ADVANCES TOWARDS AUTOMATIC SURFACE DOMAIN DECOMPOSITION AND GRID GENERATION FOR OVERSET GRIDS

William M. Chan * and Robert L. Meakin †
Army/NASA Rotorcraft Division
NASA Ames Research Center
Moffett Field, California 94035-1000

Abstract

An algorithm for surface domain decomposition and grid generation for overset grids is described. A complex surface domain is covered by two types of grids: seam grids and block grids. Seam grids are grids used to wrap around surface crease lines and regions of high surface curvature. Block grids are grids that cover the remaining surface regions not occupied by the seam grids. The seam and block grids form a set of overlapping surface grids that cover the entire surface geometry. Given a set of seam grids, an automated method for generating the block grids is presented in this paper. Examples of seam and block grids are given for several configurations including the V-22 tiltrotor fuselage and the X-CRV Crew Return Vehicle.

1. Introduction

The generation of high quality grids is typically the most time consuming step in the numerical computation of aerodynamic flow around complex configurations using structured grids. Amongst the structured grid methods, the overset grid approach[1] has been successfully used in a variety of applications.[2–8] However, existing grid generation software requires significant effort from the user to generate high fidelity surface grids for such complex cases. This is primarily due to two reasons. The first reason is that there is no robust algorithm or software available to decompose the surface domain into appropriate sub-domains for surface grid generation. Hence this step depends heavily on the experience of the user and requires intensive manual effort. The second reason is that there is virtually no software available that is targeted for overset surface grid generation where a marching scheme[9] is frequently more convenient than an algebraic interpolation or elliptic approach. Since the marching scheme described in Ref. [9] is not available in any grid generation system with CAD capabilities, overset grid users have been forced to use software designed for patched grids such as GRIDGEN,[10] ICEMCFD,[11] and NGP.[12]

In this paper, the focus will be on methods to improve the two problem areas described above: surface domain decomposition and surface grid generation. There have been several efforts to automate domain decomposition for the patched grid approach.[13,14] These require the decomposition of the volume space around the object as well as the surface of the object. This strategy is not ideally suited for an overlapped grid approach. Domain decomposition requirements for overset grids can be reduced to a surface geometry decomposition problem where the surface grids are allowed to overlap. Once the surface domain is decomposed, and the overlapping surface grids generated,[9] body conforming volume grids can be produced hyperbolically from the surface grids with little effort.[15,16] These volume grids will naturally overlap if the marching distance is not too large. The rest of the 3-D space can then be filled by layers of Cartesian box grids.[17,18]

Given a description of the surface geometry (reference surface), the algorithm for surface domain decomposition and surface grid generation presented in this paper consists of two steps: seam surface grid generation and block surface grid generation. Seam grid generation is primarily a manual process currently and is described briefly in section 2. Methods for performing block grid generation automatically are described in detail in section 3. The V-22 tiltrotor fuselage will be used as an illustrative example in these sections. A panel network definition of the V-22 fuselage is shown in Figure 1. Further examples of seam and block surface grids are given in section 4.

* Senior Research Scientist, MCAT, Inc., Member AIAA.
† Research Scientist, US Army Aeroflightdynamics Directorate, Member AIAA

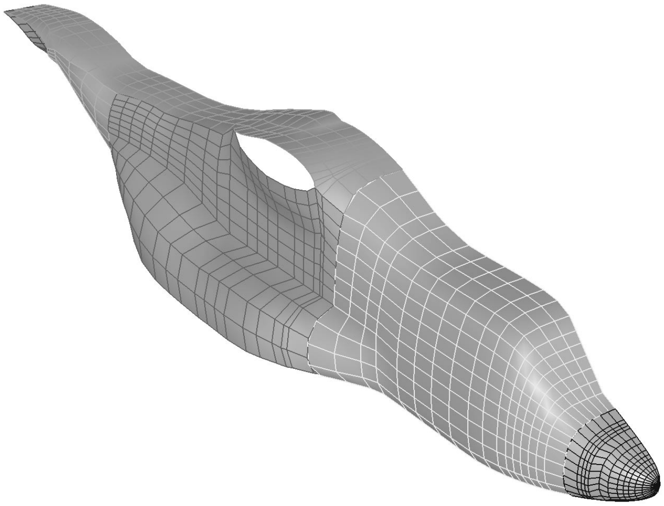

Figure 1. Surface definition of V-22 tiltrotor fuselage in the form of multiple panel networks.

2. Seam Surface Grid Generation

Seam surface grids are grids bounded by one or more special control curves (seam curves) of the surface geometry. The generation of seam surface grids can be divided into four steps. Since seam grid generation is not the focus of this paper, each step is described only briefly below.

(1) Identification of seam curves.

The seam curves are special control curves of the surface geometry. These control curves can be surface crease lines, curves along high curvature contours of the surface, intersection curves between different components, and other special curves. Currently, this step is performed manually but there is good potential for automation in the future.

(2) Identification of seam surface grid domains.

After the seam curves are identified, the surface domains that are to be bounded by one or more of the seam curves have to be determined. This may involve splitting or concatenating parts of different seam curves. Currently this step is performed by the user; it depends heavily on the user's experience, and is perhaps the most difficult step to automate. However, the availability of a graphical user interface (GUI) can significantly reduce the human effort required.

(3) Distribution of grid points on the seam curves.

The distribution of grid points on the seam curves can be specified directly by the user or determined automatically based on requirements of global resolution and local geometric constraints. The user may impose a global maximum grid spacing, while local geometric features may demand a smaller grid spacing.

(4) Generation of seam surface grids.

Finally, the seam surface grids are generated from one or more bounding seam curves. For a given seam surface grid, hyperbolic or algebraic marching schemes can be used if there is only one bounding seam curve. If multiple bounding seam curves are needed, algebraic interpolation schemes or elliptic methods can be used.

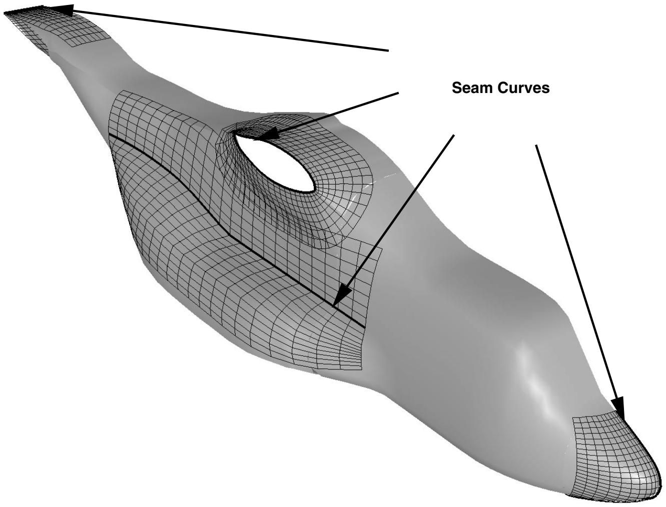

Figure 2. Seam curves and seam surface grids for the V-22 tiltrotor fuselage. The seam curves are marked by thick lines.

The seam curves and seam surface grids for the V-22 fuselage example are shown in Figure 2. A seam grid is introduced at the nose to avoid the need for a singular axis topology for a surface grid at this location. Other seam grids are introduced at the crease line on the fuselage, the wing/fuselage intersection line and the fuselage trailing edge. All grids are generated hyperbolically[9] by marching from the seam curves.

3. Block Surface Grid Generation

The algorithm used to automatically construct block grids is described in this section. It is assumed that the surface geometry definition (reference surface) and a set of seam surface grids are supplied.

The reference surface consists of one or more reference patches where each patch can be a Non-Uniform Rational B-Spline surface (NURBS surface), a panel network (rectangular array of surface points), or some other geometric entity. The principles presented below are independent of the form of these patches.

A software program called SBLOCK (stands for Surface BLOCKing) has been developed that uses multiple panel networks as the reference surface definition. The panel networks are assumed to be abutting but small gaps are tolerated between different networks. Point to point match between adjacent panel networks is not required. The generation of block surface grids can be divided into six steps. Each step is described in more detail in the following subsections. Treatments for other types of reference surface definitions such as NURBS surfaces can follow similar steps. In these cases, the reference surface is first converted to a multiple panel network representation. Then the steps below are performed followed by one additional final step where the block surface grids produced are projected back onto the original reference surface definition.

3.1. Creation of Marker Cells

The first step in block surface grid generation is to create a set of marker cells on the reference surface. Each marker cell is just a small region of the surface domain created by fracturing the surface domain in some convenient way. Groups of these cells can be flagged to mark regions where block surface grids should be generated as described in section 3.2.

Whether the reference surface is prescribed by patches of NURBS surfaces, panel networks or some other entities, it is assumed that a rectangular array of points can be created to represent each patch. For patches that are not four-sided, these can be subdivided into four-sided sub-patches where a rectangular array of points can be placed on each sub-patch. These points form the corners of a rectangular array of quadrilateral marker cells on each patch. The marker cells can be made to be uniformly spaced in physical space or follow some other constraints. In the SBLOCK code, points on each panel network of the reference surface are redistributed to create a set of uniform marker cells such that the size of each marker cell is no larger than a user-prescribed arc length ΔS_g.

3.2. Identification of Block Cells

The next step is to identify a set of block cells from which block surface grids are to be generated. Any marker cell that is not completely enclosed by a seam surface grid is flagged as a block cell. The identification of block cells is accomplished by projecting the vertices of all the marker cells onto the set of seam surface grids. Any vertex should lie either inside (including the boundary) or outside of a seam surface grid. A marker cell with one or more vertices lying outside of all the seam surface grids is identified as a block cell. This procedure guarantees overlap between the block cells and the seam surface grids. The block cells identified for the V-22 fuselage example are shown in Figure 3.

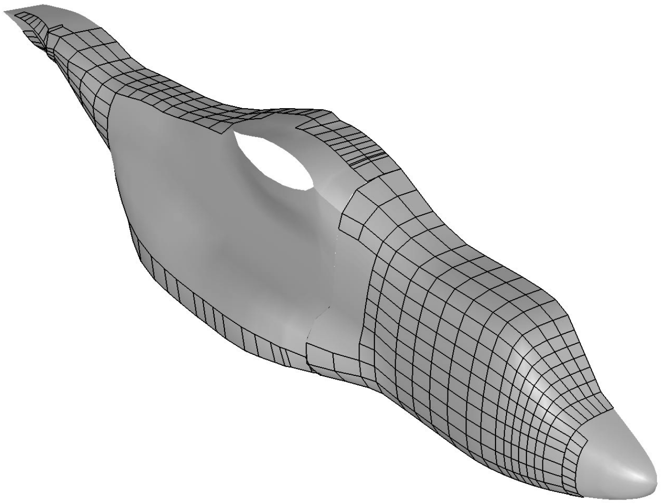

Figure 3. Block cells identified for the V-22 tiltrotor fuselage.

The critical element in this step (and in a number of steps discussed in subsequent sub-sections) is a robust and accurate projection routine. Given a vertex of a marker cell (target point), the set of seam surface grids that may contain the target point is first identified via the bounding boxes of the seam surface grids. Next, the seam surface grid that contains the closest point to the target point is selected. A Newton stencil walk[9] is then used to try to find bilinear interpolation coefficients for the target point within the seam surface grid. Normal displacements of the target point from the seam surface grid are disregarded provided the target point is within a certain tolerance from the surface. If the converged coefficients indicate that the target point is outside the seam surface grid, the next closest seam surface grid whose bounding box contains the target point is tried. This process terminates when the target point is accepted to be inside a seam surface grid, or when the target point fails to converge inside any of the seam surface grids. In this latter case, the point is flagged to be outside the seam surface grids.

3.3. Creation of Block Patches

The next step is to gather groups of block cells to form block patches. Let J and K be the index directions for the marker cells in a reference patch. Block cells from a reference patch are related to each other in the same index space. On each reference patch, a coalescence scheme is used to group together neighboring block cells in index space to form one or more block patches. Since block cells from different reference patches are from disparate index spaces, a different method is used to coalesce these cells (see section 3.4).

For each reference patch, block cells are first coalesced in the J direction to form J-pencils for each K = constant row of marker cells (see Figure 4). For a particular K = constant row of marker cells, two block cells are coalesced if they are separated by no more than J_c non-block cells. For example, if J_c is zero, then two block cells must be adjacent to each other for coalescence to occur. A larger value of J_c would result in fewer J-pencils at the end. Figure 4 shows three different arrangements of block cells that would result in the same J-pencil after the J sweep with $J_c = 2$. Typically, J_c is defaulted to a value of 5 in SBLOCK.

After the coalescence sweep in the J direction, the result is a collection of J-pencils. Each K = constant row may contain zero, one or more J-pencils. Next, adjacent J-pencils in the K direction are coalesced to form a block patch if their end points differ by no more than K_c cells. For $K_c = 0$, the end points of adjacent J-pencils have to match exactly for coalescence to occur. A larger value of K_c would result in fewer block patches at the end. Figure 4 shows three different arrangements of J-pencils that would result in the same block patch after the K sweep with $K_c = 2$. The value of K_c in SBLOCK is defaulted to 10 but may be modified by the user. The boundaries of the block patches of coalesced block cells for the V-22 fuselage example are shown in Figure 5.

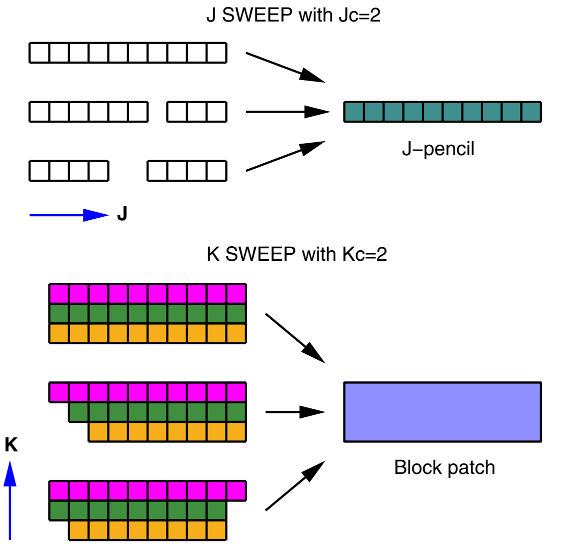

Figure 4. Coalescence sweep in J to form J-pencils and in K to form block patches. The J_c and K_c parameters can be used to reduce the number of J-pencils and block patches created, respectively.

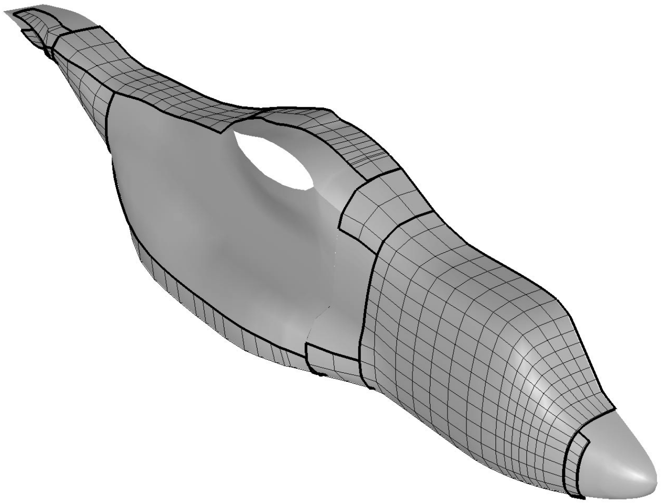

Figure 5. Block patches for the V-22 fuselage. Boundaries of block patches are marked by thick black lines.

3.4. Stitching of Block Patches

The scheme described in the last section serves to coalesce block cells that are derived from the same reference patch by performing connections in index space. In order to combine block cells from different reference patches, connections must be considered in physical space. A group of block patches originating from the same or different reference patches are 'stitched' together to form a stitched block patch by the algorithms described below. First, the candidate block patches for stitching are identified. Then, these block patches are joined by Type-1 stitching followed by Type-2 stitching. Type-1 stitching is used to join pairs of adjacent block patches that are almost the same size along the stitch direction. Type-2 stitching is used to combine a small block patch with an adjacent large block patch. These steps are described in more detail in the subsections below.

3.4.1. *Type-1 Stitching*

Type-1 stitching is performed between any pair of adjacent block patches with almost matching boundaries. For each point on each non-periodic boundary of each block patch, the neighboring block patch to the point is identified and stored. Identification of the neighboring patch is accomplished by the same search procedure as the projection scheme described in section 3.2. Point to point match between adjacent block patches is not required. Moreover, a user-specified tolerance parameter is used to allow for small gaps between adjacent block patches.

For the ith block patch P_i, let B_i be a boundary on P_i, M_i be the number of points on B_i, and $M_{i,j}$ be the number of points on B_i that has block patch P_j as the neighbor. Type-1 stitching will be performed between boundary B_i of block patch P_i and boundary B_j of block patch P_j if

(1) $M_{i,j}/M_i \geq f$, and

(2) $M_{j,i}/M_j \geq f$,

where f is a threshold parameter set to 0.95. By setting f to be less than 1, block patches that almost but not exactly match may be stitched.

The stitching scheme is designed such that the new patch covers at least the same surface regions as occupied by the original block patches. Moreover, the grid resolution on the new patch is at least as fine as either of the original block patches. The Type-1 stitching steps are outlined briefly below.

Suppose a Type-1 stitch is to be formed between boundary B_i of block patch P_i and boundary B_j of block patch P_j. Let the number of points on B_i and B_j be M_i and M_j, respectively, with $M_i > M_j$. Points on patch P_j in the direction along the stitch are redistributed uniformly to contain M_i points and the two

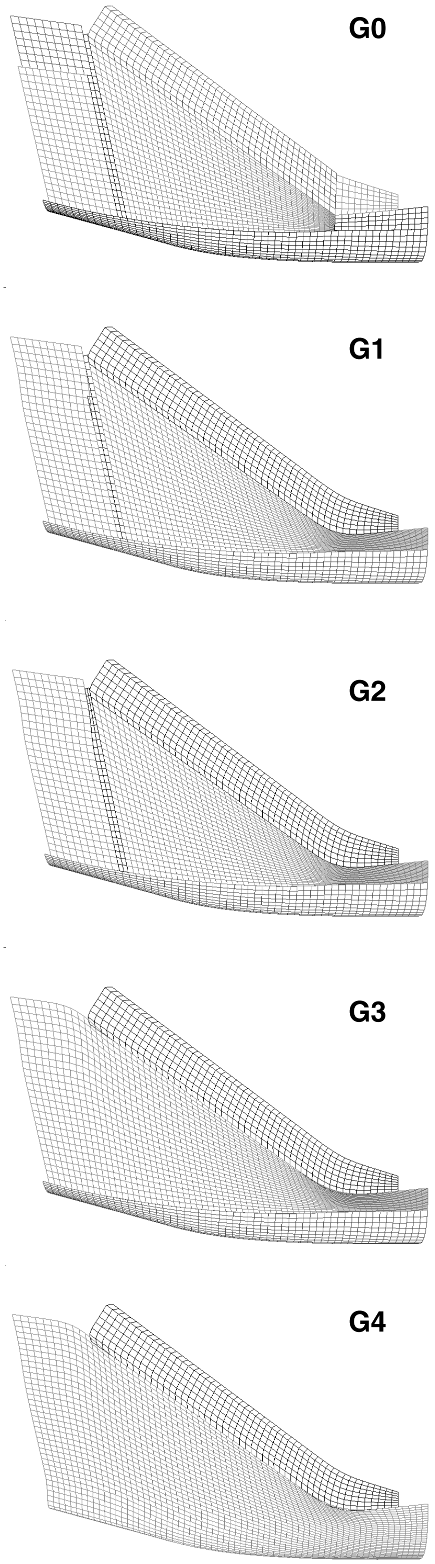

Figure 6. Multiple generations of block patches created by Type-1 stitching. Generation 0 (G0) contains 9 patches while generation 4 (G4) contains 2 patches.

patches are then concatenated. No redistribution is necessary if $M_i = M_j$. Since the grid points on the block patches are uniformly spaced to start with, points on the patch boundaries B_i and B_j should be almost point to point matched after the point redistribution. A few steps of elliptic smoothing is applied in the vicinity of the stitch to produce a smooth stitched block patch. Finally, the resulting points are projected back onto the reference surface.

After one round of Type-1 stitching, some of the block patches from the new generation may be combined further. Succesive rounds of Type-1 stitching are performed until no candidates for Type-1 stitching are found. Figure 6 shows 9 block patches in generation 0. After 4 rounds of Type-1 stitching, only 2 patches remain.

3.4.2. *Type-2 Stitching*

Type-2 stitching is performed after Type-1 stitching to combine a small block patch with an adjacent large block patch. Let $J_{\mathrm{max}i}$ and $K_{\mathrm{max}i}$ be the number of points in the J and K directions, respectively, for block patch i. A block patch with $J_{\mathrm{max}i}$ or $K_{\mathrm{max}i}$ less than or equal to N_{small} is flagged to be 'small', where N_{small} is a user-specified parameter typically set to be about 4.

If $J_{\mathrm{max}i} \leq N_{\mathrm{small}}$, then the $J = 1$ and $J = J_{\mathrm{max}i}$ boundaries will be eligible for stitching, while if $K_{\mathrm{max}i} \leq N_{\mathrm{small}}$, then the $K = 1$ and $K = K_{\mathrm{max}i}$ boundaries will be eligible for stitching. Type-2 stitching will be performed between boundary B_i of a small block patch P_i and boundary B_j of neighboring block patch P_j if $M_{i,j}/M_i \geq f$ where f is a threshold parameter set to 0.95 (see Figure 7a). The Type-2 stitching steps are outlined briefly below.

The outer boundary of the new patch is constructed from outer boundaries of the original patches as shown in Figures 7a and 7b. Boundary B_j from the large patch is modified to enclose boundary B_i' from the small patch. Interior points from the large patch are retained while interior points from the small patch are removed. New points are inserted in the new patch by linear interpolation in the interface region between the two old patches (Figure 7c). A few splining and smoothing steps are performed to produce a smooth stitched block patch (Figure 7d). Points on the new patch are projected back onto the reference surface.

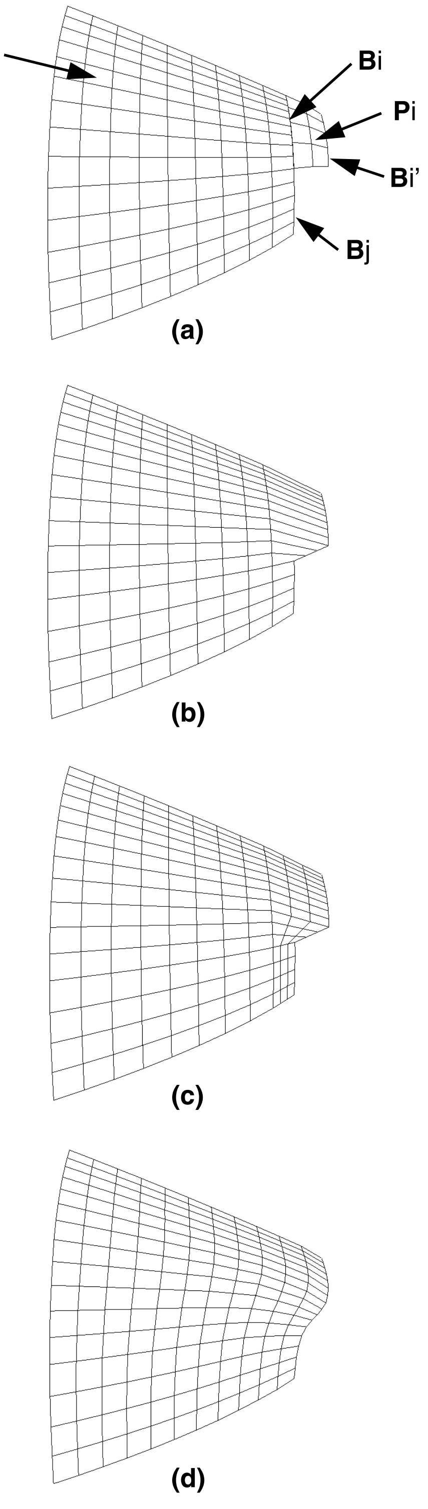

Figure 7. Steps for Type-2 stitching. (a) Original block patches. (b) Boundaries from the large block patch are modified to enclose the small block patch. (c) New points are inserted by interpolation near the modified boundaries. (d) Splining and smoothing are performed to produce a new stitched block patch.

3.5. Grid Spacing Adjustments

After the stitching step, the surface domain is now covered by a collection of seam surface grids and stitched block patches. One can think of the set of stitched block patches as a surface domain decomposition of the regions not covered by the seam surface grids. The next step is to distribute grid points appropriately on the stitched block patches to form block surface grids. Since the regions of the surface domain with surface discontinuities or high curvature have been covered by the seam grids, the surface regions covered by the block grids should typically be smooth. Hence a simple approach would be to use a uniform grid spacing for the block surface grids.

Recall that the marker cells have been created with a uniform spacing no bigger than a user-specified grid spacing ΔS_g. This spacing has been carried through the block patches and stitched block patches. One can certainly use the grid points on the stitched block patches as they are. Then the block surface grids are just the final stitched block patches. A useful enhancement has been made to this simple scheme by allowing a variable grid spacing on the block surface grids. Grid points on the block surface grids are redistributed using the hyperbolic tangent function such that (1) grid spacings at the grid boundaries match the grid spacings of any neighboring seam grids, and (2) the global grid spacing upper limit of ΔS_g is maintained throughout the grid. This ensures a comparable grid spacing between the block surface grid and the neighboring seam surface grids, thus allowing good inter-grid communication of flow solution information. Once again, the redistributed grid points are projected back onto the reference surface.

3.6. Extrapolation to Provide Overlap

The final step is to ensure that the collection of seam and block surface grids overlap each other. The block cells are constructed such that they overlap the seam surface grids. However, there is no overlap between neighboring block surface grids since they are derived from block cells that do not overlap (assuming the original reference surface patches do not overlap). Moreover, small gaps may sometimes appear between adjacent block surface grids from smoothing steps performed during the stitching procedures described in section 3.4. Excessive smoothing may deform the grid boundaries such that small gaps or overlaps are introduced.

In any case, an extra layer of points is added by linear extrapolation at each non-periodic boundary of each block surface grid to provide overlap with neighboring surface grids. However, an extra layer of points is not added if the block grid boundary lies on the domain boundary of the surface geometry, i.e. there is no seam surface grid or other block surface grid next to the block grid boundary. Finally the extrapolated points are projected on to the neighboring seam and block surface grids. The overlapping seam and block surface grids for the V-22 fuselage example are shown in Figure 8.

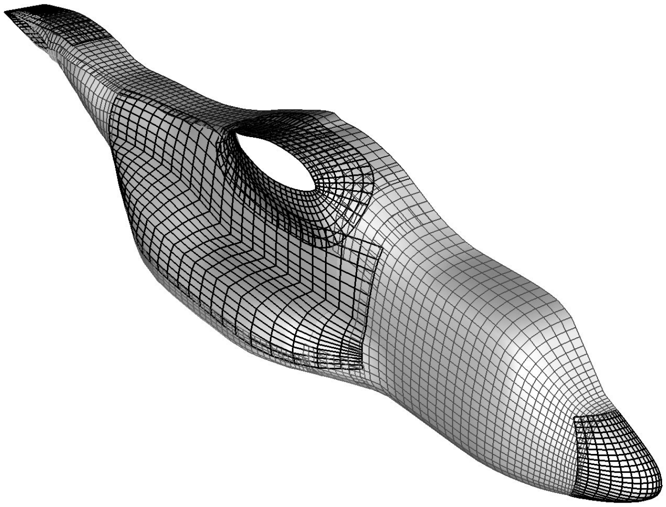

Figure 8. Overlapping seam and block surface grids for the V-22 tiltrotor fuselage (seam grids are drawn with thicker and darker lines).

4. Examples

Two other configurations of seam and block surface grids are presented in this section: a fin-store combination and the X-CRV Crew Return Vehicle. The fin-store case consists of 3 reference surface patches. The seam surface grids consist of a fin-store collar grid and a fin tip cap grid (Figure 9a). Six block surface grids were automatically created by SBLOCK (Figure 9b).

The geometry for the X-CRV was provided in IGES format with 38 NURBS surfaces (Figure 10a). Although small gaps exist at several locations between the NURBS patches, no attempt was made to fill the gaps before proceeding with the steps below. In order to use SBLOCK, the NURBS surfaces were converted to panel networks (PLOT3D[19] multiple surface grid format) using NGP[12] (the National Grid Project software package). This step required only a small amount of effort from the user. The next step involved the creation of the seam curves and the distribution of grid points on the seam curves. There is currently no software available that is convenient to use for accomplishing this step. Some work was required to perform this step using NGP.

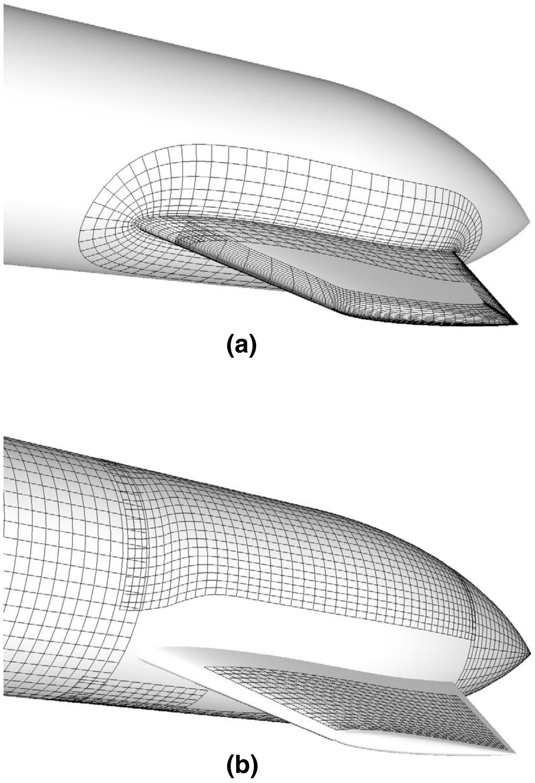

Figure 9. Fin-store configuration. (a) Seam surface grids. (b) Block surface grids.

Seam surface grids were then generated from the seam curves using SURGRD, an overset surface grid generator on multiple panel networks.[9] The 7 seam surface grids required for the X-CRV were produced with very little effort (Figure 10b). It took about 2 seconds of CPU time on a Silicon Graphics (SGI) R10000-175 MHz workstation to generate all 7 grids (total of about 8000 grid points). Block surface grids were then generated using SBLOCK. A global maximum grid spacing ΔS_g of 1/125 the length of the vehicle was used. The code produced 32 block patches after the coalescing step, and 19 overlapping block surface grids after the stitching steps (Figure 10c). The final 19 block surface grids contain about 23000 grid points and were created in about 28 CPU seconds on the above SGI workstation.

5. Concluding Remarks

This paper describes part of an on-going effort to fill the void in software available to overset grid users for surface domain decomposition and grid generation. An algorithm and a software program are described for performing partial surface domain decomposition and surface grid generation for overset grids. A complex surface domain is covered by overlapping seam surface grids and block surface grids. Given the surface definition and the seam surface grids, the block surface grids are automatically generated. As a result, the effort required from the user is significantly reduced.

The automated process involves fracturing the complex surface domain into small pieces (marker cells), and then using heuristic rules to coalesce/stitch the pieces back together into domains suitable for generation of overset surface grids. Grid points are then automatically placed in these domains using simple algebraic methods. Future work beyond the scope of the current paper will include exploring the development of algorithms and graphical user interfaces to reduce the effort required for seam surface grid generation.

Acknowledgements

The authors would like to thank Bell Helicopter Textron, Inc. and Boeing Helicopters for furnishing the V-22 geometry; and Ray Gomez of NASA Johnson Space Center for providing the X-CRV geometry. This work was performed as part of the DOD High Performance Computing Modernization Program funded through NASA contract number NAS2-14109, Task 23.

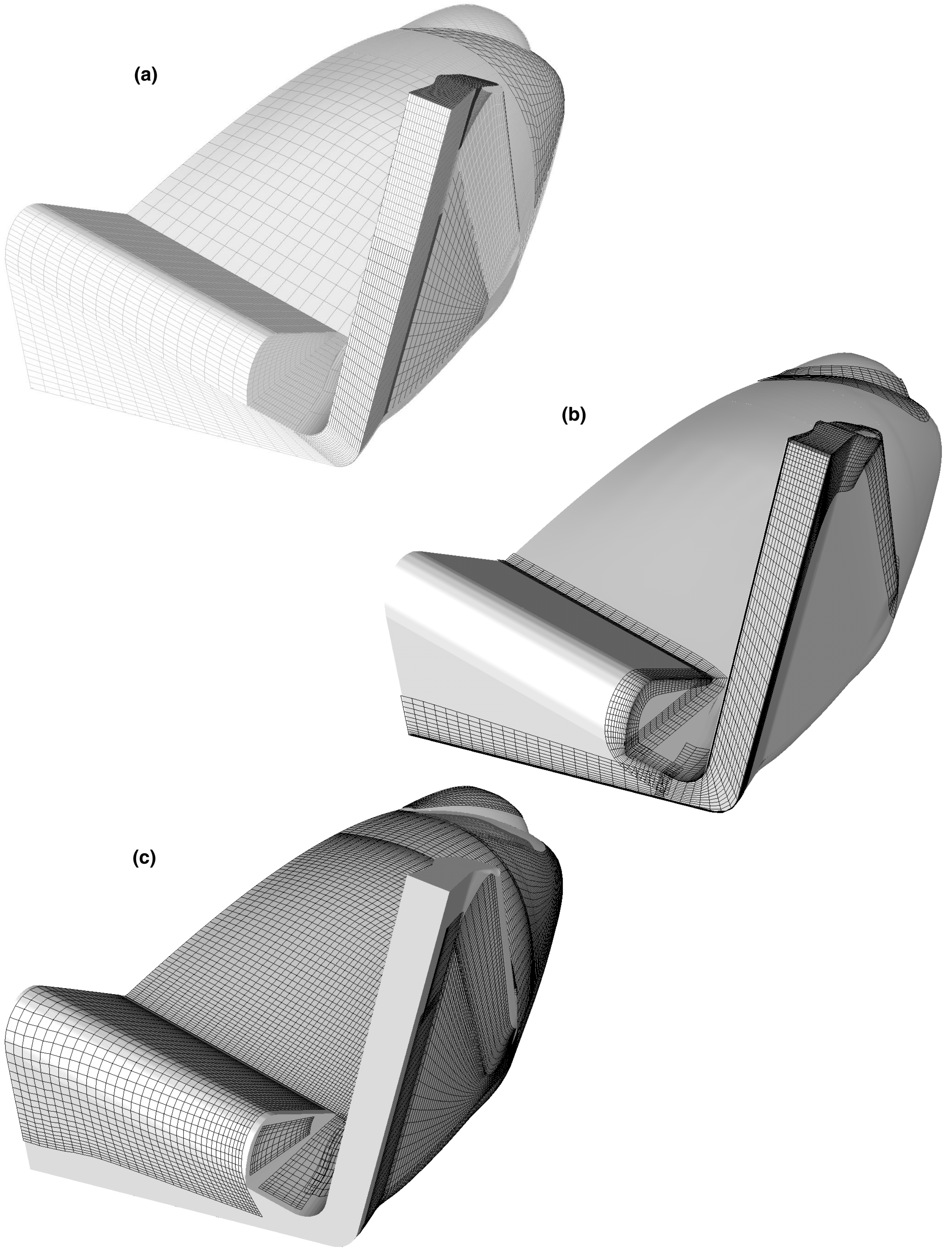

Figure 10. X-CRV Crew Return Vehicle. (a) Reference surface consisting of 38 NURBS patches. (b) Seam surface grids. (c) Block surface grids.

References

1. Steger, J. L., Dougherty, F. C. and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, K. N. Ghia and U. Ghia, eds., ASME FED-Vol. 5, June, 1983.

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

3. Meakin, R., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper 93-3350, Proceedings of the 11th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, July, 1993.

4. Gomez, R. J. and Ma, E. C., "Validation of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle," AIAA Paper 94-1859, Proceedings of the 12th AIAA Applied Aerodynamics Conference, Colorado Springs, Colorado, June, 1994.

5. 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 94-1860, Proceedings of the 12th AIAA Applied Aerodynamics Conference, Colorado Springs, Colorado, June, 1994.

6. Gea, L. M., Halsey, N. D., Intemann, G. A. and Buning, P. G., "Applications of the 3-D Navier-Stokes Code OVERFLOW for Analyzing Propulsion-Airframe Integration Related Issues on Subsonic Transports," ICAS Paper 94-3.7.4, 19th Congress of the International Council of the Aeronautical Sciences, September, 1994.

7. Wai, J., Herling, W. W. and Muilenburg, D. A., "Analysis of a Joined-Wing Configuration," AIAA Paper 94-0657, 32nd Aerospace Sciences Meeting & Exhibit, Reno, Nevada, January, 1994.

8. Duque, E. P. N, Berry J. D., Budge A. M. and Dimanlig A. C. B., "A Comparison of Computed and Experimental Flowfields of the RAH-66 Helicopter," Proceedings of the 1995 American Helicopter Society Aeromechanics Specialist Meeting, Fairfield County, Connecticut, 1995.

9. Chan, W. M. and Buning, P. G., "Surface Grid Generation Methods for Overset Grids," *Computers and Fluids*, **24**, No. 5, 509–522, 1995.

10. Chawner, J. R. and Steinbrenner, J. P., "Automatic Structured Grid Generation Using GRIDGEN (Some Restrictions Apply)," Proceedings of NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, NASA CP 3291, May, 1995.

11. Wulf, A. and Akdag, V., "Tuned Grid Generation with ICEM CFD," Proceedings of NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, NASA CP 3291, May, 1995.

12. Gaither, A., Gaither, K., Jean, B., Remotigue, M., Whitmire, J., Soni, B., Thompson, J., Dannenhoffer, J., and Weatherill, N., "The National Grid Project: A System Overview," Proceedings of NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, NASA CP 3291, May, 1995.

13. Dannenhoffer, J. F., "Automatic Blocking for Complex Three-Dimensional Configurations," Proceedings of NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, NASA CP 3291, May, 1995.

14. Eiseman, P. R., "Multiblock Grid Generation with Automatic Zoning," in Proceedings of NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, NASA CP 3291, May, 1995.

15. Chan, W. M. and Steger, J. L., "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme," *Appl. Math. and Comput.* **51**, 181–205, 1992.

16. Chan, W. M., Chiu, I. T. and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, October, 1993.

17. Meakin, R. L., "An Efficient Means of Adaptive Refinement Within Systems of Overset Grids," AIAA Paper 95-1722, Proceedings of the 12th AIAA Computational Fluid Dynamics Conference, San Diego, California, June, 1995.

18. Meakin, R. L., "On Adaptive Refinement and Overset Structured Grids," AIAA Paper 97-1858, Proceedings of the 13th AIAA Computational Fluid Dynamics Conference, Snowmass, Colorado, June, 1997.

19. Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM-101067, 1990.