N95- 28738

1995122317

THE ROLE OF OVERSET GRIDS IN THE DEVELOPMENT OF THE GENERAL PURPOSE CFD CODE

Davy M. Belk
Air Force Wright Laboratory
Eglin AFB, Florida 32542

SUMMARY

A discussion of the strengths and weaknesses of overset composite grid and solution technology is given, along with a sampling of current work in the area. Major trends are identified, and the observation is made that generalized and hybridized overset methods provide a natural framework for combining disparate mesh types and physics models. Because of this, the author concludes that overset methods will be the foundation for the general purpose computational fluid dynamics programs of the future.

INTRODUCTION

Overset grid methods have been used to great advantage in the solution of flow fields over geometrically complex configurations, and for multiple bodies with relative motion. The usefulness of the overset approach has been demonstrated for problems ranging from three-dimensional flow around a ship (see appendix, Malmliden) to prediction of trajectories of aft ejected submunitions in supersonic flow (see appendix, Sahu and Nietubicz) and hypersonic HEDI shroud separation (see appendix, Narain). Other approaches for modeling complex geometry, such as the use of unstructured grids or block structured grids, do not offer the advantage of no regridding when bodies move relative to one another.

Although proven useful, there are definitely areas that need improvement. Among these are the usability of the codes that assemble systems of overset grids, the awkwardness of having multiple solutions in certain flow regions, and concerns over conservation errors at internal boundaries. Fortunately, there is a great deal of current interest in overset methods, and innovative solutions are being tested against these problems. Much of this work was discussed at the "Second Overset Composite Grid and Solution Technology Symposium," held at Fort Walton Beach, Florida in October 1994. The symposium was hosted by the Northwest Florida Section of the American Institute of Aeronautics and Astronautics. At this symposium, both development and applications related to overset grid methods were presented. (This paper will draw heavily on material presented at the symposium. The procedure for obtaining copies of a particular presentation are given in the appendix.) A review of accomplishments and current development in overset grid and solution technology has led the author to the conclusion that overset methods will be widely used in the future. In fact, the assertion is made that overset methods, in generalized and hybridized forms, will be the foundation for the general purpose computational fluid dynamics programs of the future.

In the remainder of this paper, strengths and weaknesses of overset composite grid methods will briefly be discussed, and representative work aimed at remedying the weaknesses will be referenced. Major trends in development will be identified, and the conclusion about the future of overset methods will be stated.

STATE OF THE ART

Strengths

The strong points of the overset grid methods are numerous. One of the most important items, and also one of the original motivations for this approach, is the ability to model bodies undergoing relative motion. As long as the bodies are rigid, the mesh system associated with the body can move with it and not require regeneration. This is a significant simplification that makes possible the solution of some problems that otherwise would be intractable. Another advantage is the flexibility afforded by the independent meshing of components. This allows selection of grid topology to suit the local geometry, and reduces the global ramifications of this selection. Flexibility also comes in the form of freedom to choose physics models and computational algorithms differently in the different grids. If the physical problem admits localization of viscous, chemical, or other effects, then a tailored grid and solution procedure can be placed in the locality required.

Overset methods preserve the efficiency of structured grid solvers (also of other solution methods, as will be discussed later). Solution algorithms typically require minimal modification to operate on an overset grid. The logic required to determine holes and interpolation coefficients and other items associated with assembling a system of overset grids is usually divorced from the solver. The solver only needs to accept an array indicating cells which are not to be included in the solution. Canned routines are usually used to perform intergrid communication, letting the solver be "ignorant" of most communication issues.

An alternative adaptive refinement is possible with overset grids. Adaptive refinement can be achieved in overset grid systems by inserting fine grids in regions in need of resolution.

The next round of advances in high performance computing capability is generally expected to come in the form of a shift from vector processing to massively parallel processing. Domain decomposition into overset grids is a natural mapping of existing algorithms to parallel machines. This has been done at Arnold Engineering and Development Center (see Benek, appendix), with quite high efficiency. Jesperson and Levit (see appendix) showed that spreading each grid across processors on a CM5 is also an option. Their overhead for intergrid communication was "noticeable but not insurmountable."

The mode of problem solution using overset grids is well suited to a production environment. The required component grids of a complex geometry can be developed semi-independently by different engineers, and assembled by yet another engineer. A high degree of reuse is possible with grids developed for use in an overset environment, resulting in libraries of frequently used geometries that can be easily put together. For example, if models of the F-16, the 370 gallon tank, the weapons pylon, and several different weapons are sitting on the shelf, then it is a relatively simple procedure to investigate the aerodynamics of the aircraft and various combinations of tank and weapons. If multiple copies of the same weapon are to be carried, then the engineer simply inputs how many and where. If another aircraft model is available, the same weapons can be put on this aircraft very quickly.

It is also relatively easy to add small appendages to previously developed models. Jolly, et al. (see appendix) showed an example of a quick turnaround analysis of an external battery pack addition to a weapon. The previously developed weapon grid was not modified, but served as the outer grid into which the battery pack grid was inserted, as shown in Figure 1.

Weaknesses

A frank discussion of overset grid and solution methods requires that consideration be given to the valid criticisms of these methods. These areas of concern fall into familiar categories of accuracy, efficiency, usability, and algorithmic complexity. A synopsis of the issues is given below, and a discussion of ongoing research targeting these issues will be given in the next section.

Conservation at internal boundaries is the most often voiced concern about the accuracy (or even the validity) of overset method application to problems involving shocks and discontinuities. Communication between domains via interpolation of boundary values is, in general, nonconservative. Meakin (ref. 1) performed a careful study of spatial and temporal accuracy of overset grid methods through numerical experimentation. He found that if a flow solution is represented smoothly in both donor and recipient grids, simple interpolation is sufficient to maintain formal solution accuracy. Still, this issue continues to be bothersome since practical situations arise where the flow solution is not represented smoothly on both grids. In these cases, according to Meakin, a conservative interface scheme is preferable, but accuracy will be compromised regardless.

The efficiency issue is one of wasted calculations on both regions with multiple overlapping grids and in blanked regions. In overlapping regions, each grid can have a calculated solution, and each solution will be different due to the different discretizations. Cells in blanked regions usually undergo the same set of calculations as field cells in order to maintain vectorization of code. This can add up to a considerable number of wasted calculations, and is particularly wasteful on cache-based (non-vector) machines.

Usability is not at the desired level as yet. Considerable user expertise and interaction is required to determine where to cut holes, from which grids to interpolate values, how to generate the grids to achieve resolution matching in interpolated regions, and how to prevent circular interpolations.

Finally, the computer code required to implement overset methods can become quite complex, especially when an attempt has been made to address some of the previously mentioned issues. What begins as a straightforward idea can quickly become unwieldy in its implementation.

DEVELOPMENT

The development work described below is aimed at alleviating the existing weaknesses of overset methods. Much of the work described was presented at the 2nd Overset Composite Grid and Solution Technology Symposium (see appendix).

Conservation and Solution Accuracy

Most of the techniques aimed at ensuring conservation have achieved this result by actually eliminating overlap between the grids in the system. Wang's method (see appendix) in a region of two grid overlap is to leave one grid whole and eliminate the other grid from the overlap region. The precise intersection of the outer boundary of the whole grid with the cut grid is mapped out, and unique fluxes are calculated and distributed appropriately along this boundary. Kao and Liou (ref. 2, also see appendix) use a method they have dubbed the Direct Replacement of Arbitrary Grid-overlapping by Nonstructured (DRAGON) Grid technique. Here, overlapping grids are pared back to eliminate overlap, and glued together with an unstructured mortar grid. Another idea, not so directly motivated by conservation issues as by the advantage of unstructured grids in a particular region, is to simply tie an embedded unstructured grid to a structured grid. This has been promoted by Wang (see appendix) and others. In this form, there is little conceptual distinction from hybrid schemes that essentially embed a structured grid near no slip boundaries and use unstructured outer grids.

Multiple Competing Solutions

Wasted Calculations.-- For complex problems, mesh arrangements can easily occur that have multiple layers of grids in some regions of space. Instead of allowing each grid to have its own different solution, and communicating between meshes only on the grid boundaries and hole fringes, Johnson and Belk (ref. 3) use these layers of grids as levels in a nonaligned multigrid scheme. Interpolation of residuals and dependent variables from finer meshes to coarser meshes throughout the region of overlap is used to calculate a defect correction driving the coarse grid solution. The coarse grid solution is then used to update the fine grid variables. This procedure was shown to provide a means for improving accuracy by embedding grids aligned with flow features, but without the need to cut holes. Also, the convergence rate of the resulting grid system was significantly enhanced in comparison to the standard overset communication scheme of cutting holes and interpolating at boundaries.

In a similar manner, Rogers and Pulliam (ref. 4) demonstrated the advantages of a defect correction approach for communication between overset grids. In this work, the simplification of not actually calculating a fine grid solution was used. Instead, the coarse grid solution is interpolated to the finer embedded grid and a defect correction is calculated to drive the coarse grid solution to higher accuracy.

One way to minimize wasteful calculations is to blank out as many overlapping cells as possible. (Of course, unless the code uses conditional execution to skip blanked cells, this has negligible influence on the number of floating point operations required for a solution step.) The grid assembly algorithm of Chesshire and Henshaw (ref. 5, also see appendix) has the property of minimizing overlap subject to certain criterion on relative cell sizes. Several others, such as Wey (ref. 6, also appendix) and Chiu and Meakin (see appendix) have techniques to minimize overlap. In these methods, the hole boundary expands until it can go no further and still maintain valid overset communication.

Still another approach is to eliminate overlap entirely, such as the schemes discussed earlier under the topic of conservation.

Force Integration. -- A practical problem that results from the existence of multiple overlapping solutions is force integration on bodies. For example, a fin added to a missile using overset methods will result in a region near the fin with both the fin grid and the body grid conforming to the missile body. There will also be a region of invalid body cells, or holes, adjacent to the fin, as shown in Figure 2. Accurate integration of predicted pressure and shear stress to obtain body forces and moments requires some scheme to prevent either doubly counting some regions, or leaving holes.

One option is to ignore the problem. If the mesh is fine, and the hole in the body grid near the fin is relatively small, then the body force contribution can be adequately represented by using the body grid alone. While not the most satisfying solution, the results of Lijewski and Suhs (ref. 7) show that trajectories of separating stores can be accurately predicted with this approximation.

Another approach is that of Dietz, who wrote the TESS code to handle this situation (see appendix). The TESS code includes all points from overlapping grids on a surface, and triangulates the intermixed points to produce a unified surface mesh. The points carry their properties with them, and the resulting triangular mesh is used to integrate forces and moments.

Chan's method (ref. 8, also see appendix), on the other hand, first eliminates overlap on the surface by removing points belonging to the coarsest overlapping grid. The pared back surface meshes are then reconnected with a triangular grid, and the forces are integrated on this mesh.

The author's approach (ref. 9, also see appendix) is to try to avoid this problem by representing static assemblages of components with block structured grids called superblocks. Within a superblock, no grid overlapping occurs, except implicitly at boundaries. The superblocks are allowed to be placed in arbitrary overset arrangements. Figure 3 shows a wing, pylon, and finned store grid system with a blocked-grid superblock around the store.

Usability

Significant improvements in usability of overset grids and solvers have come from attempts to automate the grid assembly process. Interactive tools have also been developed to aid in the grid assembly process. One of the most important developments from a user's standpoint, however, is the work of Chan et al. (ref. 8, 10, also see appendix) to develop grid generation tools for overset systems. Most of the existing overset grid assembly tools still require externally generated grids. If the grids are not compatible in terms of having comparable point density and a sufficient number of cells to allow an interpolated fringe after hole cutting, then the meshes must be modified. In addition, grids around intersecting surfaces must either be built to conform to both surfaces, or a collar grid must be used to avoid a region where points from all grids have been blanked. Chan's SURGRD code (ref. 8) is a surface grid generator tailored for overset grids, and the HYPGEN (ref. 10) code is a volume grid generation code particularly useful for overset grids.

Interactive tools are available to set up the input for Meakin's DCF3D Code. These tools simplify the specification of analytic hole cutting surfaces, and give immediate feedback on the relative location of the hole boundaries and bodies. Search hierarchy, boundary information, and other information is set interactively.

Easy assembly of overset grids has always been a goal. However, different approaches have been taken concerning the degree of user control desired or required in the process. The most mature code that has attempted to minimize user control requirements is Chesshire and Henshaw's CMPGRD program (see appendix). The code uses an algorithm that determines hole size and overset connections with little user intervention, attempting to minimize resolution mismatch in interpolated regions. CMPGRD is also set up to allow for multigrid within each component grid. Chiu and Meakin presented work towards automating domain connectivity for overset grids (see appendix). Their methods use the inverse map data structure to allow inexpensive determination of the curvilinear coordinates (i.e. interpolation coefficients) of a point. The specification of hole cutting surfaces is greatly simplified by using cartesian approximations of the body surfaces called hole maps. Dynamic hole expansion and shrinkage to allow minimization of overlap, or optimization of interpolation locations in the future, is also available. Wey modified an advancing-front technique for generating an unstructured mesh to efficiently and automatically assemble overset structured grids (ref. 6, also see appendix). His method considers the boundaries as a collection of vertices, edges, and facets, and is in principle quite flexible. Wey uses an implementation of the "enlarged orientation theorem" to efficiently solve the important problem of whether a point lies in the interior, exterior, or on the boundary of a prescribed surface or front. Other's have also attempted varying degrees of increased automation and decreased user input, including the author.

Maple and Belk have developed an integrated flow solver and overset mesh system assembler called the Beggar code (ref. 9, also see appendix). Blocked, patched, and overset grids are used by the Beggar code. A hierarchical arrangement is used to improve efficiency, with collections of blocked and patched grids grouped together in a superblock. Superblocks are allowed to overlap with other superblocks to form the overset grid system. The idealistic goal for this computer program was the ability to accept input solely consisting of:

1. Good grids.
2. Physical boundary conditions other than farfield, e.g. tangent flow, no slip wall, or mass flow at an inlet face, etc.
3. Flow solver and six degree of freedom integrator parameters, e.g. CFL number or time step size, number of subiterations, etc.
4. Specification of required output.

The code should then assemble the grids, determine the physical and numerical boundary conditions on all unspecified grid surfaces, and produce the solution. We have had fair success in meeting these goals, with the primary difficulty being in the definition of good grids.

The data structure used in the Beggar code to facilitate determination of holes and interpolation coefficients is a variant of the polygonal mapping (PM) tree (ref. 11, 12). Stencil jumping (Newton iteration) is used to determine precise interpolation coefficients, but this process requires good starting guesses, and can be very expensive to use to determine that a cartesian coordinate does not lie within a grid. The desired benefit from the Beggar data structure is to unambiguously provide a list of grids containing a point, and then to give a stencil jumping starting guesses for these grids guaranteed to converge. The PM tree satisfies these requirements in nearly all cases. Basically, an octree data structure provides gross subdivision of the solution domain into smaller regions that have the property of being entirely within a grid, entirely outside a grid, or on a grid boundary. Those regions that are entirely within a grid satisfy the requirement that a single start point is guaranteed to result in successful stencil jumping within the region. Those regions that are on the boundary contain a small binary space partitioning (BSP) tree that provides accurate in/out determination for all points in the region, as well as appropriate stencil jump starting points depending on the grid in which the point falls.

Complexity

As discussed in the previous section, many new methods are being developed to automate the process of overset mesh assembly. This results in significantly less workload on the end user of these techniques, but this is not without some cost. The codes themselves tend to become more complex, and less inviting for the engineer or even the numerical algorithm specialist to delve into. The advanced algorithms having to do with overset mesh assembly are often computer science algorithms, not numerical algorithms. Overture++ is a C++ class library designed by Reider and Quinlan (see appendix) to lessen the difficulty of writing partial differential equation solvers using overset grids. This class library will offer utility routines to perform tasks that are generic to all overlapping grid operations, including memory management. Pao (see appendix) gave results for a compressible low speed flow algorithm implemented using Overture++.

SUMMARY AND DISCUSSION

Reviewing the material that has been presented here, it is evident that the push for more accuracy, efficiency, and usability of overset methods is slowly, but fundamentally, changing the overset community's approach. Formerly pure overset methods are allowing combination with blocked, patched, and cartesian structured grids. Even more revolutionary for this community is the use of unstructured and prismatic grids in some regions. Communication between all these disparate grid types is usually done in one of two ways: 1. the traditional hole cutting, overlap, and interpolation (Schwartz alternating procedure), or 2. Elimination of overlap and use of unstructured grid mortar across the gap. The other fundamental change is in embracing more complex data structures and advanced algorithms. This includes use of unstructured grids, octrees, BSP trees, inverse maps, alternate digital trees, and C++ class libraries.

The original impetus for the use of overset grids was the desire to model complex geometry using body fitted curvilinear coordinates. The rationale for this was accuracy, efficiency, and code reuse. Accuracy -- because body fitted coordinates allowed accurate boundary condition imposition, provided for viscous layers, etc. Efficiency -- since the computational rectangle associated with each structured component grid allowed vectorization. Code reuse -- because each grid was just another computational rectangle to the solver. Another historical motivation was that relative rigid motion of bodies could be modeled with no regridding required.

The environment is of course different now than fifteen years ago. There are still strong arguments for the use of overset methods, at least for the generalized overset methods that have come into use recently. One argument that has stayed the same is the usefulness of overset methods in modeling moving bodies without regridding. Overset methods still offer advantages in accuracy and efficiency. Accuracy and efficiency can no longer be taken as strictly synonymous with structured grids and vectorization, however. Instead the argument is that overset methods allow use of the most appropriate mesh type and physics model in each domain of the problem. Adaptive refinement via additional meshes is possible. Also, parallelization by domain decomposition is natural and does not require a homogeneous processor environment. Code reuse is still a real advantage of overset methods. Here, too, the emphasis has changed. No longer does this imply that a single solver package should be reused on all the different grids. Instead, when putting disparate mesh types and physics models together, the solvers developed for structured, unstructured, cartesian meshes, and so on can be linked together with relatively minor modifications to operate on a generalized overset grid system.

CONCLUSION

Overset methods are changing to meet the demands of current problems and to take advantage of available technology. The overset techniques and computer codes can usually incorporate different mesh types and physics models without major restructuring of either the overset code or the stand alone version of the new solver or model. Because of this, the overset codes are ideally suited to growing into the "general purpose" CFD codes of the future.

APPENDIX

The following are the presentations given at the 2nd Overset Composite Grid and Solution Technology Symposium, held at Fort Walton Beach, FL on October 25-28, 1994. Copies of particular abstracts and presentation material are available for a nominal fee from the symposium General Chair, Dr. Lawrence Lijewski. He can be reached at phone (904)882-3124, ext 3376, e-mail lijewski@eglin.af.mil, or

WL/MNMW
101 W. Eglin Blvd. Suite 346A
Eglin AFB FL 32542

Presentations

Introductory Remarks - J. Benek

Research Topics in Computational Methods for Unsteady Multiple-Body Aerodynamics - R. L. Meakin, Overset Methods Inc.

Issues and Advances in Overlapping Grid Generation - G.S. Chesshire and W.D. Henshaw, Los Alamos National Laboratory

A New Approach to Domain Decomposition: The Beggar Code - D. Belk and R. Maple, Wright Laboratory

Overture++: A C++ Class Library for Overlapping Grid Solvers -M. Reider, Los Alamos National Laboratory

Two New Chimera Methods: Application to 3D Store Separation - J.P. Gillyboeuf, Aerospatiale-Missiles; P. Mansuy, Matra Defense; and S. Pavsic, ONERA; France

A Method for Computing the 3D Flow Around a Ship using Composite Overlapping Grids - J. F. Malmliden, Royal Institute of Technology, Stockholm, Sweden

Chimera Grid Application for Fighter Configurations - M. Mani, McDonnell-Douglas

Verification of a Transonic Euler Solution of an F-16 Aircraft with a Generic Finned Pressure-Instrumented Store Using Chimera Grid Scheme - W.C. Riner, B.A. Jolly, N.C. Prewitt, Sverdrup Technology Inc.; and J.M. Brock, Jr., Test Wing, Eglin AFB, Florida

On Automating Domain Connectivity for Overset Grids - I.T. Chiu and R.L. Meakin, Overset Methods Inc

Development of an Automatic Mesh Interface Generator for Overlapped Structured Grids - T. C. Wey, NASA Johnson Space Center

Tutorial: PEGSUS - N.E. Suhs

Computational Fluid Dynamics for Multiple Projectile Configurations - J. Sahu and C.J. Nietubicz, Army Research Laboratory

Numerical Simulation of a Paper Coating Flow - F. Olsson, Royal Institute of Technology, Stockholm, Sweden

Computational Methodology for Time-Accurate Multiple Body Motion -R.D. Thoms and J.K. Jordan, Calspan Corp

The Prediction of Unsteady HEDI Shroud Separation Event - J.P. Narain, Lockheed Missiles and Space Co.

Tutorial: BEGGAR - D. Belk

Conservation and Linear System Issues on Overset Composite Grids - J.S. Saltzman, Los Alamos National Laboratory

An Advance in Overset Grid Schemes: From Chimera to DRAGON Grids, - K.H. Kao and Meng-Sing Liou, NASA Lewis Research Center

Conservative Chimera for 3D Euler Equations on Structured/Structured, Structured/Unstructured Grids, - Z.J. Wang, CFD Research Corp

Tutorial: DCF3D, R.L. Meakin

Comparisons of Overlapping Grid Communications with Beggar and Pegsus - N.C. Prewitt, Sverdrup Technology, Inc

Navier-Stokes Analysis for Propulsion-Airframe Integration by Using OVERFLOW/Chimera Overset Grid Approach - L.M. Gea, McDonnell- Douglas Corp.

Some Experiences with the NPARC Overset Grid Capability - H.J. Thornburg, B.K. Soni, M.H. Shih, B.K. Kishore, Mississippi State University

General Approach to Calculating Forces and Moments on Overset Grid Configurations - W.E. Dietz, Calspan Corp

Recent Developments in Grid Generation and Force Integration Technology for Overset Grids - W.M. Chan, NASA Ames Research Center

Adaptive Composite Overlapping Grids for Hyperbolic Conservation Laws - K.D. Brislawn, D.L. Brown, G.S. Chesshire and J.S. Saltzman, Los Alamos National Laboratory

Adaptive High-Order Godunov Projection Methods for the Incompressible Navier-Stokes Equations on Overlapping Grids - D.L. Brown and W.J. Rider, Los Alamos National Laboratory

A Structured/Unstructured Overset Grid Flow Solver for Helicopter Rotor Flows with Adaption - E.N. Duque, Army ATCOM, NASA Ames Research Center

Parallel Adaptive Mesh Refinement for Overlapping Grids - D. Quinlan, Los Alamos National Laboratory

A Navier-Stokes Chimera Code on the Connection Machine CM-5: Design and Performance - D.C. Jespersen and C. Levit, NASA Ames Research Center

Progress Report on High-Performance High-Resolution Simulations of Coastal and Basin-Scale Ocean Circulation - D.W. Barnette, J.M. Swisshelm, R. Tuminaro and C.C. Ober, Sandia National Laboratory

Tutorial: CMPGRD - G. S. Chesshire

CGINS: A Solver for the Incompressible Navier-Stokes Equations on Overlapping Grids - W.D. Henshaw, Los Alamos National Laboratory

An Algorithm for All Speed Flows - K. Pao, Los Alamos National Laboratory

Analysis of the Space Shuttle Ascent Aerodynamic Environment - R.J. Gomez and F.W. Martin,Jr., NASA Johnson Space Center

GBU-28 Pressure Port Analysis - B.A. Jolly, Sverdrup Technology Inc.; J.M. Brock,Jr., Test Wing, Eglin AFB; and L. Coleman, Tybrin Corp.

Analysis of the Re-Designed Space Shuttle APU Control Value - C.H. Campbell and T.C. Wey, NASA Johnson Space Center

REFERENCES

1. Meakin, R.: On the Spatial and Temporal Accuracy of Overset Grid Methods for Moving Body Problems. AIAA Paper 94-1925, June 1994.

2. Kao, K. H. and Liou, M. S.: Direct Replacement of Arbitrary Grid-Overlapping by Nonstructured Grid. AIAA Paper 95-0346, January 1995.

3. Johnson, R. A. and Belk, D. M.: A Multigrid Approach to Embedded-Grid Solvers. AIAA Paper 93-0769, January 1993.

4. Rogers, S. E. and Pulliam, T. H.: Accuracy Enhancements For Overset Grids Using a Defect Correction Approach. AIAA Paper 94-0523, January 1994.

5. Chesshire, G., and Henshaw, W. D.: Composite Overlapping Meshes for the Solution of Partial Differential Equations. JCP, vol. 90, pp. 1-64, 1990.

6. Wey, T. C.: Development of a Mesh Interface Generator for Overlapped Structured Grids. AIAA Paper 94-1924, October 1994.

7. Lijewski, L. E. and Suhs, N.: Chimera-Eagle Store Separation. AIAA Paper 92-4569, August 1992.

8. Chan, W. M. and Buning, P. G.: A Hyperbolic Surface Grid Generation Scheme and its Applications. AIAA Paper 94-2208, June 1994.

9. Maple, R. C. and Belk, D. M.: Automatied Set Up of Blocked, Patched, and Embedded Grids in the Beggar Flow Solver. Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, ed. N. P. Weatherill et al., 1994, Pine ridge Press, pp. 305-314.

10.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.

11.Samet, H.: Applications of Spatial Data Structures: Computer Graphics, Image Processing, and Other Areas, Addison-Wesley, 1989.

12.Samet, H.: Design and Analysis of Spatial Data Structures. Addison-Wesley, 1990.

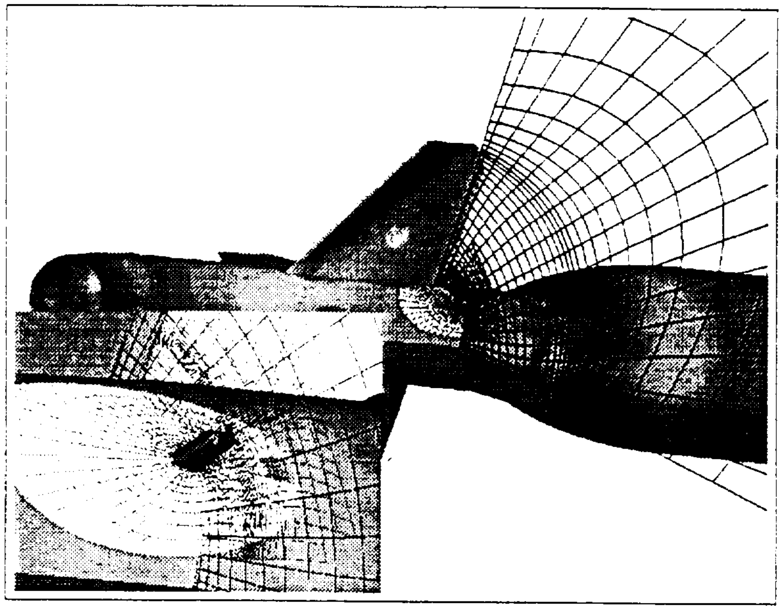

Figure 1: Use of an overset grid to add an appendage to an existing configuration.

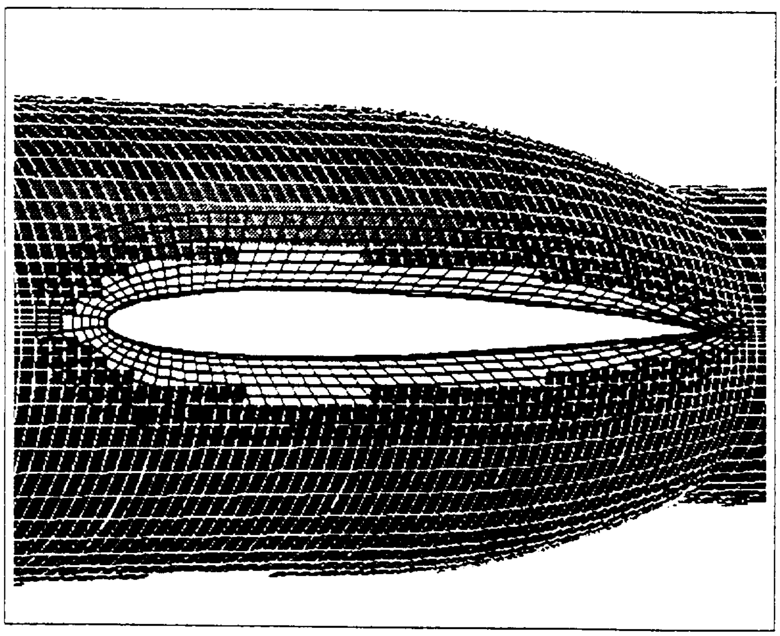

Figure 2: Example of problem area for force integration on intersecting surfaces.

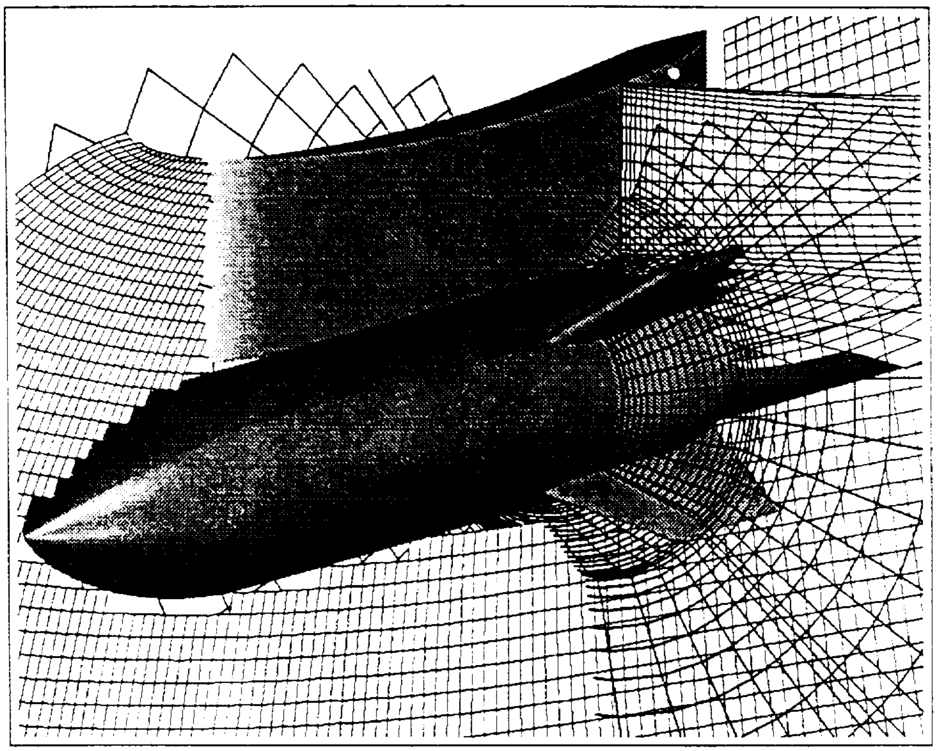

Figure 3: Beggar grid with blocked grid for finned store.

CARTESIAN GRID TECHNOLOGY