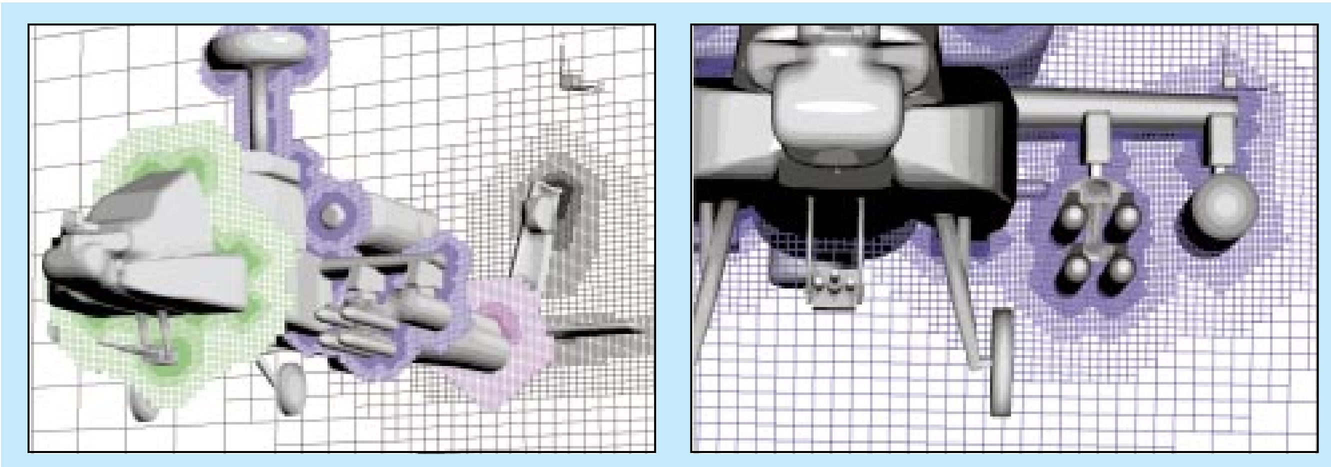
Robust and Efficient Generation of Cartesian Meshes for CFD

This algorithm is suitable for component-based flow geometries.

Ames Research Center, Moffett Field, California



A **Cartesian Mesh Around an Attack Helicopter**, containing 5.81×10^6 cells, was generated by the algorithm in a computation time of 320 seconds on a moderately powerful engineering computer workstation with a central processing unit running at a speed of 195 MHz.

An algorithm for the robust and efficient generation of Cartesian meshes for computational fluid dynamics (CFD) has been developed. The algorithm generates a mesh for computing the flow in a region bounded by solid components that have prescribed sizes, shapes, positions, and orientations, and that may be moving with respect to each other.

Some background information is prerequisite to a summary of the algorithm. Unlike some other computational grids, Cartesian meshes are not body-fitted. The cells of Cartesian mesh are hexahedral (more specifically, right parallelepipeds) and some cells can extend through surfaces of solid components in the computational domain. Therefore, part of any Cartesian-mesh-generation algorithm must include identification of cells that intersect solid surfaces and the flagging or removal of cells that are completely internal to the solid objects and thus not in the flow field. The remaining cells are then considered general volume mesh elements.

Fundamentally, in Cartesian approaches, one trades the case-specific problem of generating a body-fitted surface mesh for the more general problem of computing and characterizing intersections between hexahedral mesh cells and body surfaces. Thus, all difficulties associated with meshing a given geometry are restricted to a lower-order manifold that constitutes the wetted surface of the geometry.

Unlike the surface cells of a body-fitted mesh, the cells of a Cartesian mesh that intersect the surface are describable without describing the surface itself. In other words, the description of the surface is no longer needed to resolve both the flow and the local geometry. Therefore, efforts to describe the surface can be focused uniquely on the task of resolving the geometry, while computations that involve mesh cells are devoted to a description of the flow. Of course, accurate representations of boundary conditions in cells that intersect surfaces are essential to successful Cartesian schemes. This concludes the background information.

The present algorithm implements a two-phase strategy: In the first phase, intersections among all components are found and used to complete the description of the wetted surface, so that all surface-intersecting Cartesian cells found subsequently are guaranteed to be exposed to the flow field. The remaining mesh-generation problem can then be treated as if it were a single-component problem. In the second phase, the volume mesh is generated.

The component-intersection part of the algorithm is a robust geometry-oriented subalgorithm that, among other things, accommodates the surface triangulations commonly used to describe the surfaces of solid components. This subalgorithm utilizes adaptive precision arithmetic, and it includes a tie-braking routine that automatically and consistently resolves geometric degeneracies. The worst-case computational complexity of the intersection subalgorithm is of the order of $N \log N$, where N is the number of triangles describing the geometry.

The volume-mesh-generation part of the algorithm takes the intersected surface triangulation as input and generates the mesh through division of hexahedral cells of an initially uniform coarse grid. This approach preserves the ability to refine the mesh to different degrees in different directions, consistently with the local geometry, thereby making it possible to avoid generating excessive numbers of Cartesian cells in three dimensions. The mesh-generation subalgorithm has linear asymptotic computational complexity, with memory requirements that total approximately 14 words per cell. The figure depicts part of a mesh generated by the algorithm.

This work was done by M. J. Aftosmis and J. E. Melton of **Ames Research Center** *and M. J. Berger of the Courant Institute. For further information, access the Technical Support Package (TSP)* ***free on-line at www.nasatech.com*** *under the category.*

This invention is owned by NASA, and a patent application has been filed. Inquiries concerning nonexclusive or exclusive license for its commercial development should be addressed to the Patent Counsel, Ames Research Center [see page 20]. Refer to ARC-14275.