Overset Methods, Inc.
A Nonprofit Public Benefit Corporation

OMI 04-92-2

12 p

PROGRESS REPORT
for NASA GRANT 2-783
SUBMITTED TO

NASA Ames Research Center
Computational Technology Branch

Point of Contact:
Dr. Pieter Buning

By

Overset Methods, Inc.
262 Marich Way
Los Altos, CA 94022

On Automating Domain Connectivity for Overset Grids

Principal Investigator: Dr. Ing-Tsau Chiu

June 30, 1994

On Automating Domain Connectivity for Overset Grids

Ing-Tsau Chiu[1]
Robert L. Meakin[2]
Overset Methods, Inc.
NASA-Ames Research Center
Mail Stop T045-2
Moffett Field, California 94035

ABSTRACT

An alternative method for domain connectivity among systems of overset grids is presented. Reference uniform Cartesian systems of points are used to achieve highly efficient domain connectivity, and form the basis for a future fully automated system. The Cartesian systems are used to approximate body surfaces and to map the computational space of component grids. By exploiting the characteristics of Cartesian systems, Chimera type hole-cutting and identification of donor elements for intergrid boundary points can be carried out very efficiently. The method is tested for a range of geometrically complex multiple-body overset grid systems.

1. INTRODUCTION

The ability to accurately simulate unsteady flowfields about geometrically complex and moving component configurations is becoming increasingly important in the analysis of modern aircraft and launch vehicles. Although significant progress has been made in recent years to apply mature computational methods to this class of problems, there are still obstacles which prevent computational fluid dynamics from making more of a direct impact on the design process. Currently available software for unsteady multiple body aerodynamics is very complex, and requires a large amount of human interaction and expertise. This, combined with limited computational capacity, greatly restricts the degree to which such problems can be studied. The primary objective of this research is to achieve algorithmic improvements which not only reduce computational demands associated with unsteady multiple body aerodynamics, but significantly reduce the corresponding demands on human resources.

Geometrically complex problems are often addressed via an overset grid approach. Geometrically complex domains are decomposed into a number of much simpler overlapping sub-domains. The approach simplifies grid generation problems, since each component can be generated independently and grid boundaries are not required to match neighboring grid in any special way. For the same reasons, an overset grid approach can be applied to problems involving motion between vehicle component parts without any additional algorithmic complications. Moving body computations have been carried out time-accurately in three-dimensions for, among others, the separation sequences of the Space Shuttle's solid rocket boosters [1,2], and aircraft store separation sequences [1,3,4]. The approach has also been successfully applied to many non-aerodynamic problems ranging from applications in biomedical fluid mechanics [5] to environmental flow simulations [6].

The price that must be paid for the geometric and computational freedoms provided by an overset grid approach lies in the need to facilitate intergrid communication. The intergrid communication process is simply the interpolation of needed intergrid boundary conditions from solutions in the overlap region of neighboring grid systems. Intergrid boundaries are the outer boundaries of minor grids, and the boundaries around holes created by neighboring body components. For example, the outer boundary points of the airfoil grid illustrated in Figure 1 are intergrid boundary points. Corresponding boundary conditions must be

---

1. Staff Scientist, Member AIAA
2. Staff Scientist, Member AIAA

interpolated from the overlap region of neighboring grid systems (background grid in this case). Hence, a generalized procedure for identifying intergrid boundary points and suitable donors for the required interpolations is needed. Algorithms for performing this task exists [7,8]. Recently, an entirely new approach to the intergrid communication problem has been set forth [9] in the code "DCF3D", and is particularly well suited to moving body problems.

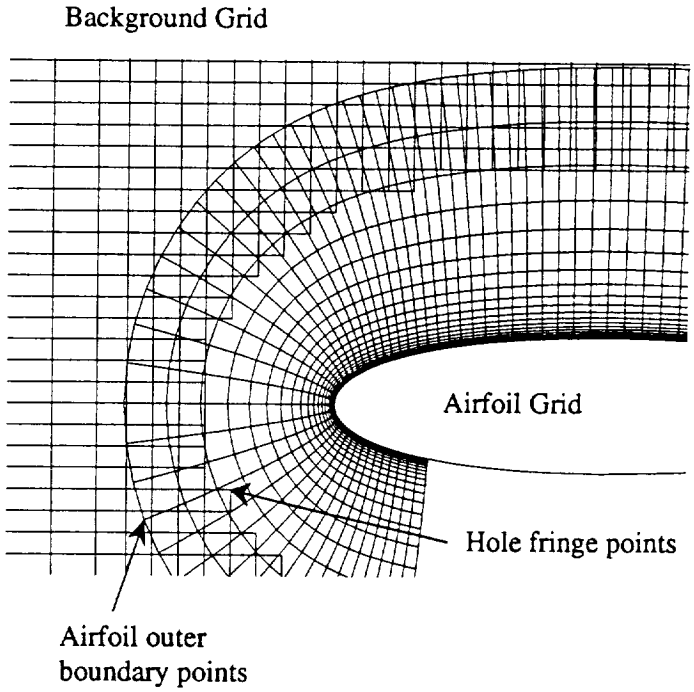


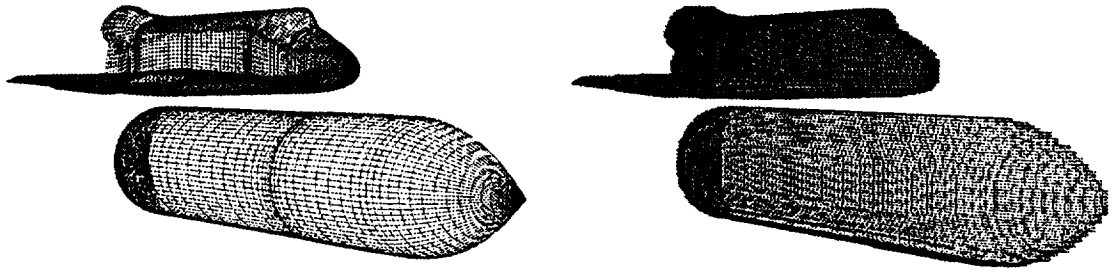
Figure 1. An example of intergrid boundary points.

The most significant advance has been the development of "hole-map" technology. Hole-maps replace DCF3D's method of hole-cutting via analytic shapes, because of its potential for automation and comparable computational efficiency.

2. METHODOLOGY OF HOLE-MAP

Hole-map technology is based on an idea of the late Professor Joseph Steger that takes advantage of the same search-by-truncation incentives that exists in the inverse-maps employed in DCF3D. Given a system of overset grids, and knowing something of the topology and flow boundary conditions, it is possible to generate approximate hole surfaces associated with each component grid. For example, Figure 2(a) illustrates the no-slip surfaces of the space shuttle external tank and orbiter grid systems. Figure 2(b) illustrates an approximation of the same surfaces defined with respect to a uniform Cartesian system of points. The approximate surfaces shown in Figure 2(b) can be used to carry out inside/outside tests for the determination of IGBPs (intergrid boundary points) far more efficiently than the PEGASUS style tests associated with the actual hole surfaces. Given the X,Y,Z coordinate of a point in the orbiter, for example, the position of that point within the external tank hole-map can be identified by simple truncation. Once this is known, the field or hole status of the X,Y,Z point in question is determined by the corresponding status of the hole-map element that bounds it. Following details the hole cutting procedure implemented in DCF3D

utilizing the hole-map technology.



(a) True hole surface. (b) Approximate hole surface.

Figure 2. Comparison of true hole surface and approximate hole surface.

1. Generate a Cartesian box bounding given hole surfaces.
   - Size of the Cartesian box is determined by the box (min/max box) bounding minimum and maximum values of hole surfaces coordinates in X, Y and Z directions.
   - Resolution of Cartesian box is based on the resolution of the given hole surfaces.
   - Hole surfaces can be given as ranges of grid index (J,K,L) or set of offset distances from no-slip surfaces. The offset distance can be given as a global parameter for all the holes or a local parameter for a particular hole.
2. Find hole boundary in the Cartesian box – Hole boundary in the Cartesian box are determined by bounding each quadrilateral surface patch with a min/max box and marked all the cells (IBLANK=0) in this min/max box as part of hole boundary (see Figure 3). To maintain the fidelity to the original geometry, each quadrilateral surface patch is usually divided into smaller patches to avoid overly marking cells as hole boundary. Figure 4(a) and 4(b) clearly illustrate that by dividing a surface patch into smaller patches, the number of extra cells marked as hole boundary is reduced.
3. Close the surfaces bounded by open boundary curves.
   - Find the open boundary curves by eliminating those curve segments that lie on other surfaces. Figure 5 shows the original surface patches and figure 6(a) presents the open boundary curves found.
   - Order the open boundary curves to form one or more continuous curves. This step is necessary since the surface patches may be given in arbitrary order.
   - Group open boundary curves into segments for triangulation. Note that each segment is projected into a 2-D plane and checked for intersection before performing the triangulation. An alternative projection plane is chosen if intersection is found. Delaunay triangulation was implemented to close the surface bounded by open boundary curves. Figure 6(b) shows that the eleven open boundary curves are grouped into three continuous segments for triangulation. Figure 7(a) and 7(b) shows the triangulation for symmetry plane of V-22 tiltrotor geometry and backend of the orbiter. It is obvious that the Delaunay triangulation did very well in maintaining the thin trailing edge for the orbiter.
   - Mark the cells in the Cartesian box occupied by the triangles resulted from Delaunay triangulation as hole boundary.

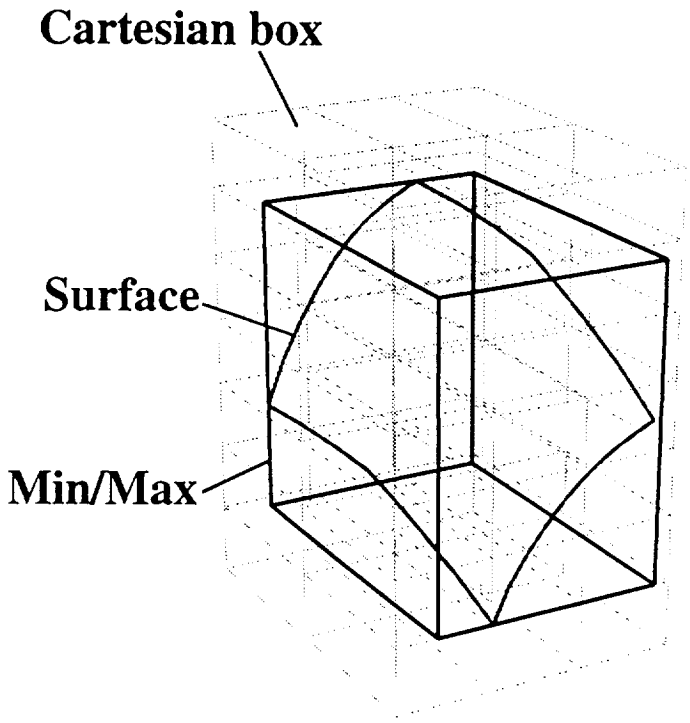
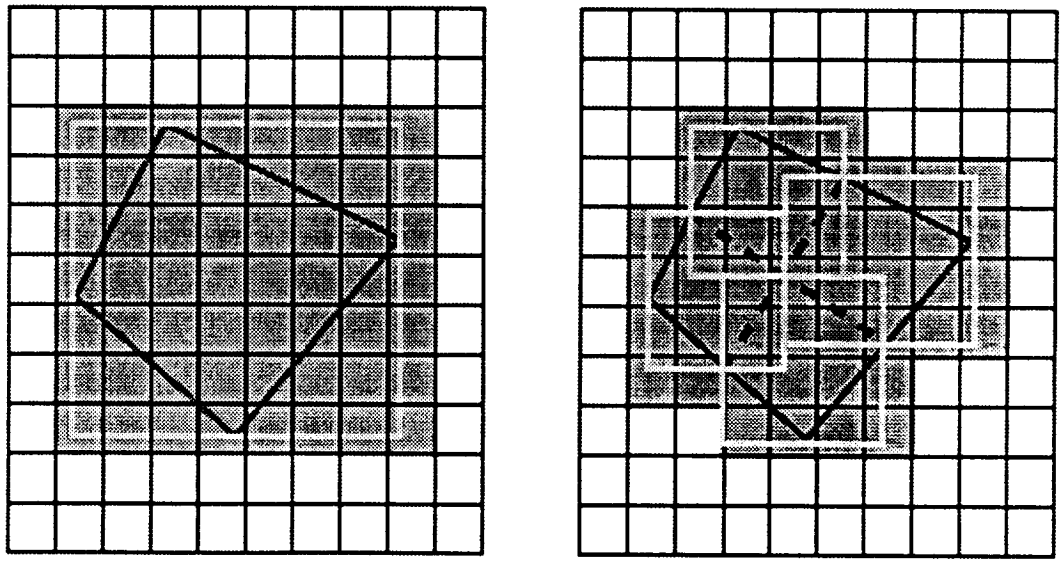


Figure 3. Method of finding hole boundary in a Cartesian box.



(a) Original surface patch. (b) Sub-division of surface patch.

Figure 4. Comparison of single surface patch and multiple surface patches approximation.

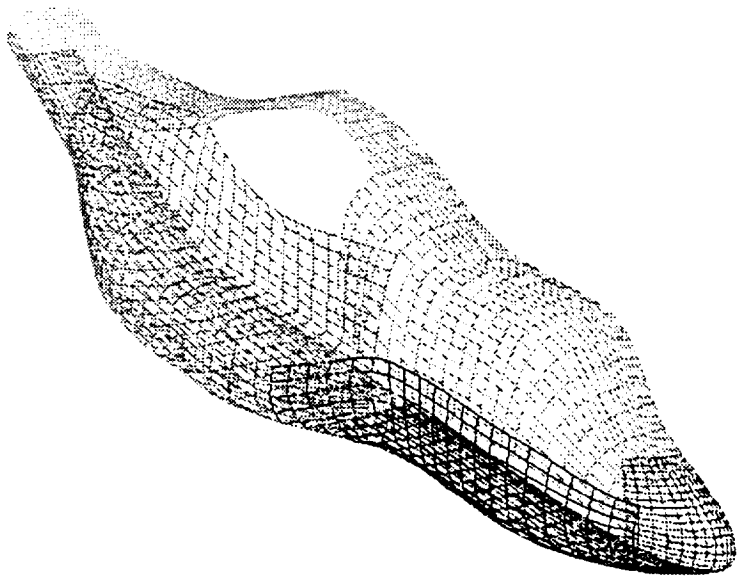
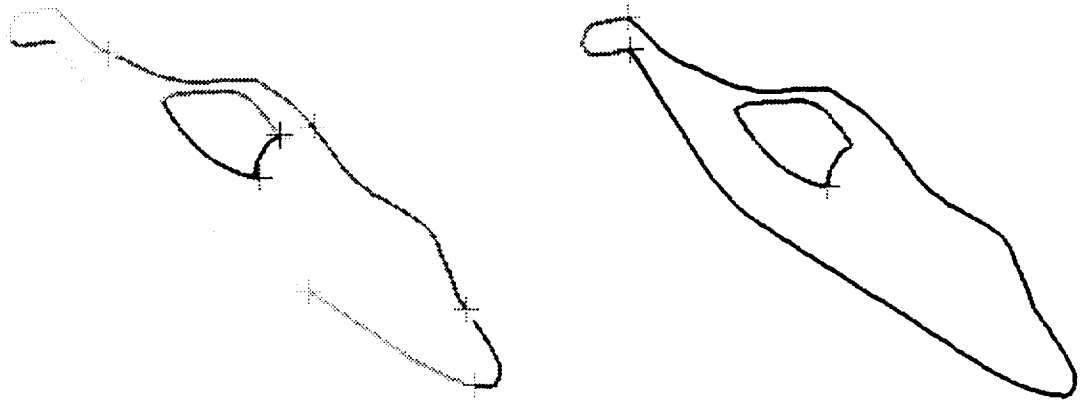


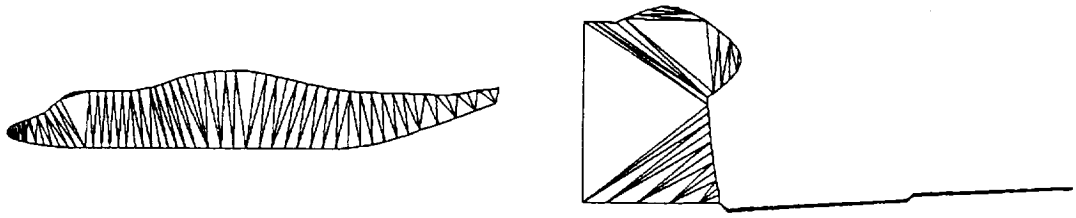
Figure 5. V-22 tiltrotor fuselage surface grid.



(a) Initial open boundary curves (11 segments) (b) Final open boundary curves (3 segments)

Figure 6. Initial and final open boundary curves.

4. Identify points inside the hole boundary in the Cartesian box – The points inside the hole boundary are identified by excluding all the points outside the hole boundary. For simplicity, a 2-D representation of this algorithm is illustrated in figure 8, though this algorithm is a 3-D algorithm. By closing in from the boundary where the status (inside or outside the hole boundary) of each point is known, one can safely say that a point is outside the hole boundary if any adjacent point is outside the hole boundary. Note that for this algorithm to work, the hole boundary needs to be a closed boundary. The advantage of this algorithm is multiple:



(a) Triangulation for V-22 tiltrotor symmetry plane

(b) Triangulation for orbiter backend

Figure 7. Triangulations for surface bounded by open boundary curves.

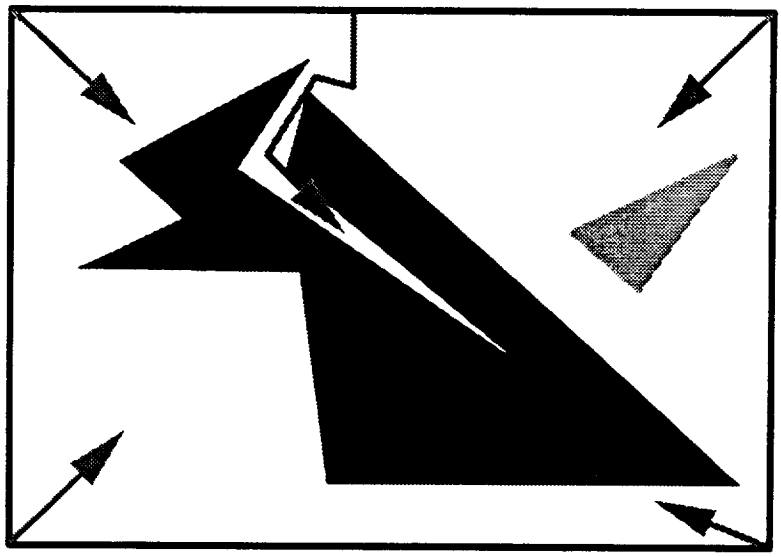


Figure 8. Algorithm to find points inside hole boundaries.

- Simple and robust – Unlike the common 2-D polygon filling algorithm[10], there is no need to find and sort intersections of the scan line with all edges of the polygon. Better, it eliminates the need to check for vertex intersections if the number of intersections on the sorted list is odd. Figure (9) shows the Cartesian approximation for the SOFIA telescope grid [11]. The current algorithm easily captures the shape of the original geometry.
- Easier to vectorize and longer vector length -- Since this algorithm does not rely on intersections of scan line to fill the 3-D volume, the vector length is usually longer than the number of points between each pair of intersections.

5. Hole cutting by truncation

- Identify cell in the hole-map that bounds the given X,Y,Z point by truncation. The indices of the

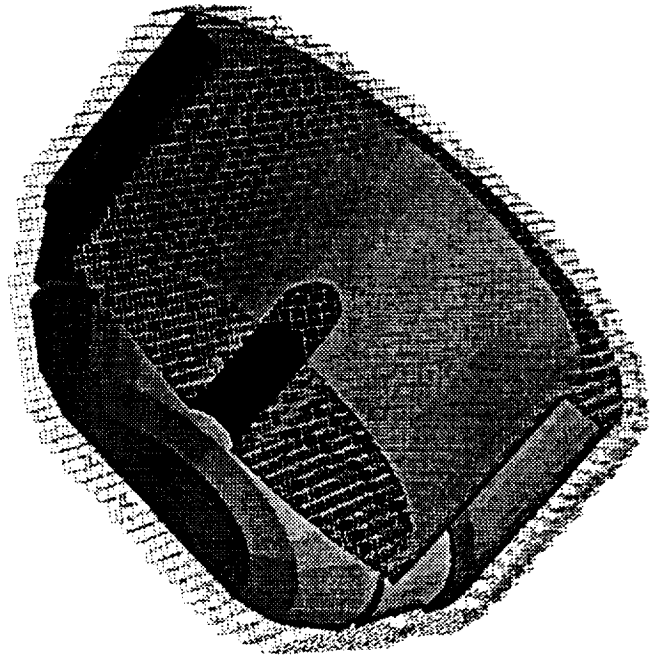


Figure 9. Cartesian approximation for the SOFIA telescope grid.

cell, J, K and L, can be found from the following equations.

$$J = int(\frac{X - X_0}{\Delta X})$$

$$K = int(\frac{Y - Y_0}{\Delta Y})$$

$$L = int(\frac{Z - Z_0}{\Delta Z})$$

where, (X_0,Y_0,Z_0) is the origin of the Cartesian box and ΔX, ΔY and ΔZ are cell size in X, Y and Z directions.

- Carry out inside/outside tests by checking the status (defined by value of IBLANK) of the hole-map element that bounds the X,Y,Z point.

3. RESULTS

Hole cutting through hole-map algorithm is tested for several testcases on a SGI 4D-210 workstation running on 25 MHz MIPS R3000 cpu. Figure 10(a) and 10(b) show hole boundaries cut in the shuttle external tank and orbiter grids, and a wing and store grid combination using hole-map technology. Table 1 shows the CPU time usage for these two cases.

The same wing/store case was also carried out using DCF3D with analytic shape cutters. The CPU time usage is 85.0 and 10.3 seconds. Thus, the performance of hole cutting using hole-map is roughly the same

as that of analytic shape cutters.

	# of Points	IGBPs	CPU Time
ET/ORB	588,240	7,628	155.4
Wing/Store	256,863	3,114	79.4

Table 1. CPU time usage for ET/ORB and Wing/Store testcases.

(a) ET/ORB hole boundary. (b) Wing/Store hole boundary.

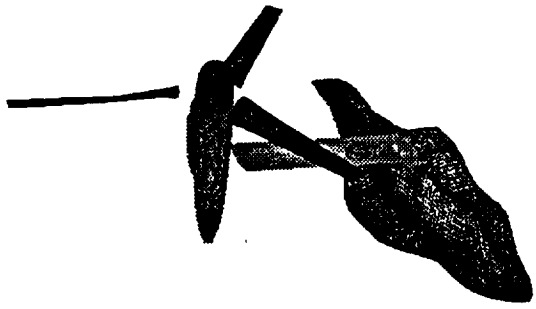
Figure 10. Hole boundaries.

Another testcase is a 25-grid 1.3 million point discretization of the V-22 tilt-rotor aircraft. Figure 11 illustrates the true hole surface and the approximate hole surface for this configuration. Figure 12 shows different shots of the hole boundaries computed through hole-map technology. As can be seen from the figure, the hole boundaries properly enclose each component grid. The hole-map is also capable of creating a hole at the fuselage/wing junction for the wing collar grid. The CPU time for cutting holes alone is 95.3 seconds with a total of 104 holes.

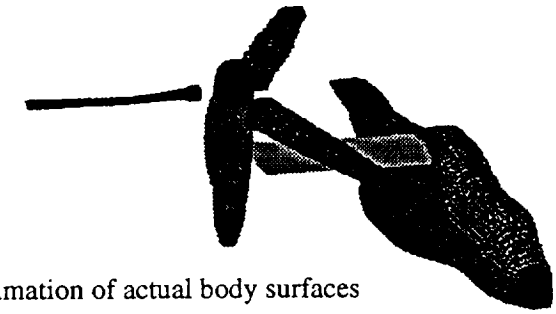
Figure 13 shows the hole boundaries due to an array of cylinders in a Cartesian mesh. All eleven holes were cut in a single pass instead of one hole at a time. This flexibility allows one to have mulitple holes in a single hole-map and also saves CPU time.

4. CONCLUSION

A Cartesian hole-map approach was implemented to take advantage of search by truncation to generate composite grid for a system of overset grids. The approach was tested on geometries of various degree of complexities – space shuttle (external tank and orbiter), V-22 tiltrotor, SOFIA telescope, generic wing/ store and array of cylinders. The performance of the algorithm proved to be approximately the same as the approach utilizing analytic shape cutters but without the tedious procedures in placing the cutter around the intended geometries. Due to the nature of the hole-map, one can reuse the hole-map for moving grids, assuming grids are not deforming, by translating or rotating the hole-map accordingly and thus eliminate the need to regenerate the hole-map. The Cartesian hole-map approach is only the first step toward automation of domain connectivity for overset grids. More work is required to obtain optimum domain connectivity – minimum overlapped region among overset grids, dynamic hole cutting and optimum interpolation accuracy.



Actual body surfaces



Cartesian approximation of actual body surfaces

Figure 11. V-22 actual hole surface and approximate hole surface.

5. REFERENCES

[1] Meakin, R. and Suhs, N., "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion," AIAA Paper 89-1996, June 1989.

[2] Meakin, R., "Transient Flow Field Responses about the Space Shuttle Vehicle During Ascent and SRB Separation," R. Aero. Soc. Store Carriage, Integration, and Release Conf., pp. 29.1-29.16, April 1990.

[3] Meakin, R., "Computations of the Unsteady Flow About a Generic Wing / Pylon / Finned-Store Configuration," AIAA Paper 92-4568, August 1992.

[4] Dougherty, C. and Kuan, J., "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme," AIAA Paper 89-0637, January 1989.

[5] Kiris, C. and Chang, I., "Numerical Simulation of the Incompressible Internal Flow Through a Tilting Disk Valve," AIAA Paper 90-0682, January 1990.

[6] Meakin, R. and Street R., "Simulation of Environmental Flow Problems in Geometrically Complex Domains. Part 2: A Domain-Splitting Method," Comp. Meths. Appl. Mech. Engrg., 68, 311-331, 1988.

[7] Dietz, W., Kacocks, J., Fox, J., "Application of Domain Decomposition to the Analysis of Complex Aerodynamic Configurations," SIAM Conf. Domain Decomposition Meths., Houston, TX, March

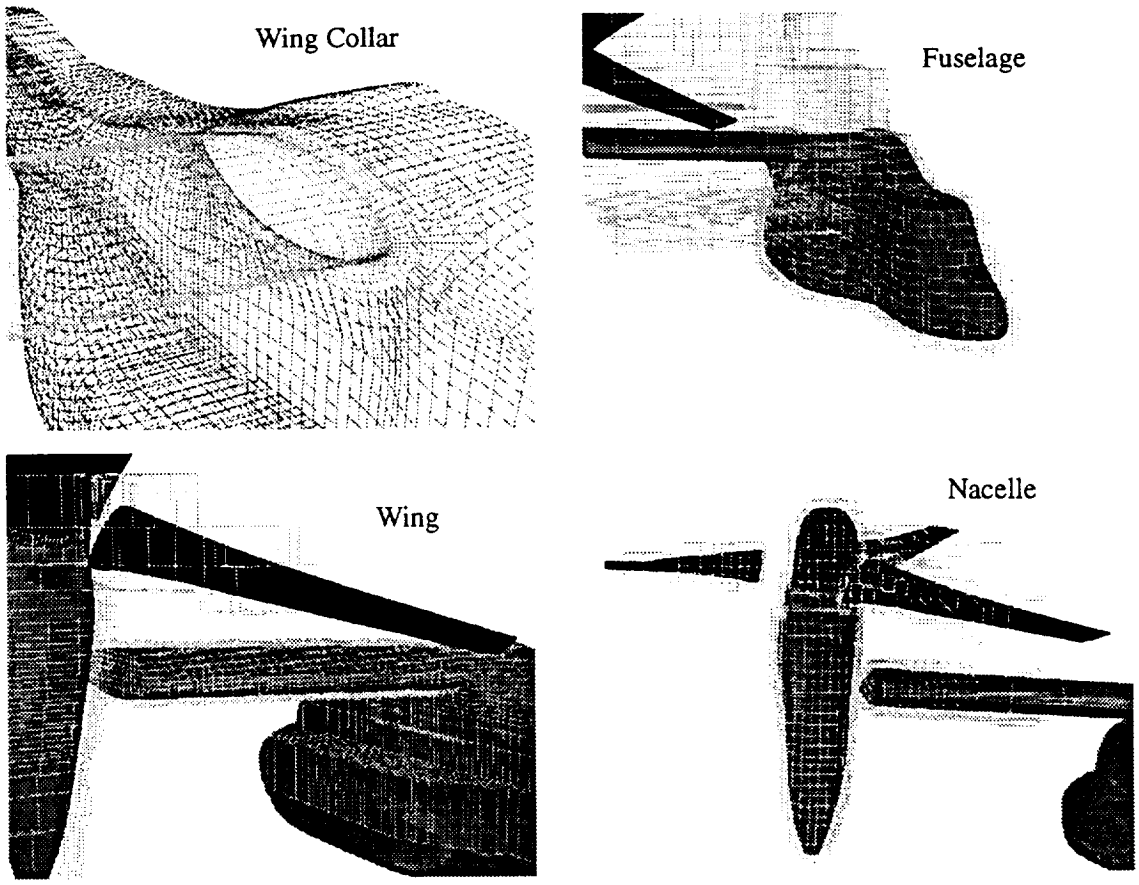


Figure 12. V-22 hole boundaries.

1989.

[8] Brown, D., Chesshire, G., Henshaw, W., and Kreiss, O., "On Composite Overlapping Grids", 7th Internat. Conf. on Finite Element Methods in Flow Problems, Huntsville, AL, April 1989.

[9] Meakin, R., "A New Method For Establishing Inter-Grid Communication Among Systems of Overset Grids," AIAA Paper 91-1586, June, 1991.

[10] Foley, J.D. and Van Dam, A., "Fundamentals of Interactive Computer Graphics," Addison-Wesley, Reading Mass., 1983.

[11] Atwood, C.A. and Van Dalsem, W.R., "Flowfield Simulation about the SOFIA Airborne Observatory," AIAA Paper 92-0656, 1992.

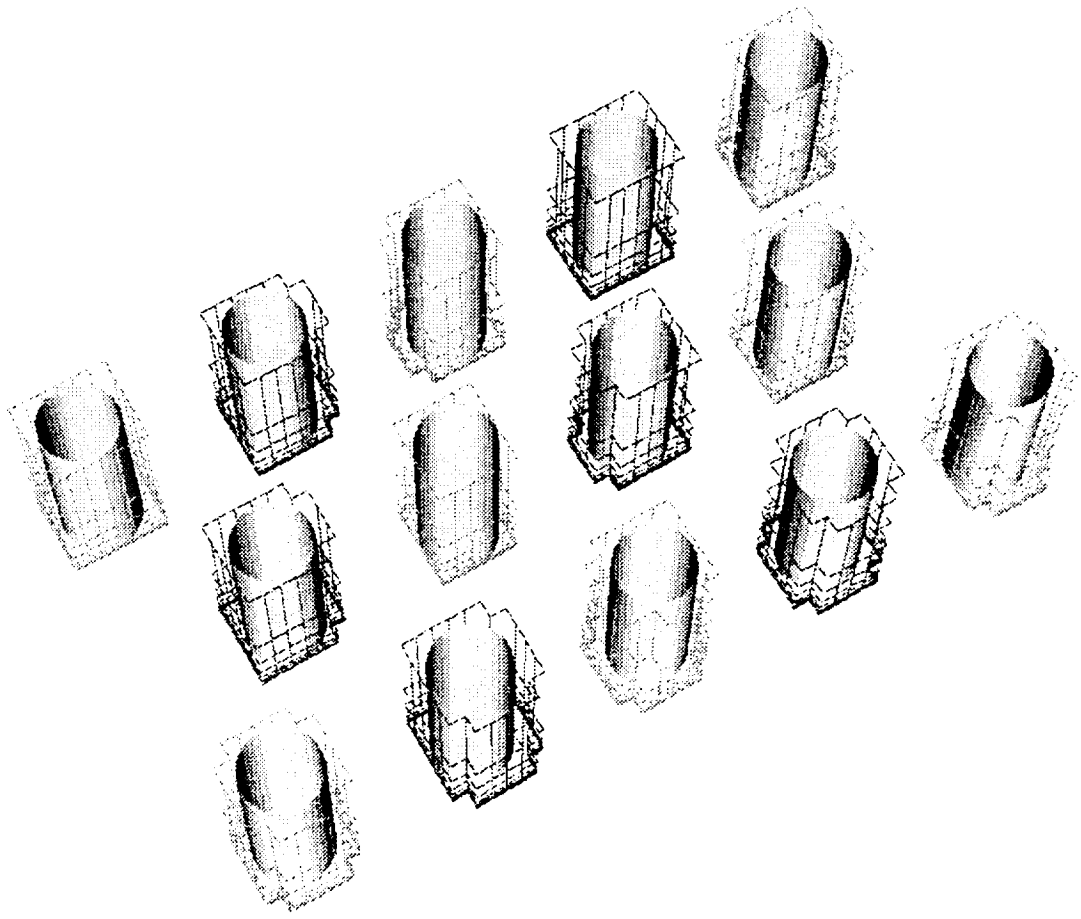


Figure 13. Hole boundaries for an array of cylinders.