Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012

ICCFD7-1201

Advances in Domain Connectivity for Overset Grids Using the X-rays Approach

William M. Chan*, Noah Kim** and Shishir A. Pandya*
Corresponding author: William.M.Chan@nasa.gov
* NASA Ames Research Center, USA; ** Stanford University, USA.

Abstract: Advances in automation and robustness of the X-rays approach to domain connectivity for overset grids are presented. Given the surface definition for each component that makes up a complex configuration, the determination of hole points with appropriate hole boundaries is automatically and efficiently performed. Improvements made to the original X-rays approach for identifying the minimum hole include an automated closure scheme for hole-cutters with open boundaries, automatic determination of grid points to be considered for blanking by each hole-cutter, and an adaptive X-ray map to economically handle components in close proximity. Furthermore, an automated spatially varying offset of the hole boundary from the minimum hole is achieved using a dual wall-distance function and an orphan point removal iteration process. Results using the new scheme are presented for a number of static and relative motion test cases on a variety of aerospace applications.

Keywords: Overset Grids, Domain Connectivity, Hole-Cutting, Grid Generation.

1 Introduction

In recent years, structured overset grid technology has been successfully applied to a wide range of complex aerospace configurations [1, 2, 3, 4, 5, 6]. While the flow solution procedure [7] is highly efficient compared to other grid strategies, the grid generation step still requires a fair amount of expert knowledge and effort to produce good quality grids. The overset grid generation step consists of three parts: surface grid generation, volume grid generation, and domain connectivity. A typical overset grid system consists of near-body curvilinear grids and off-body Cartesian grids. Surface grid generation for the near-body grids is usually the most time consuming step requiring much human interaction and work [8]. Once a set of good quality surface grids has been created, volume grid generation requires little effort and time. Domain connectivity is the last step which involves the identification of grid points that fall inside solid boundaries (hole points), the adjustment of the boundaries of the resulting holes to provide accurate inter-grid communication, and the search for interpolation stencils for the fringe points at the outer and hole boundaries of each grid. The process of identifying hole points in the first part of the domain connectivity step is also referred to as 'hole-cutting'.

A number of schemes have been available to handle the domain connectivity process for structured grids [9, 10, 11, 12, 13]. Each scheme and software requires varying degrees of human interaction and computational time depending on the quality and complexity of the grid system. One of the most effective schemes for hole-cutting is the X-rays approach [10] since its speed allows it to be efficiently utilized in problems involving relative body motion where hole-cutting has to occur at every time step. Examples of relative body motion problems occur in many aerospace applications such as rocket stage separation [14] and store separation [15], debris trajectory analysis [16], and rotor blade systems in rotorcraft [17]. However, the creation of input parameters and files for the X-rays approach requires much human knowledge and effort, and the resulting interpolation boundaries are sometimes not satisfactory (see Section 2 for more details). The main objective of the work described in this paper is to develop algorithms and software to reduce or eliminate the deficiencies of the original X-rays scheme, while maintaining its speed advantage for relative

body motion problems. In order to allow easy interface with other software modules such as a flow solver, code development is performed in a library framework.

In this paper, the procedures described are applied to overset structured grids. However, many ideas discussed are equally applicable to overset unstructured grids. Hole cutting in the current paper is based on the concept of a component, typically defined to be a geometric part of a complex configuration. For example, a complete aircraft may consist of the fuselage, wings, horizontal tails, vertical tail, pylons, and nacelles components. Each near-body grid point in a volume grid system is associated with a component by tracing a grid line from the point to the solid wall surface of the component. A grid point belonging to an off-body grid with no solid walls can be thought of as associated with a phantom component with no walls. A hole-cutter for a component is only allowed to blank grid points associated with other components. Such a hole cutter can be defined from a CAD model that represents the geometric part that has been tesselated into an unstructured surface triangulation, or it can be defined from surface subsets of the structured volume grid system. In the latter case, the hole cutter surfaces may or may not form a closed volume. An X-ray is a discrete representation of a hole cutter which is allowed to cut holes in specific volume grid subsets.

A review of the original X-rays scheme is given in Section 2. Details of the enhancements made to the original scheme are described in the Sections 3, 4, and 5. Results from application of the improved scheme to various aerospace problems are presented in Section 6 with comparisons to the original X-rays method. Summary and conclusions are given in Section 7.

2 Original X-rays Scheme

The X-rays method is a scheme for performing hole cutting for overset grids which involves the identification of grid points where solution of the governing flow equations will not be performed. There are typically two types of such grid points. First, these may be grid points that fall inside solid wall boundaries of the geometry. Second, these may be grid points in regions of the field domain where there is coverage from more than one grid. In such regions, the coarser grid points may be 'blanked' (identified as hole points), while the finest grid remains to resolve the flow.

In the original X-rays method, hole cutters are defined by a collection of surfaces. Such surfaces may be part of the surface grids that represent the solid wall boundaries of geometric components of a complex configuration, or slices of volume grids that bound a hole-cut region. The hole-cutter surfaces can be grid slices from a structured grid system, or unstructured surface triangulations that represent various geometric components of a CAD model of the geometry.

An X-ray is basically a Cartesian map of the hole-cutter generated with the following procedure. First, a bounding box is determined for the hole-cutting surfaces in the X-Y plane, also called the image plane (Figure 1). A uniform Cartesian grid is created on the image plane with spacing Δs equal to the average grid spacing of the component surface grids. A ray is then cast in the Z direction from each point on the image plane, and the pierce points on the surface grids of the component are stored (Figure 1). Given an arbitrary point P in 3-D space, its X and Y coordinates are used to easily look up the appropriate cell in the X-ray image plane. The pierce points in Z are used to determine if P is inside the hole-cutting surface (hole point), or outside (field point). This results in a minimum hole around the surface of the geometry. A user-specified constant offset distance is then used to expand the hole away from the geometry surface.

Once an X-ray map is created for each hole-cutter, the hole-cutting operation is very fast with the Cartesian look-up scheme described above. While this method is efficient for static geometry, its speed is essential in solving problems with components in relative motion. The X-ray map for each moving component is created only once in the static position and orientation (local coordinate system of the component). Hole-cutting during dynamic motion for an arbitrary grid point is performed by a rigid body transformation (3x3 transformation matrix) to the local coordinate system of the moving component where a fast Cartesian look-up is done using the component's X-ray map.

While hole-cutting in the original X-rays scheme is very fast, it also comes with a number of disadvantages. The surfaces that make up each hole-cutter need to be manually specified by taking subsets of a structured grid system, or tesselating each component of the geometry from its CAD representation. If such surfaces do not form a closed volume, further manual effort is needed to generate additional surfaces to close the volume. Since rays are cast in the Z-direction only, open faces that have zero projected area in the Z-direction are

allowed. For example, in a wing-body problem using only a half configuration in the space occupying $y >= 0$, the body is allowed to be open along the $y = 0$ symmetry plane. After the X-ray maps are created, the user has to specify the list of grids that can be cut by each X-ray. This can be error prone and resolution of resulting errors may require running the connectivity code multiple times. Furthermore, a constant offset distance from the minimum hole represented by the X-ray also has to be prescribed so that fringe point interpolation occurs away from the high flow gradients near the wall. Specification of an appropriate offset distance is a process that typically has to be iterated several times until proper overlap is achieved between neighboring grids. Insufficient overlap will result in orphan points, which are fringe points that fail to find an acceptable interpolation stencil. Moreover, a constant offset distance may not always produce the best overlap between neighboring grids, especially in cases where components are in close proximity to each other.

Preliminary work to reduce such disadvantages are described in Ref. [18]. New developments to eliminate the remaining disadvantages of the original X-rays method are presented in this paper. First, component hole-cutter surfaces that do not form a closed body are automatically closed. Second, adaptive secondary X-ray maps are introduced to accurately perform hole cutting near the boundaries of a component. This avoids the need to use extremely fine uniform X-ray maps which results in excessively large X-ray files for cases with components in close proximity. Third, interpolation boundaries are automatically adjusted away from the minimum holes using a wall-distance function followed by further iterations to reduce the number of orphan points, resulting in a spatially varying offset from the minimum hole.

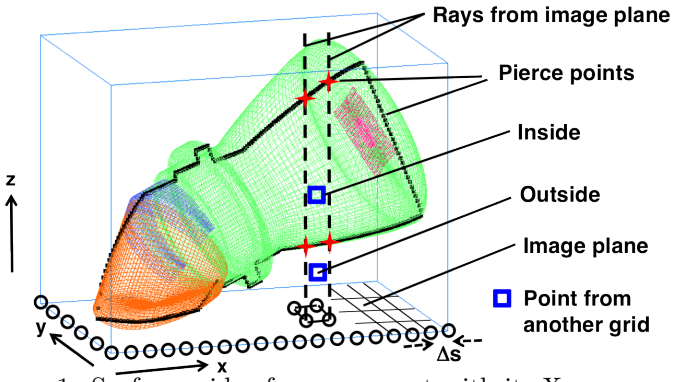


Figure 1: Surface grids of a component with its X-ray map and pierce points (red symbols). Rays are represented by dotted lines.

3 Automated Hole Cutter Closure

In the original X-rays scheme, if the hole-cutting surfaces do not form a closed set, new surfaces have to be constructed manually to close the open boundaries so that a valid X-ray map can be generated. The current work eliminates the manual closure of open boundaries by utilizing existing surface grid cells from other components to complete the closure. Structured grid subsets that make up the hole-cutting surfaces are first converted into unstructured surface triangulations. Open boundaries are easily identified by marking edges that have only one triangle neighbor.

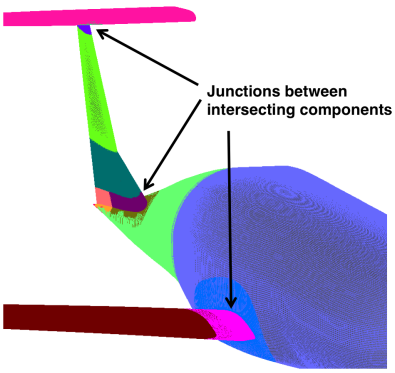
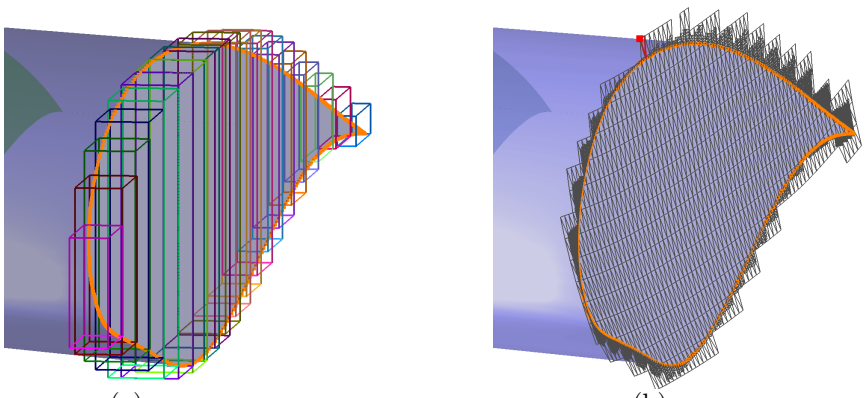


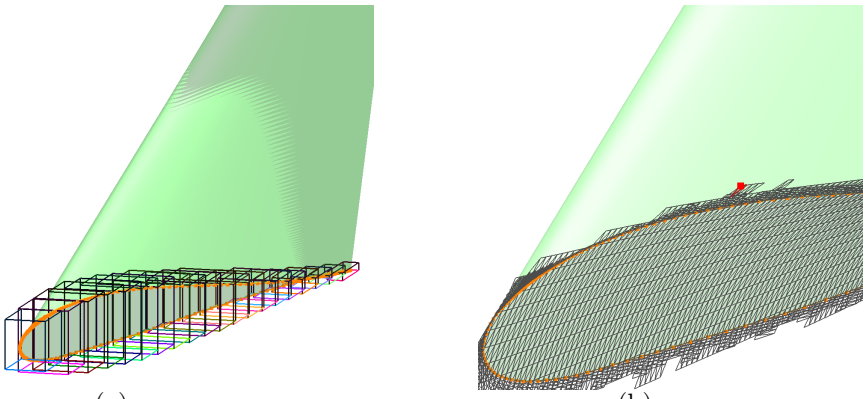
Figure 2: D8 double-bubble aircraft geometry consisting of four components: fuselage, wing, vertical tail, and horizontal tail.

Open boundaries on a component of a complex body can only occur at the junction between itself and other components. For example, Figure 2 shows the junction between a wing and a fuselage. The wing component has an open boundary at the root where it intersects with the fuselage (Figure 3a). Since the fuselage and wing grids are created independently of each other, there are fuselage grid points that fall inside the wing that need to be removed by the hole cutting process. These hole points include fuselage surface grid points that fall inside the wing root (Figure 3b). Surface grid cells from the fuselage that fall within the wing root open boundary are used to close the open boundary. This is accomplished by subdividing the bounding box of the open boundary in



(a) (b)

Figure 3: Junction between a wing and a fuselage. (a) Wing surface and bounding box set over open boundary at wing root used for clipping triangular cells from the fuselage. (b) Triangular cells from fuselage clipped by the bounding box set.



(a) (b)

Figure 4: Junction between a vertical tail and a fuselage. (a) Vertical tail surface and bounding box set over open boundary at tail root used for clipping triangular cells from the fuselage. (b) Triangular cells from fuselage clipped by the bounding box set.

the longest direction into smaller sub-bounding boxes (Figure 3a), and extracting fuselage surface grid cells that intersect these sub-bounding boxes. This process creates sub-bounding boxes that closely follow the 3-D contour of the open boundary and ensures that no new geometry is created in the hole-cutter closure process (Figure 3b). Note that the clipped cells from the fuselage do not have to be point matched with the wing grid points. They merely have to cover the open wing root so that valid pairs of pierce points can be generated by the ray shooting process during X-ray map creation. A more precise closure of the open boundary could be obtained by cutting the clipped triangular cells from the neighboring component (the fuselage in the example shown in Figure 3) with the open boundary edges. For the test cases performed so far, this extra cutting step was not found to be necessary.

Figure 4 shows the open boundary at the vertical tail root where it intersects with the fuselage. Even with this highly three-dimensional open boundary, the scheme is able to clip an appropriate set of triangular cells from the fuselage for hole cutter closure.

4 Minimum Hole Creation Using Adaptive X-rays

In the original X-rays scheme, the X-ray image plane uses a uniformly-spaced set of rays to construct pierce points on a component surface representation. In applications where there are very tight gaps between neighboring components, an accurate representation of the boundary of the components is needed. For

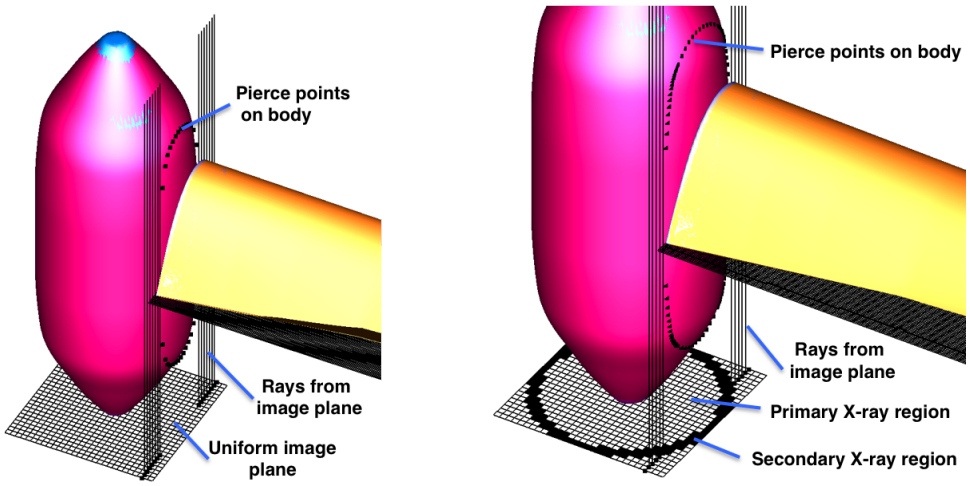


Figure 5: Rotor-hub system with tight gap. (a) Original X-rays with uniform image plane. (b) New X-rays with adaptive image plane.

example, this situation can be found between stages in rocket stage-separation, and between a movable flap and its parent wing or rotor-blade. A very finely-spaced set of X-rays is then needed, resulting in prohibitively large X-ray file sizes.

The current work solves this problem by using two sets of rays: a relatively coarse primary set, and a fine adaptively created secondary set over marked cells in the X-Y image plane of the primary set. The typically 5- to 10-times finer secondary set is automatically introduced only where higher geometric resolution is needed. This occurs where components are in close proximity to each other, and where the geometry surface normal is almost perpendicular to the ray-casting direction.

Close-proximity in the X-Y direction between components is taken care of by marking up refinement cells in the primary X-ray's X-Y image plane around the projected boundaries of the components in the X-Y plane. Such cells have the property that at least one but not all four rays from the corners of the cell pierce the component. Examples of X-Y proximity are shown in the rotor-hub systems in Figures 5 and 7. Close-proximity between components in the Z-direction can be detected using the primary X-ray map of each component. Since this is just a Cartesian map look-up, the process is very fast. For a given component, a cell on the X-Y image plane of its primary X-ray map is marked for refinement if the Z-coordinate of any surface grid point from another component falls within distance ϵ of any Z pierce points on any of the four rays from the corners of the cell (Figure 6). Since refinement of a cell is only needed when the proximity distance between components is of the same order as the X-ray image plane spacing Δs, the threshold ϵ is chosen to be about $2 \times \Delta s$. An example of this situation is found in the abort motors of a candidate space launch system (Figure 8).

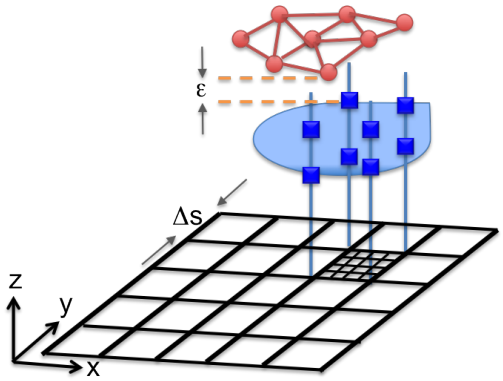


Figure 6: X-ray image plane of blue component with a cell marked for refinement due to proximity of red grid point from surface of red component. Pierce points on the surface of the blue component are shown as blue symbols along the four rays emanating from the four corners of the cell marked for refinement.

The adaptive X-ray is used to identify all grid points that are inside the solid-wall boundaries of all components, resulting in the creation of a minimum hole. Figure 5b shows a two-component system (rotor and hub) with a tight gap. The finer secondary X-rays provide a much more accurate representation of the

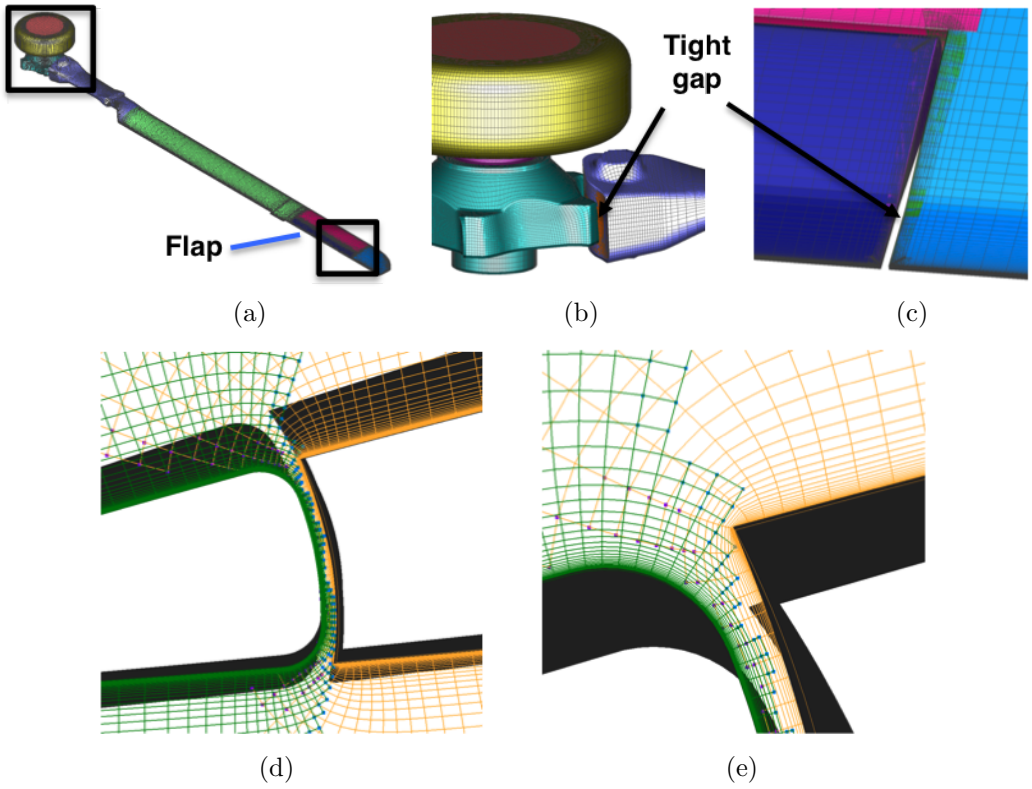


Figure 7: Rotor blade system with flap and tight gaps. (a) Rotor blade, hub and flap geometry. (b) Close-up view of the tight gap between the hub and the blade. (c) Close-up view of the tight gap between the blade and the flap. (d) Overlapping volume grid slices between the blade and the flap (view 1). (e) Overlapping volume grid slices between the blade and the flap (view 2).

boundary of each component. With a 5-times finer resolution in the adapted cells, the adaptive secondary X-ray file size is only a factor of 2 larger than the original. A uniform factor of 5 refinement would have grown the file size by close to 25 times.

Figure 7a shows a rotor blade system consisting of a hub, a blade, and a flap that is allowed to move relative to the blade. A tight gap exists between the hub and the blade, and between the blade and the flap (Figure 7b, c). The blade length and chord are about 195 inches and 10 inches, respectively, but only a 0.03-inch gap exists between the blade and the flap. With the original X-rays scheme, 57 X-rays were manually created for the hub, blade and flap system. A different uniform image plane spacing had to be carefully selected for each X-ray to minimize the total X-ray file size. The smallest image plane spacing was 0.008 inch for the X-rays around the flap region, while the total X-ray file size for all 57 X-rays came to 189MB. With the new adaptive X-ray scheme, only 3 primary X-rays were automatically created with one for the hub, one for the blade, and one for the flap. Different primary X-ray spacings were automatically selected by the scheme. The finest adaptive secondary X-ray spacing was 0.009 for the flap, while the total X-ray file size for the 3 X-rays came to 63MB. Not only does the new scheme provide a much smaller total X-ray file size, automation of the X-ray creation process has significantly reduced the user time required to generate the X-rays. Figures 7d and 7e show that double fringe overlap has been maintained using the new adaptive X-ray scheme and the hole boundary adjustment scheme described in the next section.

Figure 8 shows an abort motor connected to an axisymmetric rocket body with a very tight gap about half way up the motor in the Z-direction. Refinement cells on the X-ray image plane of the abort motor are shown around the boundary of the motor, and in the vicinity of the tight gap. Similarly, refinement cells on the X-ray image plane of the main rocket body are automatically generated in the same vicinity. With the original X-rays scheme, an X-ray file size of 15.3MB was needed to resolve the gap, while the adaptive X-rays file size needed was only 2.9MB.

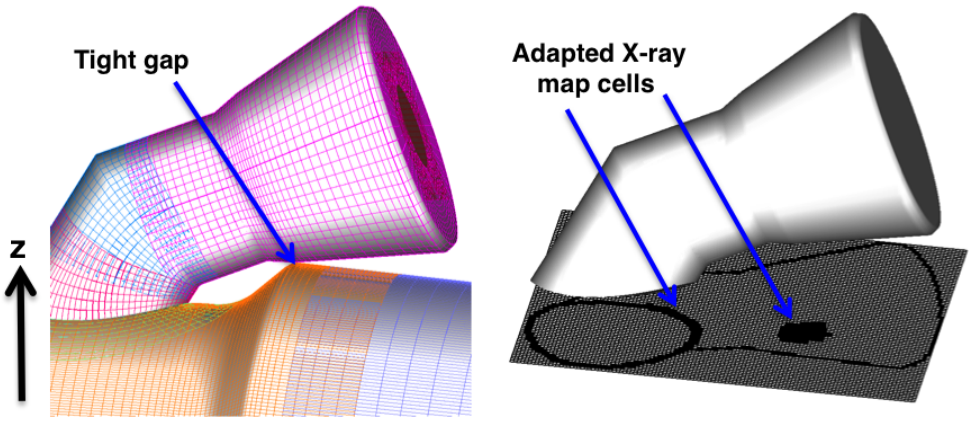


Figure 8: Gap between an abort motor and a main rocket body showing refined X-ray cells in the abort motor primary X-ray map. Refined cells are located around the boundary of the abort motor in the X-Y plane and in the vicinity of the tight gap between the motor and the rocket body.

5 Automated Interpolation Boundary Adjustment

The adaptive X-ray scheme described in the previous section can be used to effectively identify all grid points that fall inside the solid wall boundaries of each defined component. Such grid points are given an iblank value of zero and are called blanked points. This procedure results in an interpolation boundary around the minimum hole which is uniquely defined. In order to avoid poor inter-grid communication between the coarser grid points at the minimum hole boundary and the finer viscous cells next to the solid wall (Figure 9a,b), the interpolation boundary should be retracted away from the minimum hole [8]. At the other extreme from the minimum hole, the interpolation boundary can be retracted so far that there is no overlap between neighboring grids (Figure 9c). Clearly, a wide range of acceptable hole boundary locations exists between the two extremes.

The original X-ray method [10] requires the user to prescribe a constant offset distance from the minimum hole which tends to be tedious and error-prone. Moreover, there are situations where a spatially variable offset is preferred such as where multiple-components are in close-proximity. Other schemes that locate interpolation stencils for all or most grid points in the system [9, 13] use a stencil compatibility measure to select an "optimal" interpolation boundary location. Such schemes provide an accurate mechanism for information transfer between grids, are more automated, but require more computer time than the X-rays approach to find interpolation stencils for all or most grid points in the system.

As far as the authors are aware, a sensitivity study on aerodynamic loads to perturbations on the interpolation boundary has not been rigorously investigated by any research groups. In the current work, it is proposed that if the interpolation boundary is located away from high flow gradient regions such as the boundary layer, aerodynamics loads within acceptable tolerances can be obtained (see Results section below for further discussion). This is the same assumption adopted by the original X-rays scheme where a user-specified constant offset is used to expand the interpolation boundary away from the solid wall boundaries of the flow. The interpolation boundary should be adjusted so that there is sufficient overlap between neighboring grids based on the differencing stencil employed by the flow solver. For example, a 5-point differencing stencil in the flow solver would require two layers of fringe points at the interpolation boundaries. This guarantees that the third point from the interpolation boundary has a full 5-point stencil required for solving the flow equations. One example of the hole boundary location that satisfies the above is shown in Figure 9d.

In the current work, a wall-distance function is used to estimate an appropriate offset to expand the hole boundaries away from the minimum holes. This is followed by further iterations on the hole boundaries to remove orphan points that may be present due to insufficient overlap between grids after the initial hole boundary expansion. Details of these procedures are presented in the following subsections.

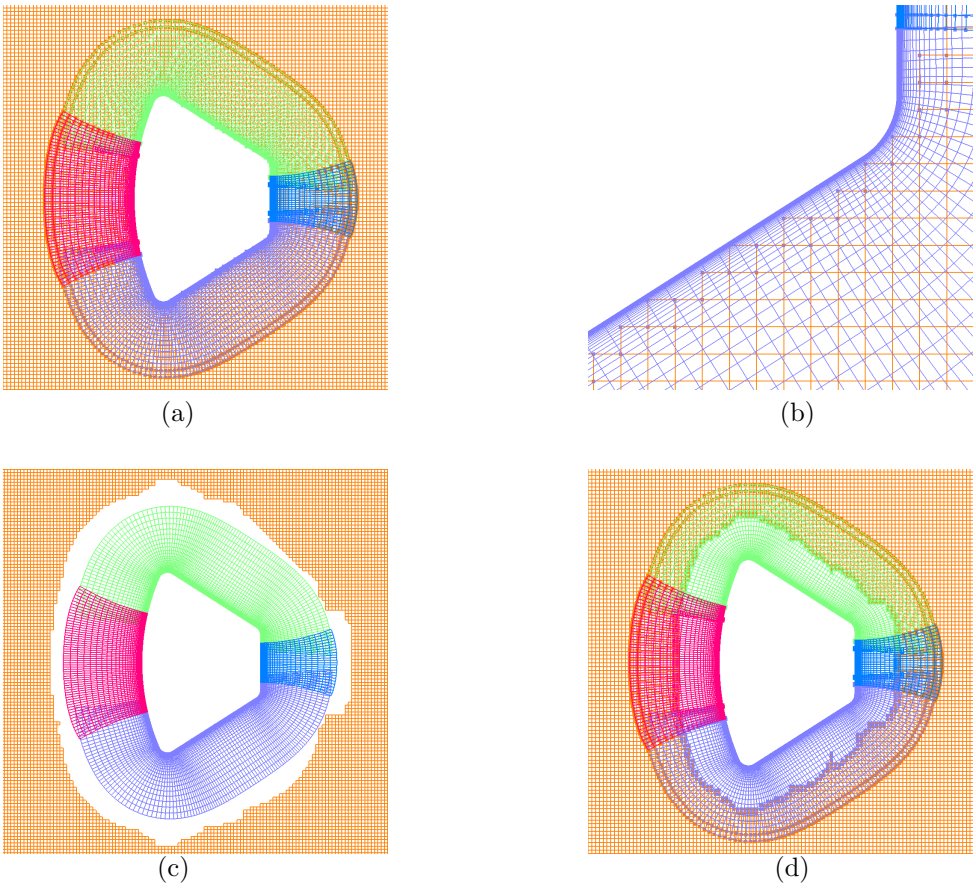
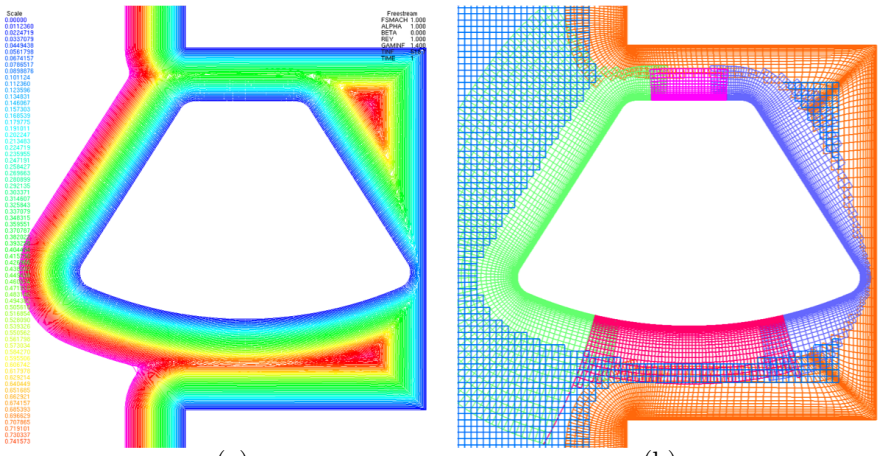


Figure 9: A capsule-in-a-box grid system. (a) Minimum hole with too much overlap (global view). (b) Minimum hole (zoomed in view). (c) Large hole with no overlap. (d) Adjusted hole boundary with proper overlap.

5.1 Initial Interpolation Boundary Estimate

An overset grid system for a complex configuration typically consists of near-body grids that contain a solid wall boundary, and off-body grids (Cartesian or otherwise) that do not contain any solid wall boundary. Recall from the Introduction section that each grid point in the system is associated with a component. Each component hole cutter is allowed to blank points associated with other components, as well as points associated with itself that are away from the geometry surface [18]. Under this scheme, three types of hole boundaries can be created: (1) between a component hole cutter and an off-body grid (Figure 9d), (2) between component hole cutters and near-body grids for disjoint components (Figure 10b), and (3) between component hole cutters and near-body grids of intersecting components (Figure 2). The work in this paper explores the feasibility of using a wall-distance function to create a first estimate of an appropriate hole boundary.

In the flow solution computation process, a standard wall-distance function is frequently computed while solving the turbulence model equations [19, 20]. The standard wall distance D_w in this case is defined to be the distance from a grid point in the volume grid system to the closest solid wall boundary in the configuration. A solid wall boundary may be represented by a structured surface grid, or an unstructured surface triangulation of the wall. An example of this wall distance function is shown in Figure 10a for two disjoint components: a capsule and a cavity. For each grid point on the solid wall surface of these components, a grid line emanates and extends away from the wall. The value of the wall distance D_w increases from zero at the wall to some finite value, and then decreases as the grid line approaches another component. This wall distance function could be used to flag points from the capsule grid that are within distance D_{cut} of the cavity grid, and vice versa. Similarly, D_w for grid points on any off-body grids can be used to trim points on these grids that are within distance D_{cut} from any walls.



(a) (b)

Figure 10: A capsule-cavity system. (a) Wall-distance function for near-body grids (small values in blue, large values in red). (b) Hole boundary estimate using wall-distance function alone.

Usage of the wall distance function D_w for adjusting hole boundaries comes for 'free' since it is already computed for standard turbulence models in the flow solver. The scheme can be used to treat hole boundary types 1 and 2 described above, but unfortunately, it is unable to optimize the type 3 hole boundary. For example, at the junction where a wing intersects a fuselage, a minimum hole cut in the fuselage by the wing cannot be expanded using D_w since the distance to the wall is zero for all points on the surface of the fuselage. In order to expand the minimum hole near the fuselage surface in this case, a dual wall distance function is needed.

This dual wall distance function has to provide two distances: (1) the minimum distance from a grid point to its associated component surface D_n, and (2) the minimum distance from a grid point to any other unassociated component surface D_f. For a grid point in a near-body grid, the grid is connected to the wall of its associated component along a grid line. All other components are defined to be unassociated components to the grid point. For a grid point belonging to an off-body grid (Cartesian or otherwise), there are no associated components. So only the unassociated component wall distance D_f exists for such points and this is the same as D_w described earlier. Also, the standard wall distance D_w can be easily computed by taking the minimum of D_n and D_f. A summary of the various distance measures and notations is given in Table 1.

Table 1: Summary of distance measures used for initial hole boundary estimate.

Measure	Defined	Description
D_w	at each grid point	Closest distance to any component
D_n	at each grid point	Closest distance to associated component
D_f	at each grid point	Closest distance to unassociated component
D_b	for each component	Average distance from component wall to outer boundary in normal direction in volume grid
D_c	for each component	Average distance from intersection line with another component to collar grid outer boundary in surface grid

The minimum hole boundary is expanded using the dual wall distance function as follows. Let $D_b(N)$ be the average distance from the wall of component N to the outer boundary in the normal direction. This is just the average of D_n over the outer boundary points of the component in the normal direction (Figure 11a). Where two components A and B intersect, a collar grid is usually present, and A and B are called the parent components to the collar grid. Let $D_c(A)$ be the average distance from the intersection line between A and B and the outer boundary of the collar grid along the surface of A, and similarly for $D_c(B)$ (Figure 11b).

For a grid point P on a near-body or off-body grid, let S be the unassociated component that contains the closest wall to P. The point is blanked if it is located within about half way between the component

surface and its outer boundary (Figure 11a). This is equivalent to the following test:

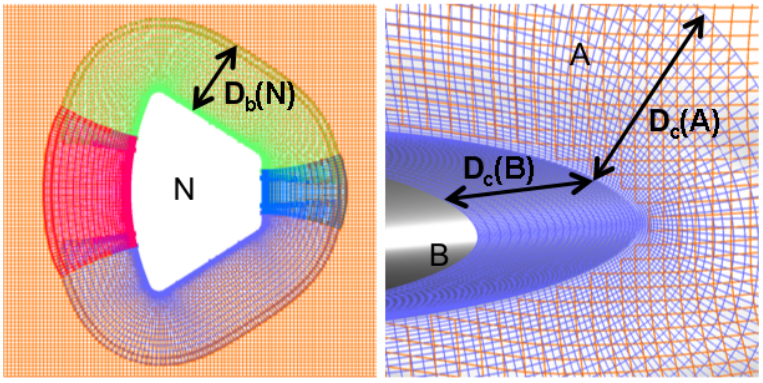
$$D_f(P) < R_b D_b(S) \tag{1}$$

where $R_b = 0.5$ is a parameter. It is adjustable by the user but is usually kept at its default value.

Additionally, for a grid point P on a near-body grid, if the near-body grid does not belong to a component that is a parent to a collar grid, the point P is blanked if Eqn. 1 is satisfied and if it is slightly closer to the wall of an unassociated component than it is to its own associated component. This is equivalent to performing the additional test:

$$D_f(P) < R_n D_n(P) \tag{2}$$

where $R_n = 0.8$ is another parameter. It is also user-adjustable but is usually kept at its default value. A value of R_n equal to 1 would imply equal distance from P to an unassociated component and its associated component. Hence the default value of 0.8 is used to provide some overlap between two near-body grids. For example, for points G and H from the lower near-body grid in Figure 12a, point G is blanked while point H remains a field point.



(a) (b)

Figure 11: Pre-processing steps for minimum hole boundary offset using wall-distance function. (a) Average outer boundary distance for component N. (b) Average outer boundary distance for components A and B along surface.

If the grid point P lies on a near-body grid that belongs to a component A that is a parent of a collar grid, the point P is blanked if Eqn. 1 is satisfied and if it is slightly closer to the other parent of the collar grid than to the outer boundary of the collar grid on component A. This is equivalent to performing the additional test:

$$D_f(P) < R_b D_c(A) \tag{3}$$

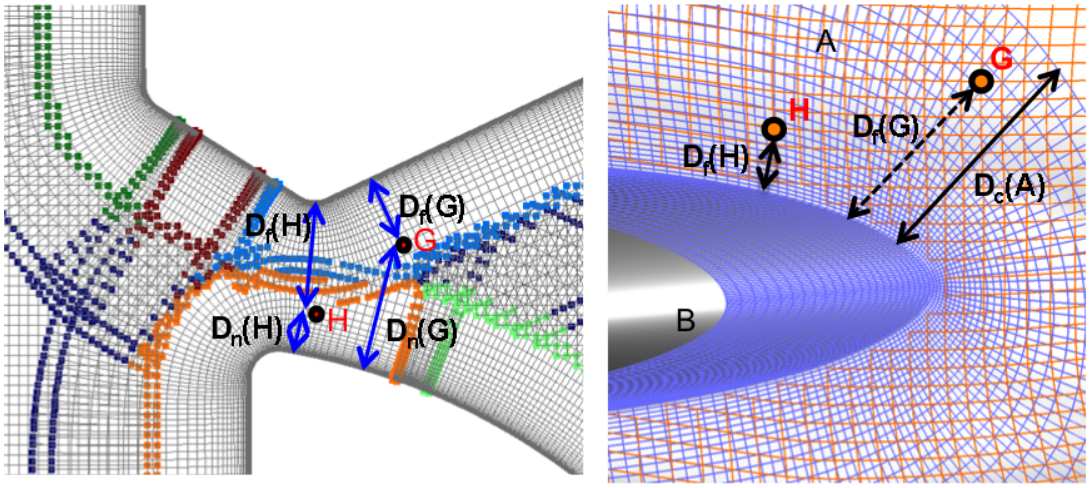
where $R_b = 0.5$ is the same parameter as above. For example, for points G and H from the fuselage (component A) in Figure 12b, point H is blanked while point G remains a field point.

5.2 Approximate Wall-Distance Function

Just like D_w, the dual wall distance function is relatively expensive to compute compared to the rest of the hole-cutting and interpolation stencil search processes. While it is possible to modify D_w in a flow solver to also report D_n and D_f, this paper explores the option of using an approximation to the dual distance function. D_n at a near body grid point is simply computed as the arc length along the grid line from the point to the wall of its associated component. Several different accurate and approximate procedures for computing D_f have been considered in the present work. One of the more promising approximate procedures is outlined below which produces about a two to three orders of magnitude speed-up over an un-optimized exact method for computing D_f.

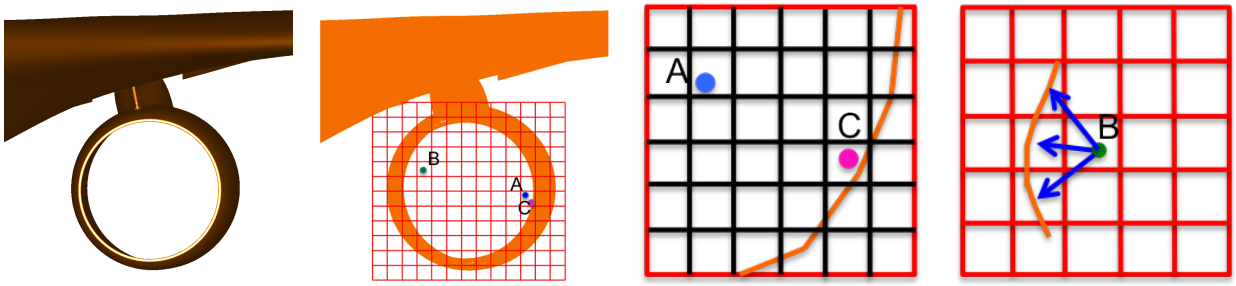
Figure 13a shows the geometry for a nacelle component. A primary 3-D Cartesian map is first built over the triangular elements that represent the component surface geometry. This is shown by the red grid in Figure 13b. A cell in this map either intersects a surface triangulation cell of the component (occupied cell), or it does not intersect any surface triangulation cell of the component (empty cell). Distance from a point to the surface is determined by computing the distance from the point to the centroid of the closest occupied cell, thus avoiding the more expensive closest distance test from a point to the list of triangles in the cell.

In the example shown in Figure 13, D_f for grid points A, B, and C to the nacelle component are computed as follows. First, the primary Cartesian map cell that contains each of these points is easily identified. For grid points A and C, the primary Cartesian-map cell is an occupied cell. In this case, the cell is further



(a) (b)

Figure 12: Heuristic rules for minimum hole boundary offset using wall-distance function. (a) Between disjoint near-body grids. Points G and H are associated with the lower component. (b) Between near-body grids associated with components A and B joined by a collar grid. Points G and H are from component A.



(a) (b) (c) (d)

Figure 13: Dual wall distance function computation for nacelle component. (a) Nacelle geometry. (b) Cartesian map over nacelle component with test points A, B, and C. (c) Secondary refined Cartesian map over Cartesian cell that bounds A and C. (d) Approximate distance computation for point B using distance to centroid of closest cell that contains a wall.

refined into a secondary Cartesian map (Figure 13c). The cells in the secondary map are also classified as occupied or empty cells. Since point C is located in an occupied secondary cell, the approximate distance from point C to a wall is estimated to be the distance from C to the centroid of this cell. For point A, it resides in an empty secondary Cartesian cell. By expanding in index space radially from the secondary Cartesian cell that contains point A, the closest occupied cell is located. The approximate distance from point A to a wall is then estimated to be the distance from A to the centroid of this cell. Since point A is at least one or more Cartesian-map cells away from a wall, this is a fair approximation to the true distance. Similarly, point B is located in a primary empty cell, its distance to a wall is estimated by the same method as point A.

5.3 Interpolation Boundary Adjustments

Once the interpolation boundary has been retracted from high gradient near-wall regions, some fringe points at the new interpolation boundary may not have sufficient overlap with neighboring grids (Figure 14a). Such points are labeled as orphan points and shown as black symbols in Figure 14. Further refinement of the interpolation boundary location is performed with the goal of reducing the number of orphan points. Let J, K, and L be the three index space directions in a structured volume grid. For each orphan point that is not on the grid boundary in the J, K, or L direction, neighboring blanked points that do not belong to

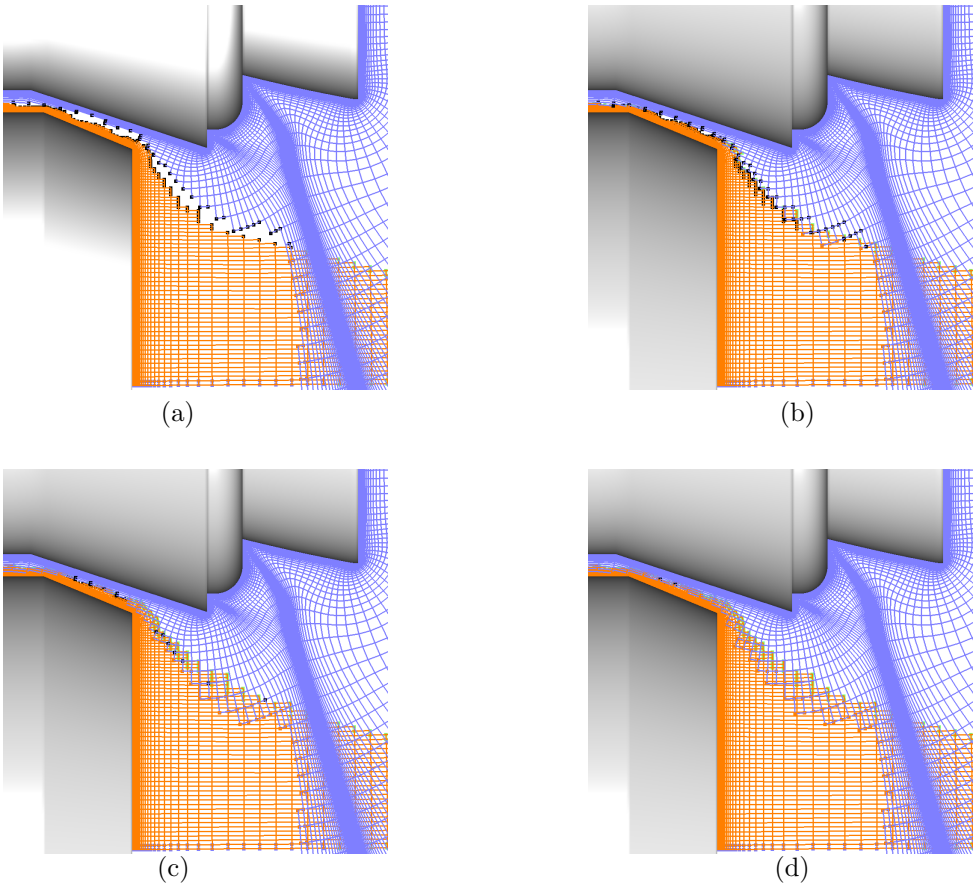


Figure 14: Hole boundary adjustment steps. Orphan points are denoted by black symbols. Fringe points with acceptable interpolation stencils are denoted by colored symbols. (a) Initial hole boundary estimate using wall distance function. (b) Hole boundary after 1 iteration. (c) Hole boundary after 2 iterations. (d) Hole boundary after 3 iterations.

the minimum hole are converted back to a field point. The appropriate number of layers of fringe points at a hole boundary is dependent on the differencing stencil employed by the flow solver. For example, a 5-point differencing stencil is best accompanied by two layers of fringe points at the hole boundaries so that the first solved-point from the hole boundary has a full stencil. For an orphan point belonging to the N^{th} fringe layer, N layers of neighboring blanked points are considered for field point conversion. This procedure pushes the fringe boundary further into the blanked zone which improves grid overlap, which should in turn reduce the number of orphan points. For an orphan point that is on a grid boundary, the fringe boundary cannot be pushed further, but modifications can be made in the grid of the potential donor interpolation cell. If the potential donor cell contains at least one blanked or fringe point, any blanked point in the donor cell or its neighbors that is not in the minimum hole is converted to a field point. Figures 14b, c, d show the adjusted hole boundaries and the reduction in the number of orphan points for the first three steps in the orphan iteration process. Almost all orphan points have been removed by the end of step 3 in this test case.

Iteration of the hole boundary using the orphan point criterion terminates when the number of orphan points has been reduced to the same number as that found at the minimum hole stage (should be zero for properly built overset grids), when the number has reached a constant, or when a pre-defined maximum number of iterations has been reached. For cases presented in this paper, typically less than 10 iterations are needed for the orphan points iteration process to terminate. In some of these cases, the number of orphan points that remained after the iteration was the same number as was found after the minimum hole cut. For some other cases, a very small number of unexplained orphan points remained after the iteration process (less than 1 for every 2 million grid points in the system). Future work will be needed to refine the iteration process to remove these unexplained orphan points.

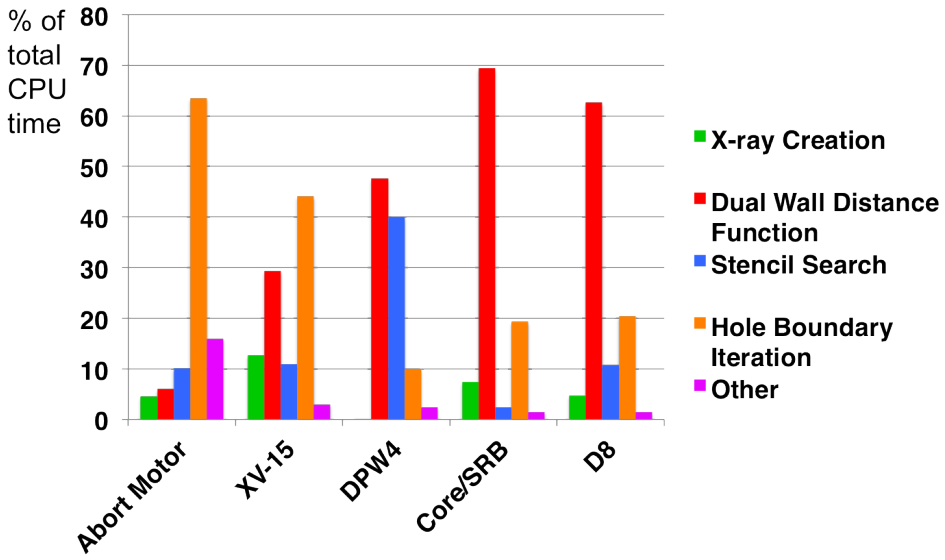


Figure 15: Percentage of total CPU time utilized by various steps of C3P for five 3-D test cases (Abort Motor = isolated abort motor connected to a cylindrical body; XV-15 = three rotor-blades and a hub; DPW4 = Drag Prediction Workshop 4 airplane; Core/SRB = simplified launch vehicle with a core stage and an SRB; D8 = D8 double bubble aircraft).

At the beginning of each hole boundary adjustment iteration, a new set of fringe points is identified. Seeking interpolation stencils for all these fringe points at every iteration step is computationally expensive. Fortunately, the hole boundary perturbation occurs over only a small fraction of the fringe points. A master array is used to store all the interpolation stencils of all fringe points that have been considered so far. At each iteration, only new fringe points that have not been visited would trigger an interpolation stencil search. This technique minimizes the computational cost of the hole boundary adjustment process.

6 Results

The scheme described in this paper has been implemented into a software library called Chimera Components Connectivity Library (C3LIB) for easy interface with other software modules, and with a driver program called Chimera Components Connectivity Program (C3P). Parallel execution has been implemented using OpenMP. However, the code has not been fully optimized. Typical load balancing efficiency is about 60-70% with 8 processors. For the new software to be usable and adopted, the resulting aerodynamic loads should be within acceptable deviations obtained from the original X-rays method. Furthermore, for the new scheme to be practical for relative motion problems, the CPU time required to run the software should be comparable to the original scheme.

Various steps in the process have been timed for a variety of aerospace engineering test cases. The main steps include: adaptive X-ray map creation, dual wall-distance computation, interpolation stencil search, and hole boundary iteration. Figure 15 shows the percentage of the total CPU time utilized by the various steps for five 3-D test cases of varying number of grid points. It is observed that for medium and large size problems, the most expensive step is the computation of the dual wall-distance function, followed by either stencil search or hole boundary iteration. The data also indicates that the X-ray creation process, hole-cutting and I/O are each below about 10% of the total CPU time.

The execution times for two cases using the new scheme with C3P are compared against the original DCF/X-rays scheme in the OVERFLOW flow solver [7]. A desktop Linux workstation was used with 8 OpenMP threads for C3P, and 8 MPI processes for OVERFLOW. The results are summarized in Table 2. Columns 3 and 4 compare OVERFLOW and C3P for performing domain connectivity (hole-cutting and stencil search for OVERFLOW; hole-cutting, stencil search, and hole-boundary adjustment for C3P). While OVERFLOW does not require a distance function for hole-cutting purposes, the time in column 3 includes

the standard wall-distance function computation for comparison purposes. Since C3P is performing more work in computing the dual wall-distance function and iterating the hole boundaries, the CPU time needed is expected to be longer compared to OVERFLOW. For the test cases shown, the additional computational cost is about 40 to 50%.

Table 2: Comparison of execution times for C3P and OVERFLOW for performing domain connectivity (time measured in minutes).

Test case	# Grid points ($\times 10^6$)	OVERFLOW with std. dist. func.	C3P with dual dist. func.	OVERFLOW without std. dist. func.	C3P without dual dist. func.
DPW4	16.8	1.1	1.75 (+54%)	0.8	0.87(+8%)
D8	77.7	14	19.9 (+42%)	11.5	6.7 (-42%)

Table 3: Comparison of approximate human input preparation times for C3P and OVERFLOW for performing domain connectivity. Lower numbers represent typical expert user time, higher numbers represent typical novice user time (DPW4 = Drag Prediction Workshop 4, D8 = D8 Double Bubble Aircraft, LV = Launch Vehicle).

Test case	# Grid points	# Grids	# X-rays	OVERFLOW	C3P	Speed up
DPW4	16.8	17	3	30 - 120 min.	10 - 20 min.	$3\times - 6\times$
D8	77.7	30	4	0.5 - 1.5 days	20 - 30 min.	$36\times - 72\times$
LV	65.4	67	25	1 - 3 days	45 - 90 min.	$32\times - 48\times$

Columns 5 and 6 compare the execution times with the distance function computation time subtracted from the total. This shows that the automated steps in C3P are about as efficient as OVERFLOW or better when the distance function is not counted. The extra computational time incurred by C3P is not as detrimental as it seems from the point of view of the flow solver. Since the standard wall-distance function is needed for the solution of most commonly used turbulence model equations, and since it can be cheaply recovered from the dual wall-distance function (section 5.1), performing domain connectivity with C3P would gain the wall-distance function for free. Further work will be needed to verify if an approximate distance function such as the one described in this paper is adequate for use in the turbulence model equations. Comparing the execution time for OVERFLOW and C3P for domain connectivity when both are using the dual wall-distance function, C3P is more expensive than OVERFLOW by only about 25% for the test cases shown in Table 2.

Even though the new scheme is more expensive, the better quality grid overlap and the human effort saved in preparing the inputs far outweigh the extra CPU time required. Table 3 shows the approximate human input preparation times for C3P and OVERFLOW for several test cases. The lower numbers indicate the approximate time for a typical expert user, while the higher numbers indicate the approximate time for a typical novice user. For large problems, input times have been reduced from days to an hour or two, while the CPU time increase is only on the order of minutes. Further effort in designing faster and more scalable dual distance function algorithms will bring the difference in CPU time even closer.

Other than the test cases already shown in the previous sections, a number of applications of current interest are presented in the following subsections.

6.1 Drag Prediction Workshop Aircraft

The Fourth AIAA Drag Prediction Workshop (DPW4) [21] involved an aircraft geometry with a fuselage, wing and horizontal tail (Figure 16). A vast variety of gridding schemes (structured and unstructured), turbulence models, and flow solvers have been applied to this problem. The simulations were performed at a prescribed lift condition, while drag and pitching moment coefficients were computed. Statistical analysis of the results indicated that one standard deviation in the variation of drag and pitching moment coefficients corresponded to about a 3% and 40% difference from the mean, respectively.

Results of the automated hole-cutting and interpolation boundary adjustment scheme described in this paper are shown in Figure 17 for a coarse version of the grid system containing 16.8 million points. After

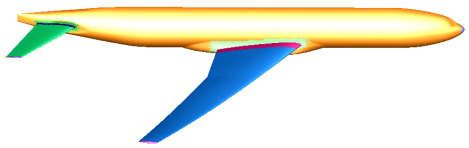


Figure 16: Fourth AIAA Drag Prediction Workshop geometry.

the orphan points iteration process, only 8 orphan points remained. Table 4 shows a comparison of lift , drag, and pitching moment coefficients obtained by the current method with the original X-rays method. All solutions were computed using the OVERFLOW Navier-Stokes flow solver [7]. The results indicate that the differences are well within the spread reported in [21] for the different methods.

Table 4: Comparison of lift, drag, and pitching moment coefficients for DPW4 aircraft with hole boundary locations created by the original and improved X-ray schemes.

	Lift coef.	Drag coef.	Pitching moment coef.
Original X-rays	0.4852	0.02741	-0.02898
Improved X-rays	0.4862	0.02736	-0.02967
Difference	0.2%	0.2%	2.3%

6.2 D8 Double-Bubble Wing-Body-Tail System

The D8 Double-Bubble aircraft from MIT consists of a fuselage, wing, and a horizontal tail mounted on top of a vertical tail. While future work will involve the design of boundary layer ingesting engines mounted on top of the fuselage, current studies on the baseline aircraft have started in the absence of the engines [22]. The geometry of this aircraft is shown in Figure 2. Hole boundaries generated by the scheme described in this paper are shown in Figure 18 for a grid system with 30 grids and 77.7 million points. After the orphan points iteration process, only 27 orphan points remained. A comparison of lift and drag coefficients computed using OVERFLOW is given in Table 5. Again the results show very good agreement between the improved and original X-rays, and the differences are well within deviations that are typically observed between different numerical methods and turbulence models.

Table 5: Comparison of lift and drag coefficients for D8 double bubble aircraft with hole boundary locations created by the original and improved X-ray schemes.

	Lift coef.	Drag coef.	Pitching moment coef.
Original X-rays	0.4560	0.0358	0.1160
Improved X-rays	0.4625	0.0360	0.1187
Difference	1.4%	0.6%	2.3%

6.3 Candidate Space Launch Vehicle

A candidate space launch system configuration is shown in Figure 19a with 65.4 million points and 25 components. It consists of the core stage, two reusable solid rocket boosters (RSRBs), the Multi-Purpose Crew Vehicle and its abort motors, and the various forward- and aft-attachments between the core and the RSRB's. Automatically created hole boundaries in the off-body box grid are shown in Figure 19b. Domain connectivity using C3P took 15.3 minutes on 8 Intel Xeon processors of a Linux workstation.

6.4 Lockheed Martin N+2 Supersonic Aircraft Model

The Lockheed Martin N+2 is a second-generation supersonic aircraft model [23, 24] shown in Figure 20a. The wind tunnel test was performed on a blade mounted half model consisting of a fuselage, wing, diverter

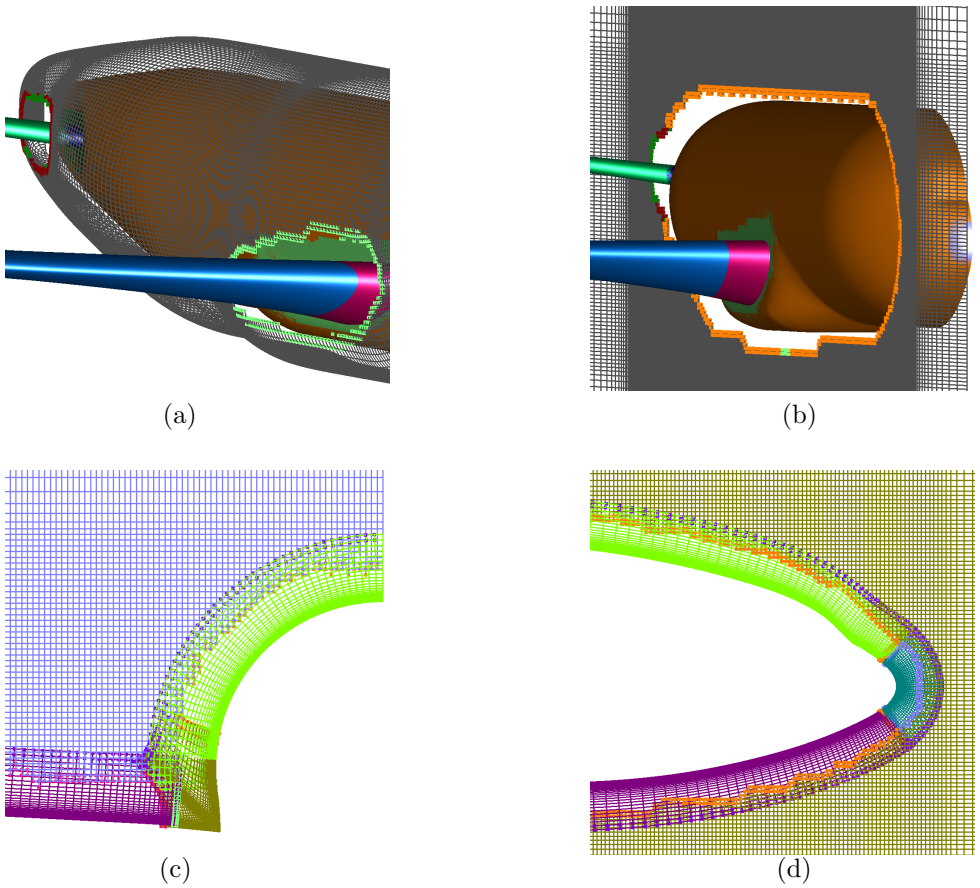


Figure 17: DPW4 grid slices with automatically adjusted hole boundaries. (a) Fuselage grid. (b) Off-body box grid. (c) Constant-x slice over wing-fuselage junction. (d) Constant-y slice over fuselage fore-body.

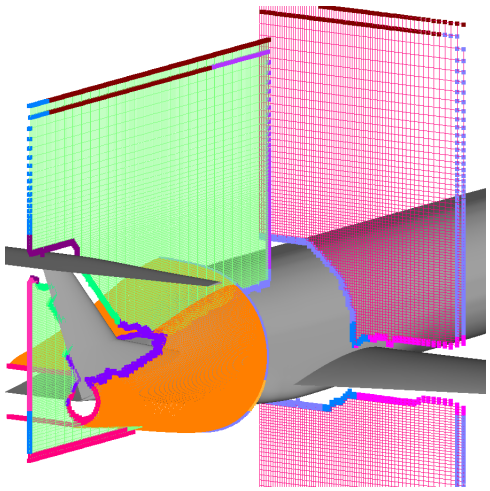
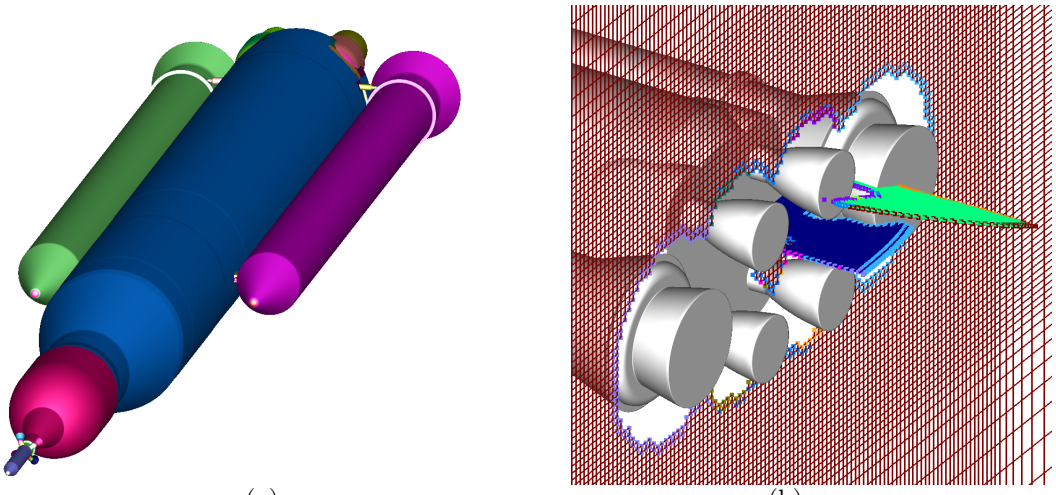


Figure 18: Slices of volume grids for D8 configuration after hole boundary adjustment.



(a) (b)

Figure 19: Candidate space launch system configuration. (a) Surface geometry. (b) Automatically adjusted hole boundary on off-body volume grid slice through base region.

and nacelle on the under side of the wing; and pylon, engine, and vertical tail on the upper side of the wing towards the rear of the fuselage. A preliminary viscous structured overset grid system has been generated for this model with 92 grids and 65 million points. Automatically created hole boundaries using the current scheme on various slices of the volume grids are shown in Figure 20b. Domain connectivity using C3P took 9 minutes on 8 Intel Xeon processors of a Linux workstation.

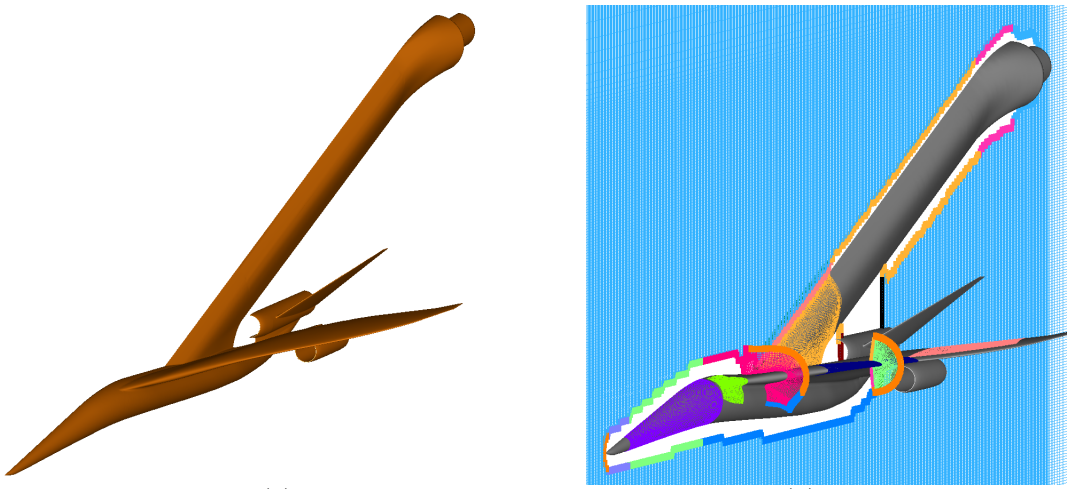
6.5 Three-Element Airfoil System

The new code has also been tested on several unsteady 2-D relative motion problems. These include two tumbling capsules rotating relative to each other in close proximity, a capsule rotating in close proximity within a cavity, and a three-element airfoil system. In this paper, only the last case will be presented. Results from the other two test cases are of similar quality.

Figure 21 shows the three-element grid system with 100,000 grid points consisting of a wing, slat and flap. A simulation of the relative motion of the three elements during deployment and retraction of the slat and flap was performed with prescribed dynamics. Since the purpose of the test case was to assess the quality of the new software for performing domain connectivity, the flow solver was not invoked. The automatic hole-cutting and hole boundary adjustment scheme described in this paper were applied at every time step during the unsteady motion. Execution time for domain connectivity for one time step was 11 CPU seconds on a single processor of a Linux workstation. Slices of the grids with both time and spatially varying hole boundaries are shown in Figure 22. It can be seen that at each instance in time, double fringe interpolation is maintained throughout the domain with spatially varying offsets from the minimum hole, i.e., the offsets are smaller where two components are in close proximity, and larger where the components are further apart. For most of the time steps, the orphan point iteration converged to zero orphans. In instances where orphan points remained, the total stayed below 10.

7 Summary and Conclusions

New developments are presented which significantly reduce the manual effort, expertise, and time needed to perform hole-cutting for complex overset grid systems using the X-rays method. Open boundaries in the component hole-cutter surfaces are automatically closed using existing surface grid cells clipped by a conforming bounding box set over the open boundaries. Tight gaps between components are efficiently resolved using an adaptive X-ray map which avoids the need to build an expensive uniformly fine X-ray map over the entire component. A spatially-variable hole boundary offset estimate is generated automatically



(a) (b)

Figure 20: Lockheed Martin N+2 supersonic aircraft model. (a) Surface geometry. (b) Volume grid slices with automatically adjusted hole boundaries.

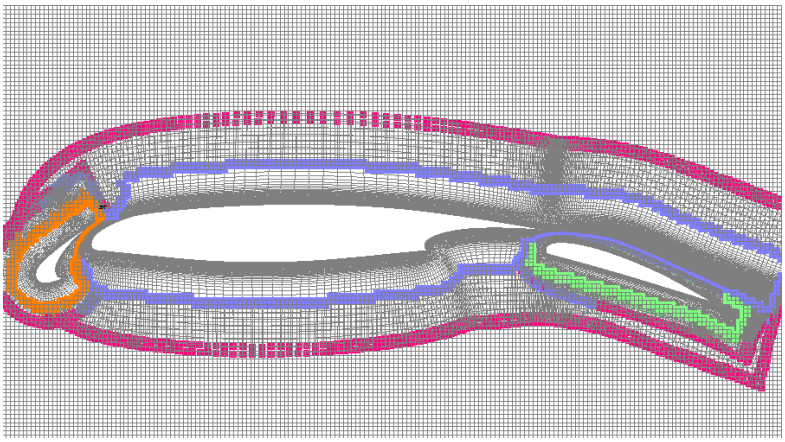


Figure 21: Three-element airfoil grid system.

using a dual wall-distance function which displaces the hole boundary away from the minimum hole. An iterative scheme is then employed to reduce or eliminate any orphan points that remain. The result produces hole boundaries that are away from the high gradient boundary layer regions near the wall.

Hole boundaries created using the new scheme are demonstrated on a range of test cases from a variety of aerospace applications. Flow solutions from selected cases show that the automatically created hole boundaries result in comparable forces and moments to cases where the hole boundaries have been manually created by the original X-rays method. The test cases suggest that the new scheme is computationally more expensive compared to the original X-rays method by about 50%, mainly due to the costly dual wall-distance function computation. However, the time savings in human effort and expertise is significant. As a result, the total time to perform domain connectivity (human plus computational) is greatly reduced. Further investigations into more efficient dual wall-distance function computation algorithms will bring the current scheme even closer in computational cost to the original X-rays method.

The new scheme automatically generates a spatially variable offset from the minimum hole which typically results in higher quality grid overlap compared to the original scheme. This automatic offset is particularly valuable in relative motion simulations. With the original scheme, a constant offset is usually inadequate in producing good grid overlap during the relative motion. Moreover, it is impractical to specify a different constant offset at each time step manually.

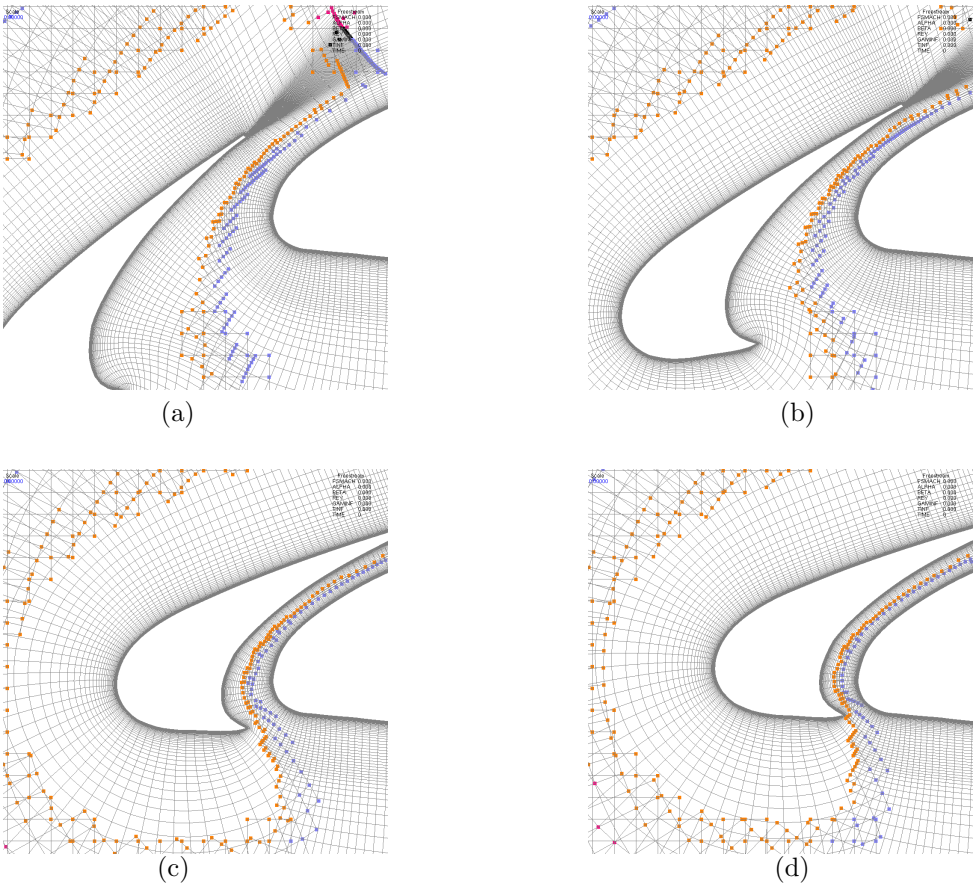


Figure 22: Three-element airfoil grids with automatically adjusted hole boundaries zoomed into slat and main element region at different instances in time during an unsteady relative motion simulation.

8 Acknowledgements

Funding for this work has been provided by NASA's Space Launch Systems and Fundamental Aeronautics Programs. The authors are grateful to Dr. Jeffrey Housman of NASA Ames Research Center for insightful discussions on various hole-cutting and wall-distance function computation algorithms, and to Dr. Thomas Pulliam of NASA Ames Research Center for providing the OVERFLOW solutions of the DPW4 cases. Assistance in grid generation for the Lockheed Martin N+2 model has been provided by James Jensen of Arizona State University.

References

[1] Slotnick, J. P., Kandula, M. and Buning, P. G. Navier-Stokes Simulation of the Space Shuttle Launch Vehicle Flight Transonic Flowfield Using a Large Scale Chimera Grid System. AIAA Paper 1994–1860, 1994.

[2] Kiris, C. C., Kwak, D., Chan, W. M. and Housman, J. A. High-Fidelity Simulations of Unsteady Flow Through Turbopumps and Flowliners. *Computers & Fluids*, Vol. 37, pp. 536-546, 2008.

[3] Bhagwat, M., Dimanlig, A., Saberi H., Meadowcroft, E., Panda, B. and Strawn, R. CFD/CSD Coupled Trim Solution of the Dual-Rotor CH-47 Helicopter Including Fuselage Modeling. Proceedings of the American Helicopters Society Aeromechanics Specialist's Conference, San Francisco, 2008.

[4] Kiris, C., Housman, J., Gusman, M., Chan, W. and Kwak, D. . Time-Accurate Computational Analysis of Ignition Overpressure in the Flame Trench. *Computational Fluid Dynamics Review*, Eds. Hafez, Oshima, Kwak, Publisher: World Scientific, 2010.

[5] N. M. Chaderjian and J. U. Ahmad. Detached Eddy Simulation of the UH-60 Rotor Wake Using

Adaptive Mesh Refinement. Proceedings of the American Helicopter Society 68th Annual Forum, Fort Worth, TX, 2012, May 2012.

[6] Chan, W. M. Overset Grid Technology Development at NASA Ames Research Center. *Computers & Fluids*, Vol. 38, pp. 496-503, 2009.

[7] Nichols, R. H., Tramel, R. W. and Buning, P. G. Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications. AIAA Paper 2006–2824, 2006.

[8] Chan, W. M., Gomez, R. J., Rogers, S. E., and Buning, P. G. Best Practices in Overset Grid Generation. AIAA Paper 2002–3191, 2002.

[9] Rogers, S. E., Suhs, N. E. and Dietz, W. E. PEGASUS5 : An Automated Pre-Processor for Overset-Grid CFD. *AIAA J.*, Vol. 41, No. 6, pp. 1037-1045, December 2003.

[10] Meakin, R. L. Object X-rays for Cutting Holes in Composite Overset Structured Grids. AIAA Paper 2001–2537, 2001.

[11] Noack, R. W. Suggar++: An Improved General Overset Grid Assembly Capability. AIAA Paper 2009–3992, 2009.

[12] Henshaw, W. D. Ogen: An Overlapping Grid Generator for Overture. Research Report UCRL-MA-132237, Lawrence Livermore National Laboratory, 1998.

[13] Sitaraman, J., Floros, M., Wissink, A. and Potsdam, M. Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids. *J. Comp. Phys.*, Vol. 229, pp. 4703-4723, March 2010.

[14] Klopfer, G. H., Kless, J. E., Lee, H. C., Onufer, J. T., Pandya, S. and Chan, W. M. Validation of Overflow for Computing Plume Effects during the Ares I Stage Separation Process. AIAA Paper 2011–170, 2011.

[15] Meakin, R. L. Computations of the Unsteady Flow About a Generic Wing/Pylon/Finned-Store Configuration. AIAA Paper 1992–4568, 1992.

[16] Gomez, R. J., Vicker, D., Rogers, S. E., Aftosmis, M. J., Chan, W. M., Meakin, R. L. and Murman, S. STS-107 Investigation Ascent CFD Support. AIAA Paper 2004–2226, 2004.

[17] Dimanlig, A., Meadowcroft, E., Strawn, R. and Potsdam, M. Computational Modeling of the CH-47 Helicopter in Hover. American Helicopter Society 63rd Annual Forum, May 2007.

[18] Kim, N. and Chan, W. M. Automation of Hole-Cutting for Overset Grids Using the X-rays Approach. AIAA Paper 2011–3052, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 27-30, 2011.

[19] Spalart, P. R., Allmaras, S. R. A One-Equation Turbulence Model for Aerodynamic Flows. AIAA Paper 1992–0439, 1992.

[20] Menter, F. R. Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows. AIAA Paper 1993–2906, 1993.

[21] Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D. W., Brodersen, O. P., Eisfeld, B., Crippa, S., Wahls, R. A., Morrison, J. H. Mavriplis, D. J., Murayama, M. Summary of the Fourth AIAA CFD Drag Prediction Workshop. AIAA Paper 2010–4547, 2010.

[22] Pandya, S. A. External Aerodynamics Simulations on the D8 Double-Bubble Aircraft Design. Paper ICCFD7-4304 (http://www.iccfd.org/iccfd7), Seventh International Conference on Computational Fluid Dynamics, Big Island, HI, 2012.

[23] Cliff, S., Elmiligui, A., Aftosmis, M., Thomas, S., Morgenstern, J., and Durston, D. Design of Rail Instrumentation for Wind Tunnel Sonic Boom Measurements and Computational-Experimental Comparisons. Paper ICCFD7-2006 (http://www.iccfd.org/iccfd7), Seventh International Conference on Computational Fluid Dynamics, Big Island, HI, 2012.

[24] Morgenstern, J. M., Buonanno M. and Norstrud, N. N+2 Low Boom Wind Tunnel Model Design and Validation. AIAA Paper 2012–3217, 2012.