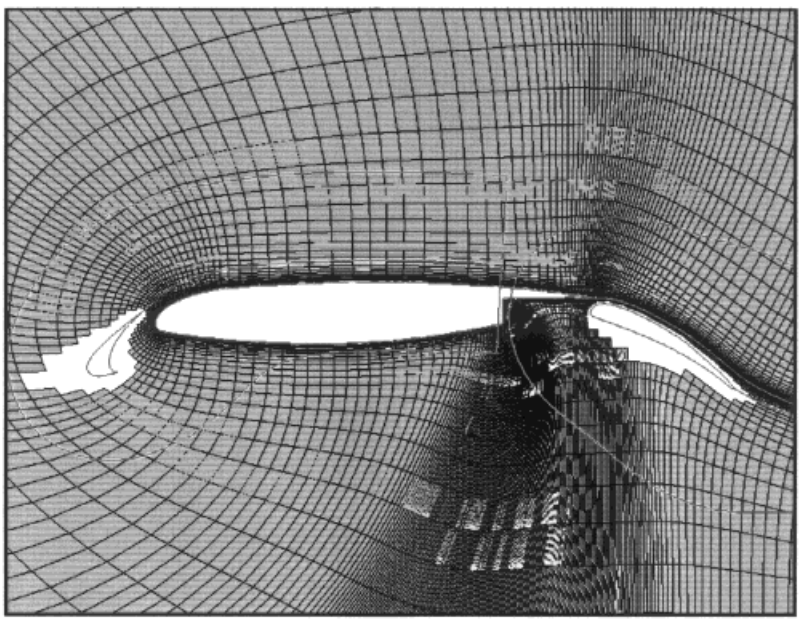

Figure 11. Chimera hole cut in the main airfoil grid around the solid wall boundaries represented by the slat and the flap.

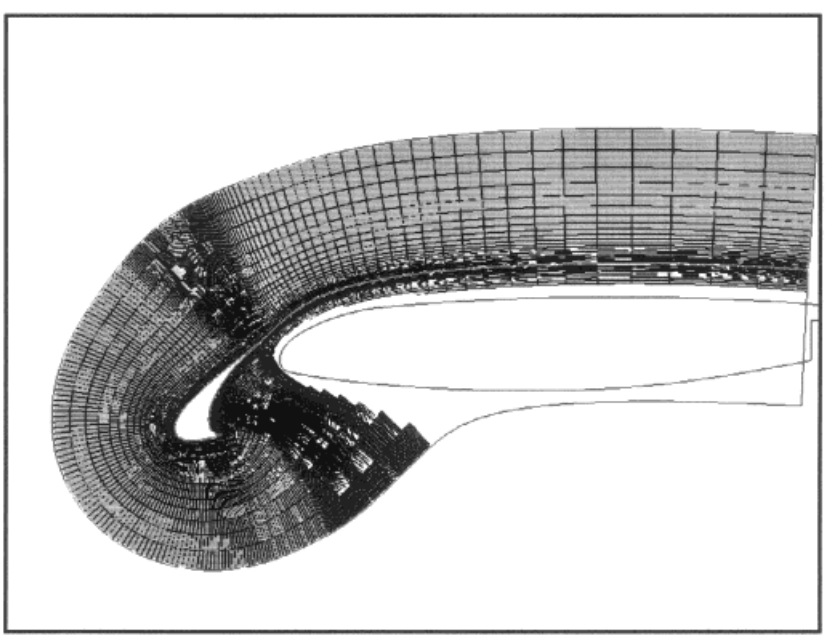

Figure 12. Chimera hole cut in the slat grid around the solid wall boundary represented by the main airfoil.

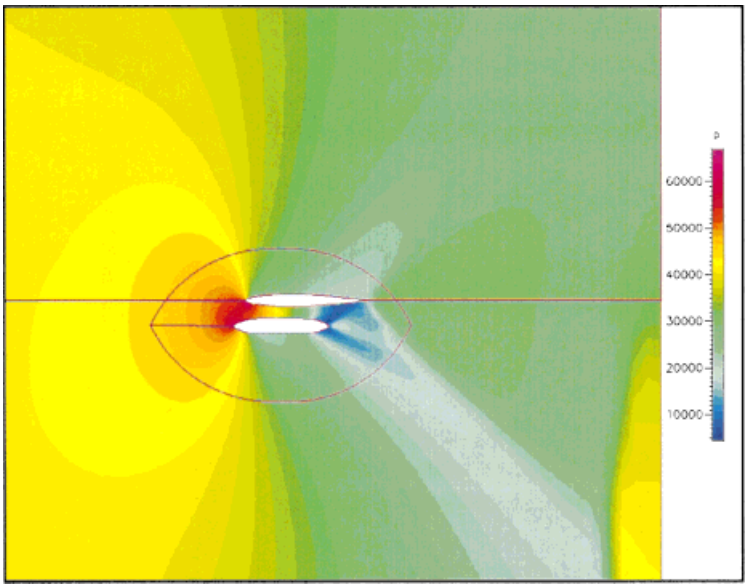

Plate 1. Pressure contours of the steady state solution obtained for the wing–store grid using Chimera scheme.

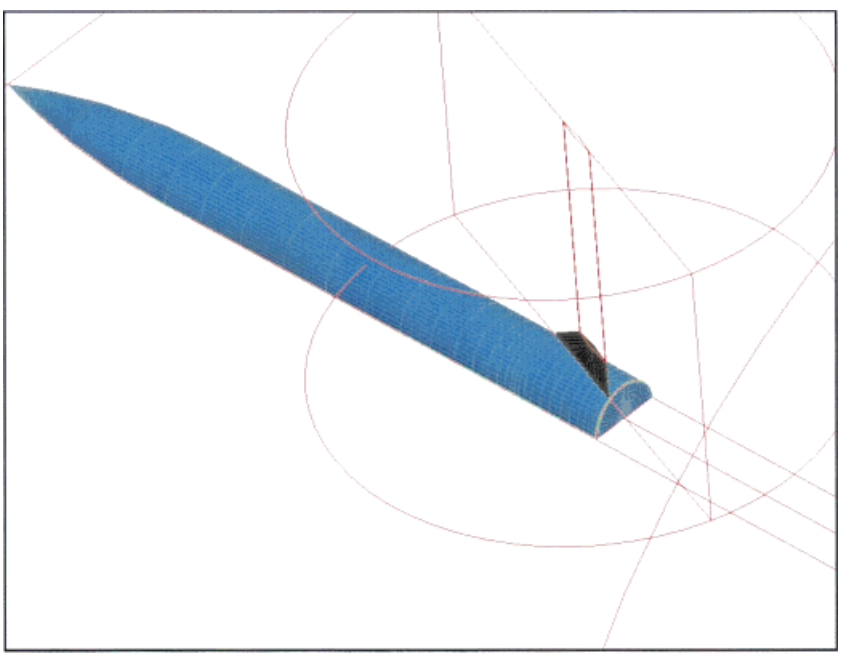

Plate 2. Overset mesh employed for modeling fin gap effects in missile control fin design. The solid wall boundaries of the missile and the fin body are shown. The multi-domain grid around the missile fin geometry comprises seven zones.

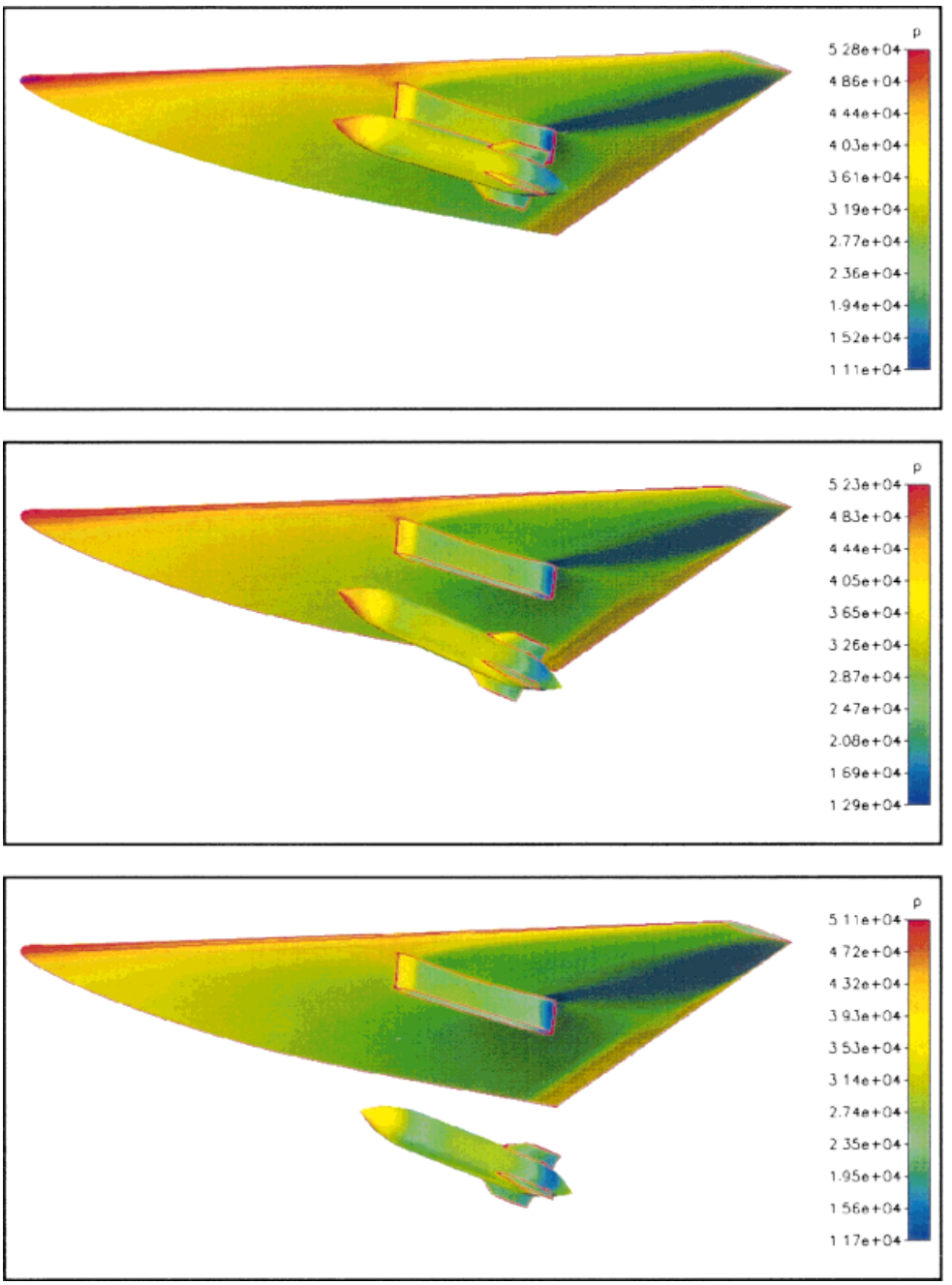

Plate 3. Pressure contours on the surface of the delta wing and the store body and $t = 0$ s, $t = 0.191$ s and $t = 0.32$ s. Inviscid solution corresponds to a free-stream Mach number of 0.95 and angle of attack of 0.0°.

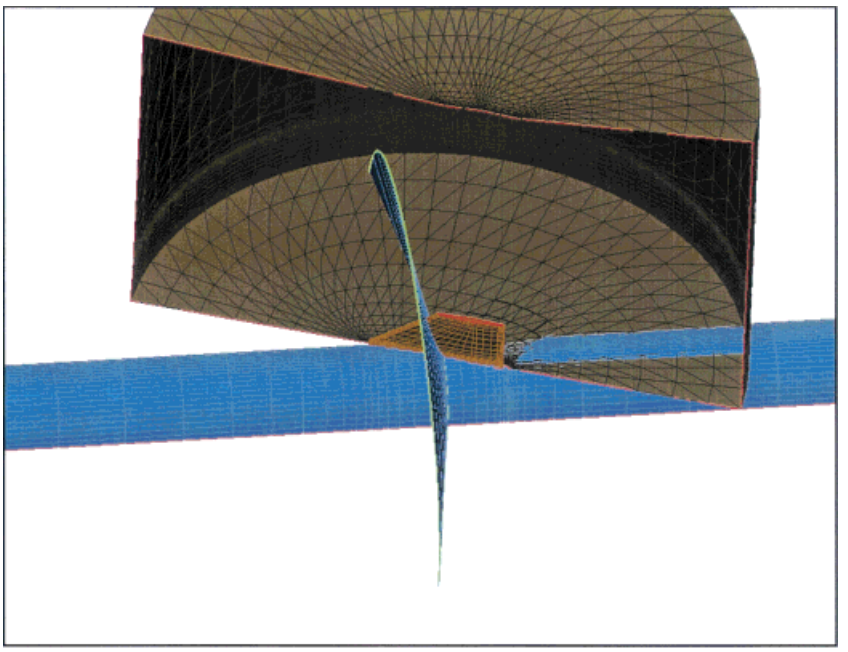

Plate 4. An illustration of inter-grid relationship in the missile fin overset mesh system. A curvilinear section of the missile grid is seen intersecting the fin body and the grid around the fin body is seen overlapping with the missile surface.

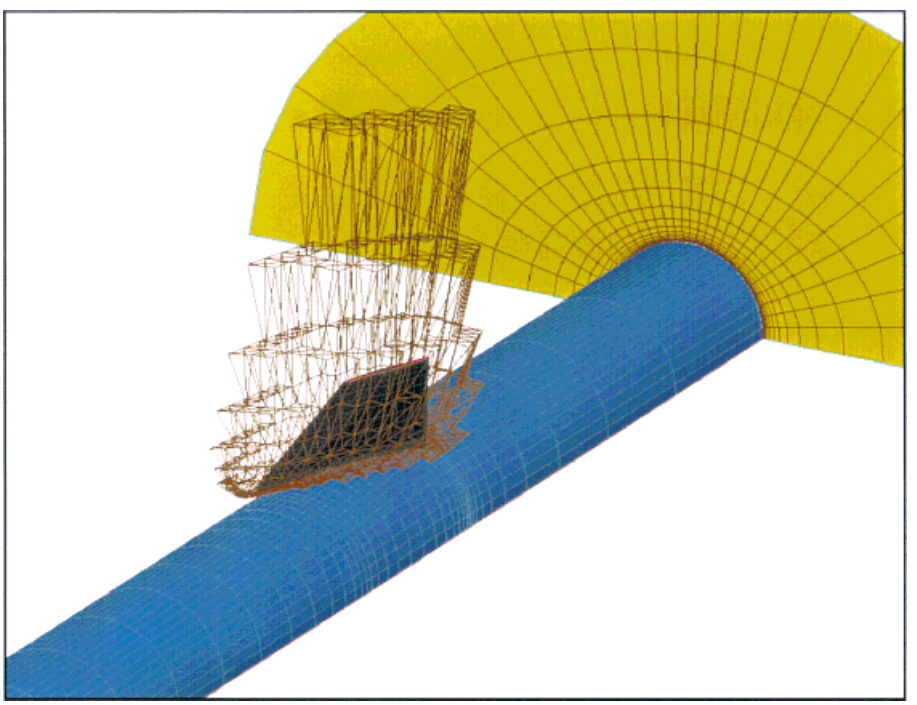

Plate 5. An illustration of hole cutting performed by the automated Chimera scheme. The wire frame shows the FBS of the Chimera hole blanked in the missile grid around the fin body.

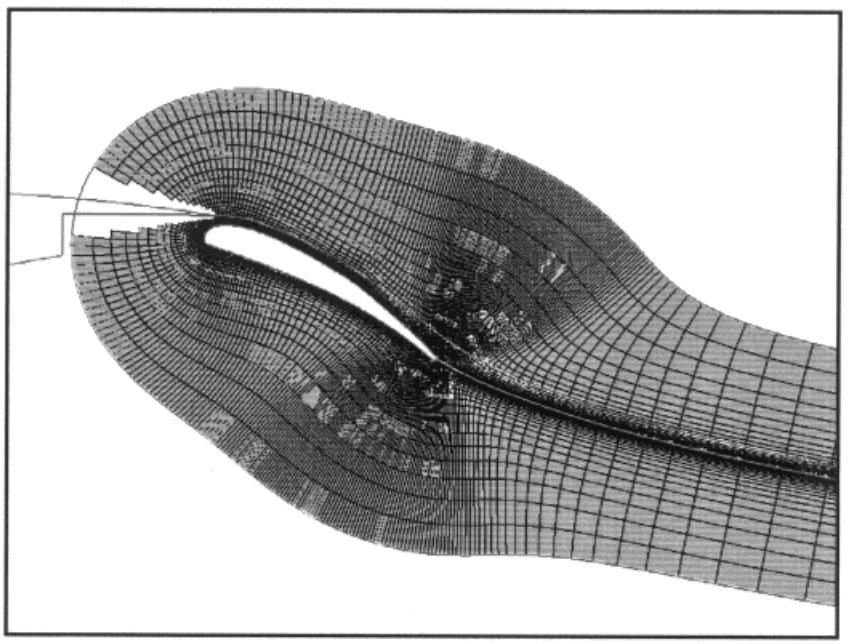

Figure 13. Chimera hole cut in the flap grid around the solid wall boundary represented by the main airfoil.

- Interpolation boundary on the outer surface of the grid around the store.
- Other boundaries, such as inlet and exit planes.

As seen from the figure, the wing grid overlaps with the NPS in the store grid and *vice versa*. Hence, the Chimera scheme identifies an FBS in both grids and cuts a hole in each grid as shown in Figure 15. It is seen that the wing grid has only FBCs whereas the store grid has both FBCs and IBCs. First, a steady state solution is obtained. Pressure contours of the solution are shown in Plate 1. The free-stream Mach number and pressure are $M_\infty = 0.84$ and $P_\infty = 3.6 \times 10^4$ respectively. The store is now allowed to move under the influence of aerodynamic and body forces. The trajectory of its movement is predicted by the flow solver, using 6DOF analysis, by integrating the forces and moments on the store. All the grid points in the store grid are displaced in accordance with the displacement of the store body. At every new location of the store grid, the relative overlap of the wing grid with the store body (and *vice versa*) changes. The Chimera scheme is seen to cut holes appropriately in both the grids based on the initial specification of the boundary types listed above. This is demonstrated by Figures 16–18, which show the Chimera holes cut in the two grids at three different locations of the store body. Notice that in Figure 18 there is no Chimera hole cut in the store grid. This is due to the fact that at this location, the store grid has been displaced to a location where it does not overlap with the NPS of the wing grid. The automated Chimera scheme perceives this change and acts accordingly without any user intervention. Comparison of CPU time was also made to show the efficacy of the hole-cutting scheme.

For the overset mesh employed in this case, which comprises about 7000 points in the wing grid and 1500 points in the store grid, the hole-cutting algorithm took about a sixth of the time

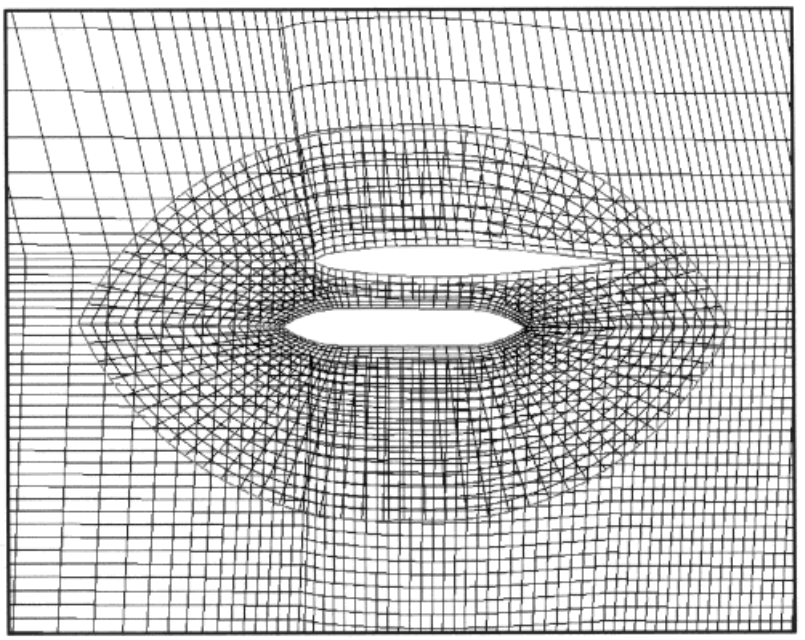

Figure 14. Overset grid system employed for flow simulation around a wing–store configuration.

that it takes for the flow solver to march the solution in all zones by one time step. In a typical unsteady simulation with time steps of the order of $(10)^{-3}$–$(10)^{-4}$ s, relative changes in the grid point locations are rather small between successive time steps. Hence, the hole-cutting algorithm does not need to be invoked at every time step. Thus, the automated Chimera scheme incurs at very little CPU overhead.

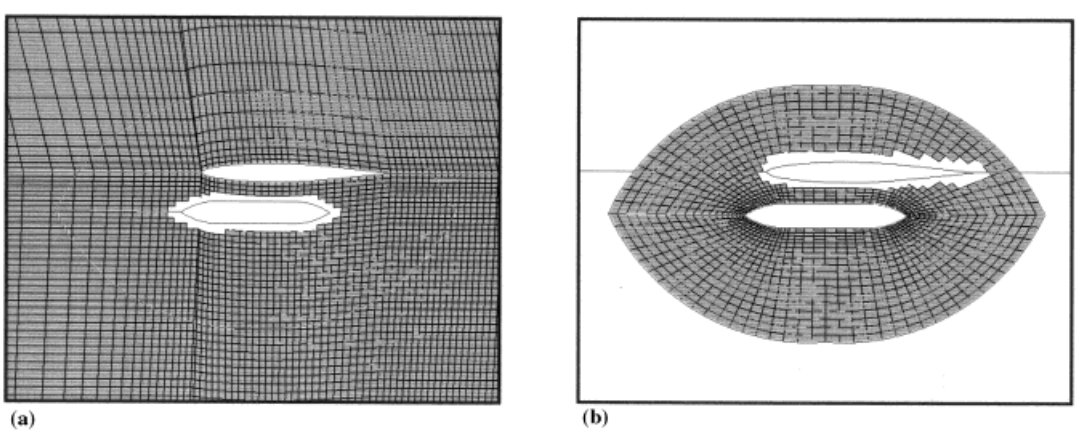

Figure 15. (a) Chimera hole cut in the wing grid around the NPS of the store body and (b) Chimera hole cut in the store grid around the NPS of the wing body.

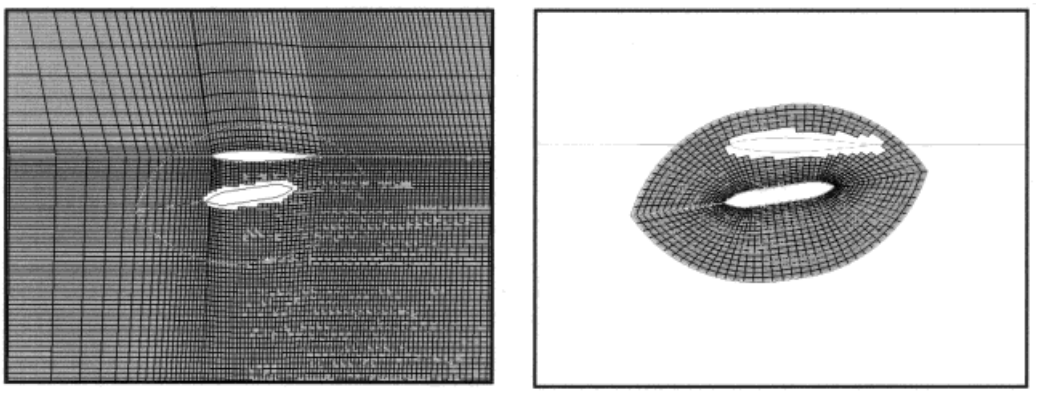

Figure 16. Chimera holes cut in the wing and store grids respectively during the course of an unsteady simulation where the store grid is moving.

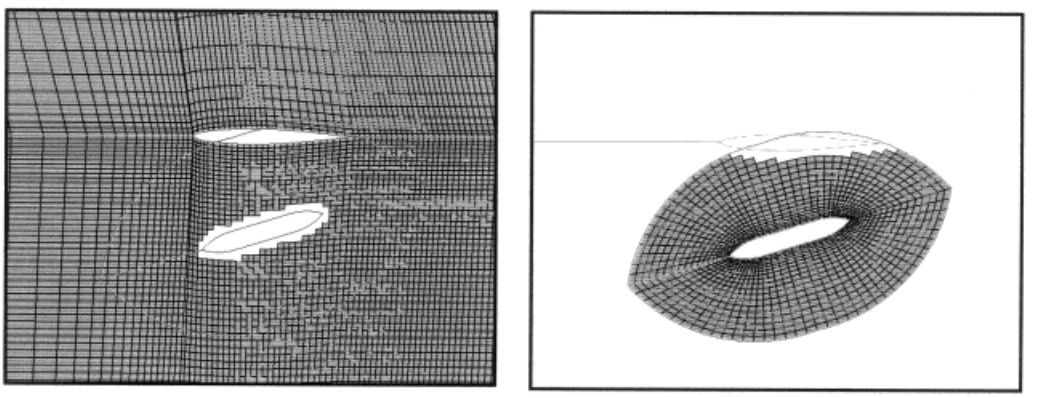

Figure 17. Chimera holes cut in the wing and store grids respectively at a second location of the store grid, which is moving.

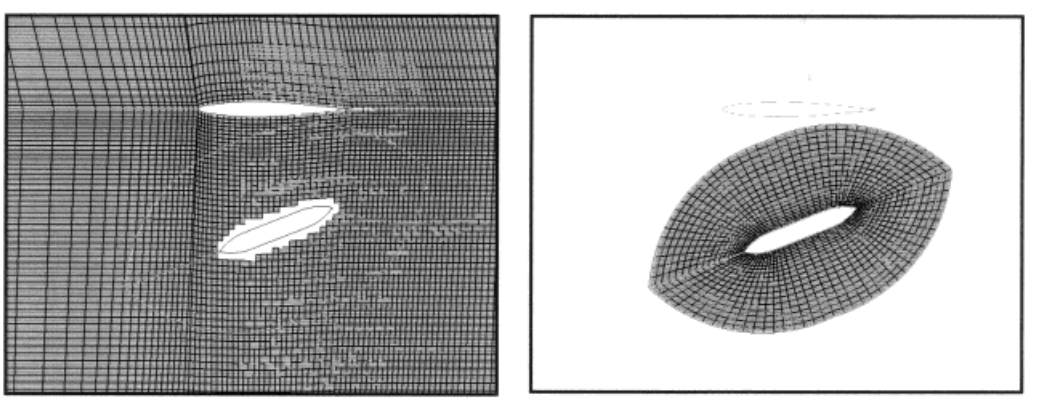

Figure 18. Chimera holes cut in the wing and store grids respectively at a third location of the store grid, which is moving.

The automated Chimera scheme was used for modeling the unsteady flow around a three-dimensional Eglin store separating from a clipped delta wing with an NACA 64A010 airfoil section. A view of the mesh employed for the wing–pylon–store body configuration is shown in Figure 19. The wire mesh represents the overset grid around the store body. The wing–pylon grid comprises about 225 000 points while the store grid has about 15 000 points. The initial hole cutting is depicted in Figure 20, which shows the FBS around the Chimera hole cut in the wing–pylon grid by the NPS of the store grid. Pressure contours on the surface of the wing and the store at different locations of the store drop are shown in Plate 3. The Chimera hole cutting in this case is performed in exactly the same way as was demonstrated in the two-dimensional case without any user intervention.

More applications of the automated Chimera methodology in the three-dimensional case are included in other publications [10,11].

4.3. Three-dimensional missile fin geometry

This case demonstrates the capability of the automated hole-cutting algorithm to tackle three-dimensional problems that involve geometries with complex inter-grid overlap. Plate 2 shows a grid employed for the simulation of gap effects on a missile control fin. The overset mesh employed comprises seven structured grid zones. Plate 4 shows a curvilinear section of the missile grid overlapping with the fin body. Similarly, the grid around the fin body is seen to overlap with the missile surface. In order to use a Chimera scheme for this mesh, hole cutting has to be performed in each grid, blanking the cells that overlap with an NPS in another grid. The problem is further complicated by the fact that the grids around the missile and the fin body are multi-domain grids generated by compositing several zones together. Despite the complexity of the geometry, the user is required to identify only the solid wall boundaries (NPS) and interpolation boundaries (IBS) of the overset mesh. Plate 5 shows the hole cutting performed for this configuration. The figure shows a wire frame mesh around the fin body. This represents the FBS of the hole blanked in the missile grid around the fin body. A similar hole cutting is performed in the fin grid (not shown in the figure) to account for its overlap with the missile surface. This case brings forth the efficacy of the Chimera method developed in the present work.

5. CONCLUSIONS

A fully automated Chimera methodology was presented in this study. The approach provides automated hole cutting and stencil identification necessary to facilitate inter-grid communications in an overset grid system. The methodology was demonstrated to be capable of handling complex flow problems involving multiple bodies. The user input, which in other Chimera schemes is very detailed and time-consuming, was reduced to the minimal task of providing only the boundary conditions that identify solid wall boundaries (NPS) in the overset grids. The search operations associated with the hole-cutting as well as stencil-identification algorithms were optimized to $O(\mathrm{Log}_2 N)$ as opposed to $O(N)$, using state-of-the-art ADT data structures. Furthermore, complete automation of the Chimera scheme was shown to render the

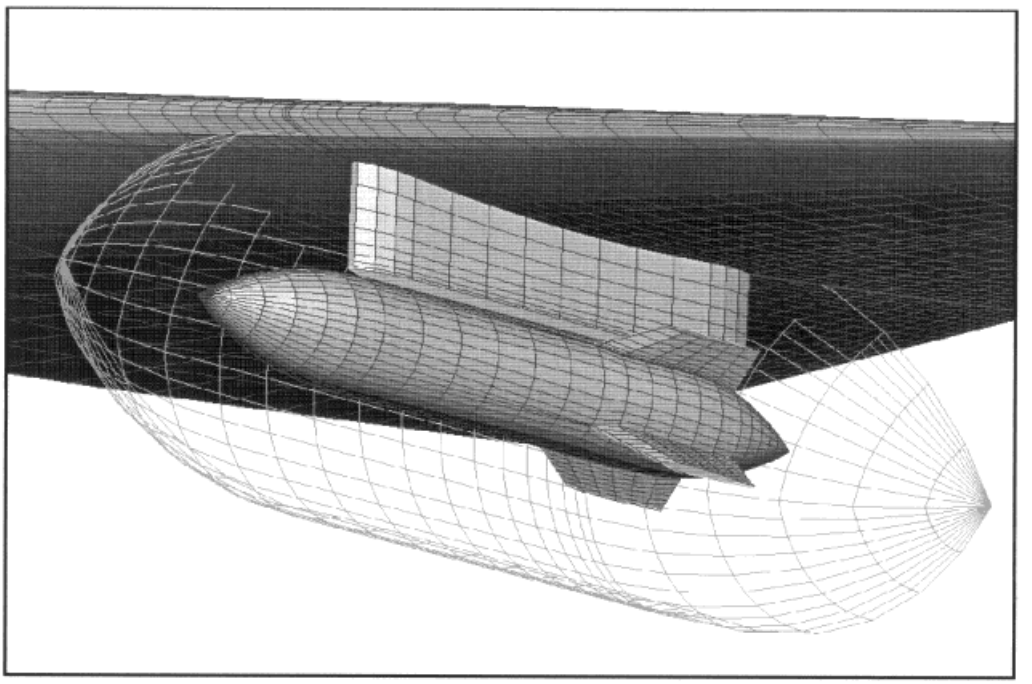

Figure 19. Overset mesh used for three-dimensional Eglin store and the delta wing with a pylon.

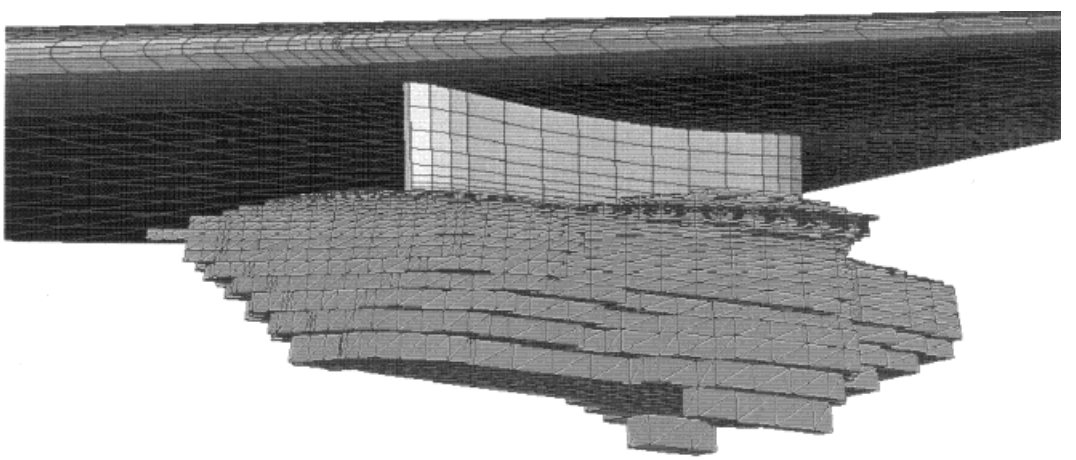

Figure 20. FBS around the Chimera hole in the wing grid cut around the store body.

present approach very suitable for unsteady simulations involving multiple moving bodies. The automated Chimera scheme was also observed to be very efficient timewise, and incurs only very small overheads in CPU time.

REFERENCES

1. Steger JL, Dougherty FC, Benek JA. *A Chimera grid scheme*. ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1982.
2. Benek JA, Steger JL, Dougherty FC. *A flexible grid embedding technique with application to the Euler equations*. AIAA Paper No. 83-1944, 1983.
3. Benek JA, Buning PG, Steger JL. *A 3D Chimera grid embedding technique*. AIAA Paper No. 85-1523, 1985.
4. Benek JA, Tonegan TL, Suhs NE. *Extended Chimera grid embedding scheme with application to viscous flows*. AIAA Paper No. 87-1126, 1987.
5. Meakin RL, Suhs NE. *Unsteady aerodynamic simulation of multiple bodies in relative motion*. AIAA Paper No. 89-1996-CP, 1989.
6. Pearce DG, Atanley SA, Martin FW, Gomex RJ, Beau G, Buning PG. AIAA Paper No. 93-0533, 1993.
7. Meakin RL. AIAA Paper No. 93-3350-CP, 1993.
8. Suhs NE, Tramel RW. *PEGSUS 4.0 User's Manual*. Calspan Corporation/AEDC Operations, 1991.
9. Bonet J, Peraire J. An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering* 1991; **31**: 1–17.
10. Rock SG, Habchi SD. *Application of an automated Chimera methodology to aircraft–canopy/ejection seat separation*. AIAA Paper No. 98-0767, 1998.
11. Hall LH, Parthasarathy V. *Validation of an automated Chimera/6DOF methodology for multiple moving body problems*. AIAA Paper No. 98-0753, 1998.

