NASA Technical Memorandum 88193

On Applications of Chimera Grid Schemes To Store Separation

F. Carroll Dougherty, John A. Benek, and Joseph L. Steger

October 1985

National Aeronautics and Space Administration

NASA Technical Memorandum 88193

On Applications of Chimera Grid Schemes To Store Separation

F. Carroll Dougherty, Ames Research Center, Moffett Field, California
John A. Benek, Calspan Field Services, Inc., Arnold Air Force Station, Tennessee
Joseph L. Steger, Ames Research Center, Moffett Field, California

October 1985

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field California 94035

On Applications of Chimera Grid Schemes To Store Separation

F. Carroll Dougherty, Research Scientist
Applied Computational Fluids Branch MS 202A-14
NASA Ames Research Center, Moffett Field, CA 94035

John A. Benek, Principal Research Engineer
Computational Fluid Dynamics Section MS 600
Calspan Field Services, Inc. / AEDC Division
Arnold Air Force Station, TN 37389

Joseph L. Steger, Senior Staff Scientist
Fluid Dynamics Division MS 202A-1
NASA Ames Research Center, Moffett Field, CA 94035

Summary

A finite-difference scheme which uses multiple overset meshes to simulate the aerodynamics of aircraft/store interaction and store separation is described. In this chimera, or multiple mesh, scheme, a complex configuration is mapped using a major grid about the main component of the configuration, and minor overset meshes are used to map each additional component such as a store. As a first step in modeling the aerodynamics of store separation, two-dimensional inviscid flow calculations have been carried out in which one of the minor meshes is allowed to move with respect to the major grid. Solutions of calibrated two-dimensional problems indicate that allowing one mesh to move with respect to another does not adversely affect the time accuracy of an unsteady solution. Steady, inviscid three-dimensional computations demonstrate the capability to simulate complex configurations, including closely-packed multiple bodies.

I. Introduction

Today's high-speed aircraft are intended to carry stores either externally or semi-submerged, and are rated as to their ability to deliver these stores to their targets reliably and accurately, without loss of aircraft performance and agility. However, store-induced aerodynamic drag can significantly downgrade aircraft performance, while aerodynamic interaction may cause the released stores to scatter, run into each other, or impact the aircraft. It is necessary, then, to develop reliable methods to predict the aerodynamics of the store/airframe interaction.

It is difficult to test various carriage configurations and releases of stores in the windtunnel, because stores cannot be freely released, or, if sting mounted, can not move with complete freedom. Moreover, scaling effects from the model to true size are difficult to overcome. Flight testing is an option available for aircraft/store configurations, but without analytical and/or experimental results to rely on, flight tests can be unpredictable, expensive, and dangerous.

The development of a numerical simulation capability using finite-difference techniques would aid in the design and testing of aircraft carrying stores. A major obstacle to a computational approach, however, is the difficulty in developing effective discretization processes for complex configurations. While a single global mesh about an aircraft with stores can be generated, it is unlikely to be computationally efficient, especially for viscous flow simulation. Moreover, allowing the stores to move would require a new grid to be generated after each time step.

In lieu of using a single mesh for complex body configurations, multiple meshes can be used to map the overall configuration. The use of a multiple grid approach can yield better grid resolution, simplify the application of boundary conditions, and ease the task of grid generation. Some of the disadvantages to multiple meshes are that the flow solver must be modified, and that the bookkeeping that tracks the relationships among the meshes can be complicated. Two of the multiple mesh techniques currently emphasized in the literature are patched grids (refs. 1-3) and overset grids (refs. 4-9), although both could be considered as special cases of a general multiple mesh approach.

We have been working with a chimera grid scheme which uses overset meshes. In this approach, minor grids are overset on a major grid to completely model a configuration. These minor meshes are generated somewhat independently and are overset on the major grid without requiring any special mesh boundaries. As such, the minor meshes can be allowed to move with respect to the major grid.

The purpose of this paper is to demonstrate the feasibility of this chimera scheme for future application to the problem of aircraft/store interaction and separation. Unsteady flow results have been computed in two dimensions, where one component in a multiple mesh system has been moved with respect to the fixed system. In three dimensions, steady, inviscid solutions have been obtained for a generic wing/body/tail configuration, and for closely-packed multiple bodies that can ultimately be run as a TER (triple ejection rack) configuration.

A brief description of the overset chimera grid package is given in section II. Section III contains information about the flow solvers used in both the two- and three-dimension applications. Details about the combined flow-solver/grid-package

codes are given in section IV. The results are shown in section V, and concluding remarks follow in section VI.

II. Chimera Grid Management Scheme

Our mesh scheme is one in which a major grid stretches to the far field and is often generated about a main body element such as a wing. Minor grids are then overset on the major grid so as to resolve secondary features of the configuration such as stores, nacelles, or flaps. In general, the minor grids are overset on top of the major grid without requiring the mesh boundaries to join in any special way. With the use of such an overset mesh system the task of grid generation is greatly simplified because individual grids can be generated independently (provided they resolve the flow gradients) and then superimposed to form the overall mesh configuration. In this way, one can build up a grid system to treat complex configurations and avoid severe mesh distortion.

To illustrate the chimera grid technique an airfoil with a detached flap tucked underneath it is shown meshed in figure 1 using overset grids. Both the airfoil and the flap are shown with body-conforming O-grids, with the minor flap grid overset on the major, or airfoil, grid. Since the flap is an impermeable body, the points of the major grid that fall within the flap (or a curve of the minor grid that circumscribes the flap) are excluded or blanked from the flow field solution. The detailed view of figure 1 shows the points (marked with filled circles) on the major grid that are excluded from the solution because of the presence of the flap. The boundary of this "hole" in the major grid becomes the first of two interface boundaries that arise because of the use of overset meshes. Data for this hole boundary must be supplied from the solution contained on the overset flap grid. The outer boundary of the minor grid forms the second interface in this problem, and nearby points from the major grid are used to supply flow values along the minor grid outer boundary. The flow field solutions on each grid are connected by transferring boundary information across these two interfaces.

In the current codes, each grid is independently advanced in time. The boundary information, however, is not transferred simultaneously, but is updated with each time step in the following manner. Using figure 2 for illustration, the flow field on the major grid is time advanced using boundary data at time level n. The minor grid solution is then advanced with the flap surface and its outer boundary also held at time n. Then the flow variables at the hole boundary are updated with the information just solved for on the minor mesh (the dotted line on the minor grid). The effect of the flap is now imposed on the airfoil solution. The flow variables at the outer boundary of the minor grid are likewise updated with the information from the major grid (the dotted line), so that the flap feels the effect of the airfoil. Finally, the body surface and far field boundaries are updated in the usual manner.

All of the multiple grid bookkeeping (the determination of the interface boundaries and communication controls) is done automatically by the chimera grid management package. Once the grids are input to the code, the chimera grid package defines the relationships amoung the meshes, excludes or blanks out the points within another body to form holes, and finds the nearby points to be used for the transfer of infomation across the interfaces. Details about the bookkeeping procedures can be found in references 6-9.

Because of the overset grid technique, grids can be generated about relatively complex configurations. Because of the automatic chimera grid manangement package, one or more grids can move with respect to each other.

III. The Flow Solvers

The overset mesh scheme has been combined with implicit approximate factorization algorithms (refs. 10, 11, 12) for solving the Euler equations in both two and three dimensions (refs. 6-9). However, calculations in which one body moves with respect to another have only been carried out in two-dimensions.

In generalized curvilinear coordinates and nondimensionalized variables, the three-dimensional Euler equations in strong conservation law form can be written as

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = 0, \tag{1a}$$

$$\hat{Q} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \hat{E} = J^{-1}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U - \xi_t p \end{bmatrix},$$

$$\hat{F} = J^{-1}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V - \eta_t p \end{bmatrix}, \quad \hat{G} = J^{-1}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W - \zeta_t p \end{bmatrix}, \tag{1b}$$

where U, V, and W are unscaled contravariant velocities, for example,

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w.$$

The metric Jacobian is calculated as

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi, \tag{2}$$

and the metric terms ξ_x, ξ_y, etc are obtained from x_ξ, x_η, etc. in the usual way.

An implicit approximate-factorization finite-difference scheme for the three dimensional equations (1) can be written as

$$(I + h\delta_\xi \hat{A}^n)(I + h\delta_\eta \hat{B}^n)(I + h\delta_\zeta \hat{C}^n)\Delta\hat{Q}^n = -h(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n), \tag{3}$$

where $\Delta\hat{Q}^n = \hat{Q}^{n+1} - \hat{Q}^n$ and $h = \Delta t$ for first-order time accuracy. The three-dimensional flux Jacobian metrics $\hat{A}$, $\hat{B}$, $\hat{C}$ are defined in reference 12.

The spatial derivative operators ∂_ξ, ∂_η, and ∂_ζ are approximated with second-order accurate central finite-difference operators. On the implicit side of equation (3), the second-order central differences produce block-tridiagonal matrix operators $(I + h\delta_\xi \hat{A}^n)$, $(I + h\delta_\eta \hat{B}^n)$, and $(I + h\delta_\zeta \hat{C}^n)$, which must be inverted sequentially to obtain $\Delta\hat{Q}^n$. Additional details about the three-dimensional Euler code are given in references 9, 10, and 12.

In this initial research phase of the developing of a two-dimensional unsteady moving body code, a diagonalization version of the approximate factorization algorithm was used in a deliberate trade off of time accuracy for reduced computational work. By diagonalizing the block tridiagonal operators into scalar operators, the resulting system can be more efficiently solved. The "diagonalized" form of the algorithm is

$$T_\xi^n[I + h\delta_\xi \Lambda_\xi^n]\hat{N}[I + h\delta_\eta \Lambda_\eta^n]T_\eta^{-1}\Delta\hat{Q}^n = \hat{R}^n, \tag{4}$$

with

$$\Lambda_\xi = T_\xi^{-1}\hat{A}T_\xi \text{ and } \Lambda_\eta = T_\eta^{-1}\hat{B}T_\eta, \tag{5}$$

and $\hat{N} = T_\xi^{-1}T_\eta$. These matrices are shown in the references 10 and 13. This form of the algorithm is at best first order accurate in time, as the eigenvector matrices are functions of ξ and η. With the addition of numerical dissipation terms, this diagonalized algorithm reduces the block tridiagonal inversion to a series of 4x4 matrix multiplies and scalar tridiagonal or (depending on the dissipation terms, see reference 14) pentadiagonal inversions.

IV. Flow Solver - Chimera Interactions

As mentioned in the introduction, a flow solver must be modified to account for the use of multiple meshes and the "holes" in the grids that result because grid points of one mesh can fall within the body boundaries of another. These hole points must be blanked or excluded from the flow field solution.

Two minor modifications have been made to the flow solver to acommodate new boundary conditions and the blanked or excluded points in the meshes. A modular boundary condition routine was developed to set up the boundary values for each mesh, because different meshes in the same configuration might require different boundary treatment. For example, the $k = 1$ line in a major, or flow field, grid might represent a free stream boundary, while the $k = 1$ line on the minor grid might represent a solid body boundary. Interpolation and update routines were also added for the transfer of information across the various mesh interfaces.

The main change in the flow algorithm itself is the treatment of the hole boundaries. The hole information from the chimera grid package is stored in an array, here called IBLANK, which is defined for each point on each grid as

$$\text{IBLANK} = \begin{cases} 1, & \text{if a point is not blanked;} \\ 0, & \text{if a point is blanked.} \end{cases} \tag{6}$$

In the flow solver, each element in the left-hand-side matrix and right-hand-side vector is multiplied by the appropriate IBLANK value before the matrix is inverted. Using a simple 8x8 tridiagonal system for illustration with points 4 and 5 blanked, the resulting equation set would look like

$$\begin{bmatrix} b & c & & & & & & \\ a & b & c & & & & & \\ & a & b & c & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & a & b & c & & \\ & & & & a & b & c & \\ & & & & & a & b & \end{bmatrix} \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \\ \Delta Q_6 \\ \Delta Q_7 \\ \Delta Q_8 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ 0 \\ 0 \\ r_6 \\ r_7 \\ r_8 \end{bmatrix}, \tag{7}$$

with no changes computed for ΔQ_4 or ΔQ_5. The blanked values Q_4^{n+1} and Q_5^{n+1} are updated in the interpolation routine outside of the flow solver. Pentadiagonal matrices are treated in a similar manner. With this approach, no special routines or logic tests are required to exclude the blanked points from the flow field solution.

Unsteady Modifications. In steady-state applications of the overset grids, the flow solver and the chimera grid management package were run independently. The grid package was run once, and the IBLANK array and transfer information arrays were passed on to the flow solver. For an unsteady, moving body case where one body is free to move relative to a second, the grid package must be called each time the body and its grid move. For these unsteady cases, care must be taken to update the interface boundaries in a time-accurate fashion.

In order to simulate, for example, the flow about a wing with a detached moving store beneath it, the main routine of the unsteady code was constructed in a modular fashion with four main units that are called each time step. These units are: the chimera grid management package, a flow solver routine adapted for multiple meshes, a trajectory routine (for the present, a simple, predetermined motion), and a package to update the interface boundaries caused by the overset meshes. Each of these is run somewhat independently, with interactions among them limited to passing arrays of information. With this modular approach, it is easy to replace one unit with another, suitably modified program without affecting the rest of the unsteady code.

Figure 3 illustrates the interactions among the four modules. At the beginning of each new time step, the trajectory routine is called to calculate the new spatial position of the minor grid. The chimera grid management package redefines the relationship of the grids by finding new blanked points and locating and storing new nearby points to update the interface boundaries. If necessary, new grid metrics are formed. The flow field solution for each mesh is then advanced to the new time level, $n + 1$, and the interface boundaries are updated for the next time step. To facilitate output of interim flow field solutions, all information at the start of the sequence is at the old time level, and, by the end of the sequence, all information has been updated to the new time step values.

V. Results

The overset grid procedure has been used to compute two-dimensional inviscid flows about geometries where one body is moved with respect to the rest of the configuration. It has also been applied to steady, inviscid three-dimensional complex configurations. To verify the two-dimensional overset mesh code in the time-dependent mode, the transonic flow about an airfoil undergoing an oscillating plunge was run. A multiple mesh case with a minor O-grid wrapped around a NACA 64A010 airfoil overset on a Cartesian major grid is compared to a single mesh case with a larger O-grid about the same airfoil. The airfoil and its mesh move with a velocity of

$$\dot{y}_a = -A\omega\cos(\omega t), \tag{8}$$

where

$$A = 0.0436\,\text{chord}, \quad k = .4, \quad \text{and} \quad \omega = kU_\infty/\text{chord}.$$

The maximum upward excursion of the airfoil from zero is $\approx 4\%$ of the airfoil chord. Time histories of the lift coefficient are shown in figure 4. Each run was started from steady state ($M_\infty = 0.8$, $\alpha = 0^\circ$). The 720 step per cycle single mesh curve shown in figure 4 is considered to be the correct range for C_l (ref. 15). A comparison between the single and multiple mesh time histories shows that the two are in phase, but that the multiple mesh results have a slight error in amplitude. This error in time accuracy can be improved by decreasing the time step, and is probably due to time lagging the outer boundary of the minor grid and possible errors in interpolation. The error is small, and the results demonstrate the capability of the unsteady code to simulate the flow over a configuration with one component moving.

Two transonic store separation simulations using predetermined paths were computed with the two-dimensional unsteady code. Both computations were started from the same steady-state solution. The initial grid arrangement for the store separation simulations is shown in figure 5. Both bodies are meshed with body-conforming O-grids. The airfoil body is a 10% thick ellipse with a chord length of 1. The minor store body is a 10% ellipse reduced by 60%. Filled circles mark the blanked points forming the hole in the major grid due to the minor body and the hole in the minor grid caused by its intersection with the airfoil. The empty circles denote the nearest points on the major grid used to update the minor grid outer boundary. The minor body nose is initially positioned at $x/c = .3$ and $y/c = -.15$. The flow field solution for the initial position is illustrated with Mach contours and is shown in figure 6. The free stream Mach number for this case is $M_\infty = .7$, and both bodies are at 0° angle of attack. In the common overlap region, within the outer boundary of the minor grid and the hole in the major grid, both sets of contours are shown. The contours at the hole boundaries and the outer boundary of the minor grid match well, showing that the communication between the grids across the interface boundaries is quite good. The flow channeled between the two bodies behaves somewhat like flow in a two-dimensional nozzle, and there is a strong shock between the major and minor bodies, as well as some spillover on the lower surface of the leading edge of the store. There is some mismatch of contours at the shock caused by the interpolation between the two dissimilar meshes at the hole boundary. The slight discrepancy between the contours in the common areas of the flow field is partially attributed to the plotting package, as again the contours are being calculated on two dissimilar meshes.

The first store simulation case has a predetermined downward translation and rotation. The downward rate of .01 corresponds to a drop of .75 chordlengths per second for an average realtime store separation profile. The pitchdown rotation on the store is approximately 2° per 100 time steps ($\Delta t a_\infty/c = 0.025$). Figures 7 and 8 show the grid positions and Mach contours for the store case after 400 time steps. The store body has dropped about 10% of chord and has an angle of attack of 8°. Again, the filled circles mark the blanked points and the open circles denote the nearest points for update of the minor grid outer boundary. In figure 8, the Mach contours reflect the movement of the store. A strong shock occurs on the lower surface of the store due to the 8° downward pitch to the freestream, and the shock between the two bodies is much weaker. The mismatch of contours at the outer boundary of the minor mesh is attributed to the interpolation and the time lag on that boundary. Figures 9 and 10 show the grid positions and Mach contours for this case after 600 time steps. The store has dropped another 5% of chord and is now at a 12° angle to the freestream. The shock between the two bodies has deteriorated to a weak shock at the tail on the upper surface of the store; it no longer spans the distance between the bodies.

The second store separation simulation involves a predetermined path in which the store first drops down, then rotates back up. If the calculation were continued, the store would hit the wing. The first two figures, 11, and 12, show the grid postions and Mach contours after 100 time steps (with the same time step as before: $\Delta t a_\infty / c = 0.025$) The store is dropping at the same rate as for the first case, with a downward rotation of the trailing edge of 2° per 100 time steps. The flow field (figure 12) shows the strong shock between the two bodies. After the 400th time step, the store starts to move upward at the same rate as before, but the trailing edge of the store continuing to rotate downward. The grid positions after 600 time steps are shown in figure 13, with the corresponding flow field solution shown in figure 14 The shock between the bodies is stronger than before and has moved forward.

Steady, inviscid three-dimensional flow about a generic wing/body/tail configuration has been computed and compared to experimental data The configuration is comprised of an ogive-cylinder fuselage with a wing and tail formed by swept, untapered wings with NACA 0012 airfoil cross-sections. Four grids are used to mesh this configuration Figure 15 shows the configuration with the surface grids for the fuselage, wing and tail, along with the outer boundaries for the wing and tail meshes. These outer boundaries did not intersect. The outermost of the four grids is used to resolve the solution from the near- to the farfield and is a warped hemispherical shell whose polar axis is coincident with the fuselage centerline. The mesh contains 37,000 points (74 x 25 x 20) and extends from 9 to 51 radii from the fuselage. The fuselage mesh is the second mesh in this configuration, another warped, hemispherical shell whose inner boundary is the fuselage surface. The grid contains 77,700 points distributed as 74 x 35 x 30 and extends to 11.5 radii. The outer boundary of the fuselage grid overlaps the inner boundary of the outer mesh by 2.5 radii. The wing and tail grids are warped, cylindrical meshes whose axes are directed along the wing- and tailspans. The end surfaces containing the roots of the wing and tail are coincident with the fuselage surface. The wing mesh has 27,720 points (66 x 28 x 15) with 20 spanwise points on the wing surface, and the tail grid contains 15,120 points distributed as 56 x 18 x 15 with 10 points along the tailspan. The composite mesh consists of 157,540 points. A solution was computed for this configuration at a Mach number of 0.9 at an angle of attack of 2°. Figure 16 shows the comparison of experimental data (ref. 16) and the computation.

As an example of another three-dimensional steady-state simulation, the flow about a generic three-body TER-like configuration was computed. The configuration consists of three ellipsoidal bodies in a triangular arrangement (figure 17) The grids of the two small bodies have major and minor axes one half those of the larger body. The two smaller bodies are embedded in the mesh of the larger, as indicated in figure 17. The figure shows the surface grids for all three bodies, and the warped outer shells of the two minor meshes. All the grids are spherical. The mesh of the large ellipsoid has 26,250 points (30 x 35 x 25); the two smaller meshes each have 15,750 points (30 x 35 x 15). The composite mesh has 57,750 points. Surface Mach contours, computed at $M_\infty = 0.8$ and $\alpha = -2°$, are shown in figure 18.

VI. Concluding Remarks

The chimera overset grid technique has been successfully demonstrated for the simulation of flow about complex configurations. While restricted here to two dimensions, the feasibility of numerically simulating transonic store separation has been carried out. Moreover, preliminary steady-state simulations about generic wing/body/tail and TER configurations show that the technique can be applied in three dimensions as well. At least for simple configurations such as the TER, the simulation of unsteady three-dimensional store separation appears feasible in the near term

References

1. Lasinski, T. A., Andrews, A. E., Sorenson, R. L., Chaussee, D. S., Pulliam, T H., and Kutler, P., "Computation of the Steady Viscous Flow Over a Tri-Element 'Augmentor Wing' Airfoil," AIAA Paper No. 82-0021, Jan. 1982.
2. Rai, M. M., "An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculation," AIAA Paper No. 85-0488, Jan. 1985.
3. Hessenius, K. A. and Rai, M M., "Applications of a Conservative Zonal Scheme to Transient and Geometrically Complex Problems," AIAA Paper No. 84-1532, June 1984.
4. Atta, E. H., "Component Adaptive Grid Interfacing," AIAA Paper No. 81-0382, 1981.
5. Atta, E. H. and Vadyak, J. A., "Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows About Aircraft Configurations," AIAA Paper No. 82-1017, 1982.
6. Dougherty, F. C., "Development of a Chimera Grid Scheme with Applications to Unsteady Problems," Stanford University Dissertation, Dept. of Aeronautics and Astronautics, June, 1985.
7. Benek, J. A., Steger, J. L., and Dougherty, F. C.; "A Flexible Grid Embedding Technique with Applications to the Euler Equations," AIAA Paper No. 83-1944, Danvers, Mass., July 1983.
8. Steger, J. L., Dougherty, F. C., and Benek, J. A.; "A Chimera Grid Scheme," ASME Mini-Symposium on Advances in Grid Generation, Houston, Texas, June, 1983.
9. Benek, J. A., Buning, P. G., and Steger, J. L.; "A 3-D Chimera Grid Embedding Technique," AIAA Paper No. 85-1523-CP, Cincinnati, Ohio, July, 1985.
10. Pulliam, T. H., "Euler and Thin Layer Navier-Stokes Codes: ARC2D, ARC3D," Computational Fluid Dynamics User's Workshop, The University of Tennessee Space Institute, UTSI Publication No. E02-4005-023-84, March 12-16, 1984.
11. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries," AIAA Journal, Vol. 16, No. 7, July 1978, pp. 679-686.

12. Pulliam, T. H., and Steger, J. L., "On Implicit Finite Difference Simulations of Three Dimensional Flows," AIAA Journal, Vol. 18. No. 2, February 1980, pp. 159-167.
13. Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," Journal of Computational Physics, Vol. 39, No. 2, February, 1981.
14. Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," AIAA Paper No. 85-0438, Jan., 1985
15. Magnus, R. J., "Computational Research on Inviscid, Unsteady, Transonic Flow Over Airfoils," Office of Naval Research, CASD/LVP 77-D10, January 1977.
16. Parker, P. L. and Erickson, J. C , Jr., "Status of Three-Dimensional Adaptive Wall Test Section Development at AEDC," AIAA Paper No. 84-0624, January 1984.

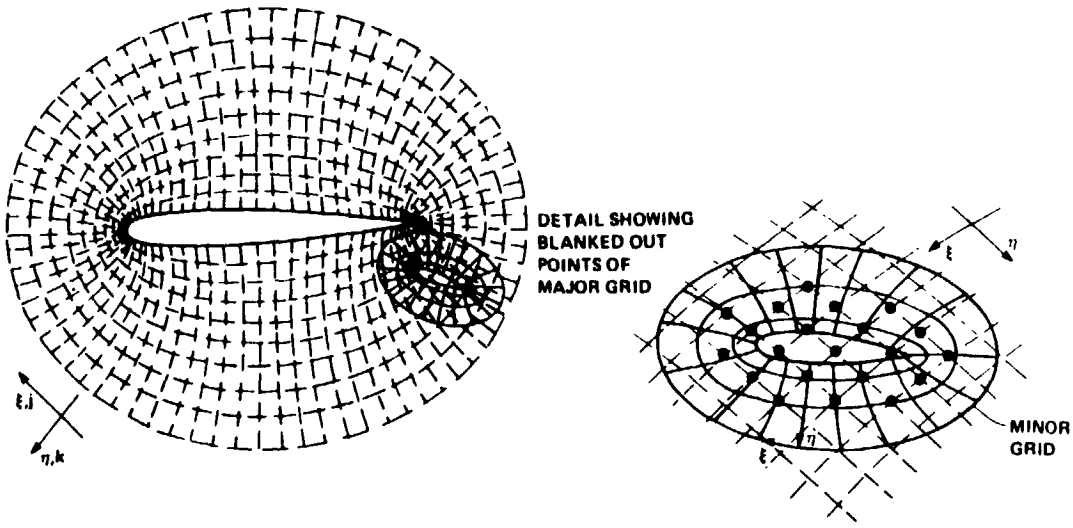

Figure 1. Overset grids for an airfoil with flap.

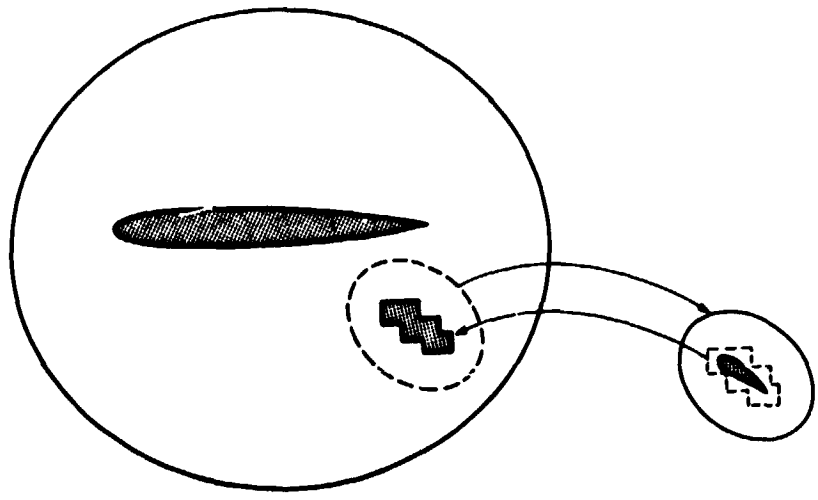

Figure 2. Transfer of information between the grids.

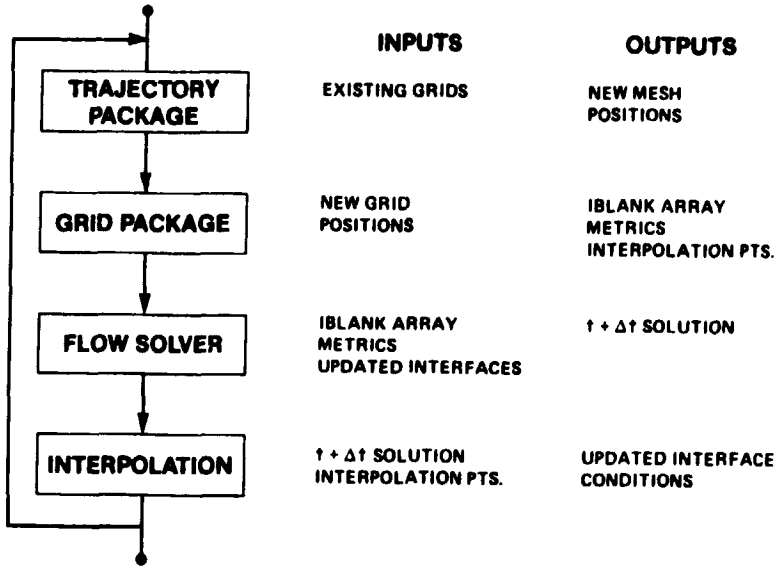

Figure 3. Calling sequence for unsteady code main routine.

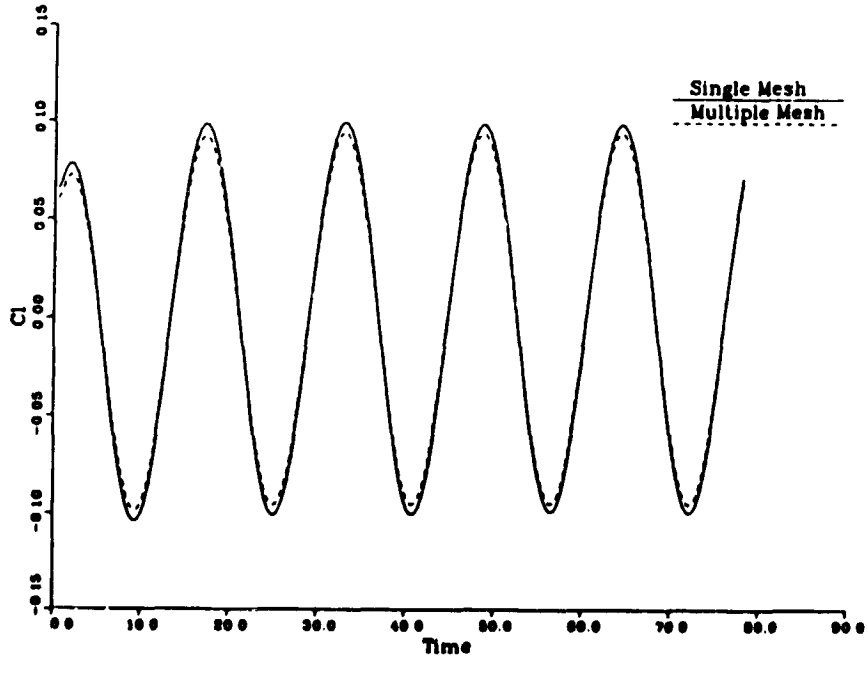

Figure 4. Comparison of single mesh and multiple mesh lift coefficient time histories; DT = 720 steps per cycle.

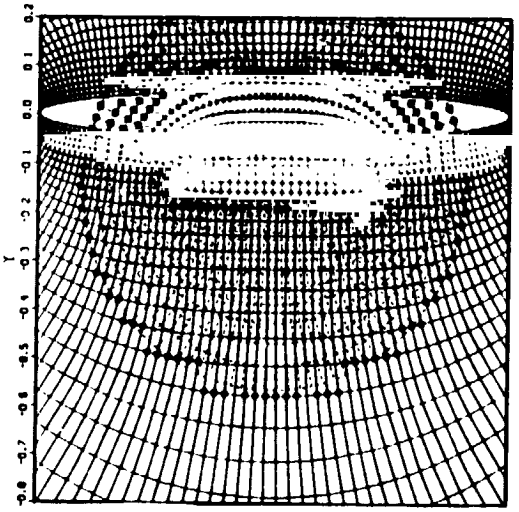

Figure 5. Initial position of overset grids for store separation simulation.

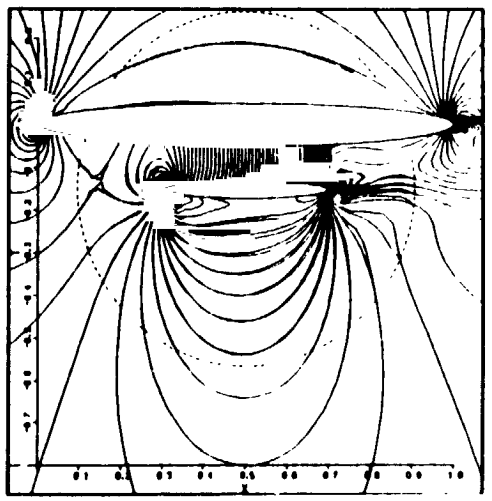

Figure 6. Mach contours at the initial position for the store separation simulation. ($M_\infty = .7$, $\alpha = 0$)

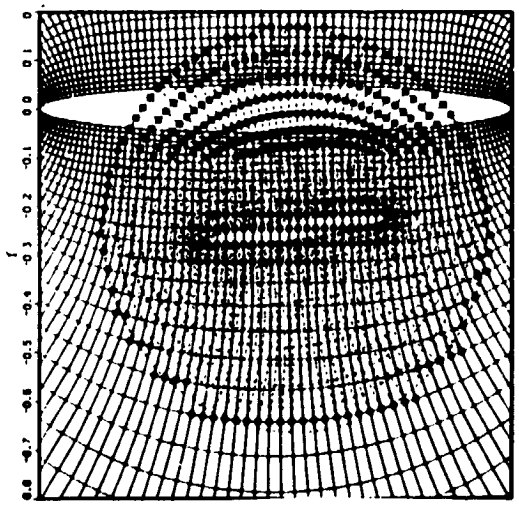

Figure 7. Overset grid positions after 400 time steps for first store separation simulation.

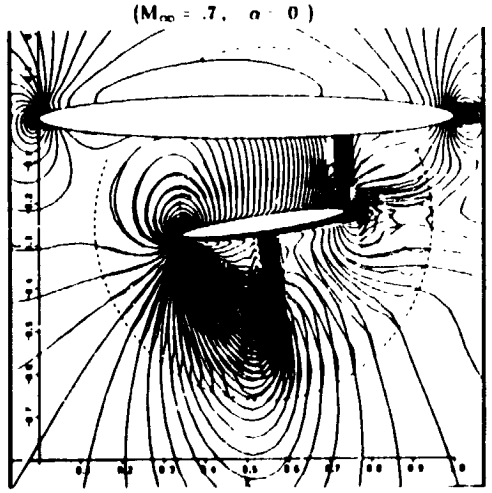

Figure 8. Mach contours for first store simulation after 400 time steps. ($M_\infty = .7$, $\alpha = 0°$)

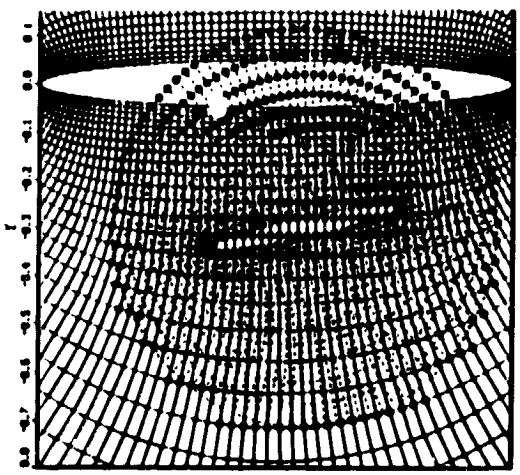

Figure 9. Overset grid positions after 600 time steps for first store separation simulation.

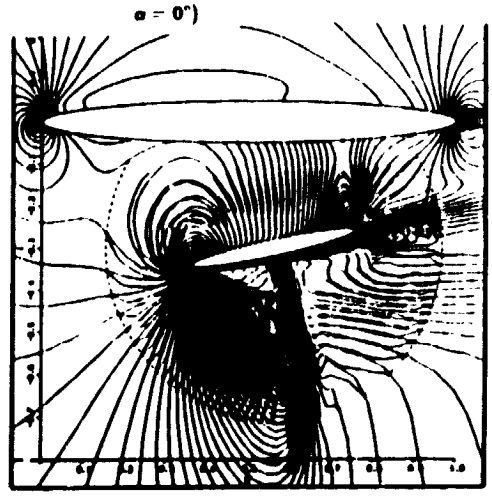

Figure 10. Mach contours for first store simulation after 600 time steps. ($M_\infty = .7$, $\alpha = 0°$)

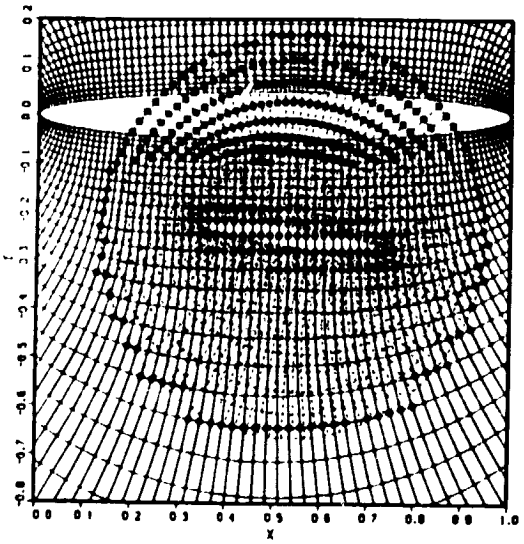

Figure 11. Overset grids for second store separation simulation after 400 time steps.

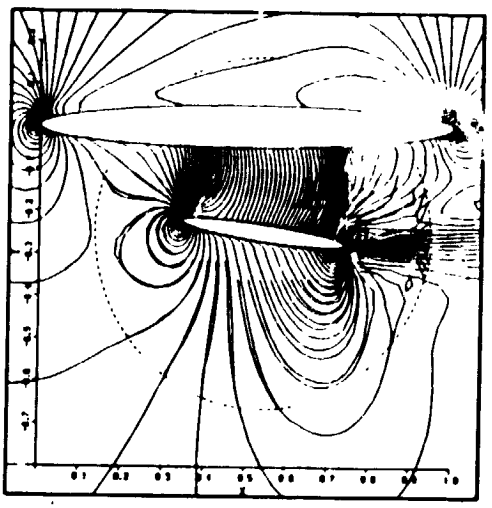

Figure 12. Mach contours for second store separation case after 400 time steps. ($M_\infty = .7$, $\alpha = 0°$)

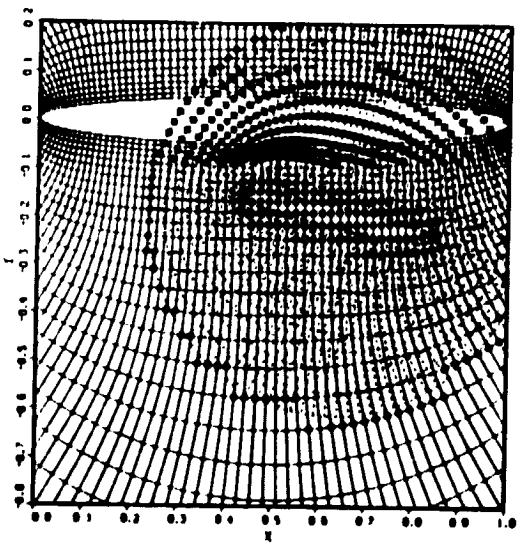

Figure 13. Overset grid positions for second store simulation after 600 time steps.

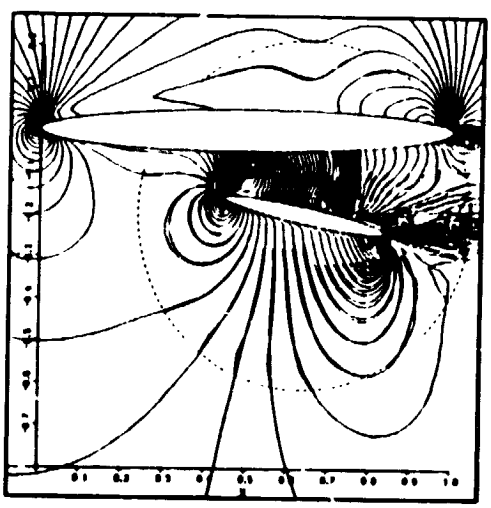

Figure 14. Mach contours for second store simulation after 600 time steps. ($M_\infty = .7$, $\alpha = 0°$)

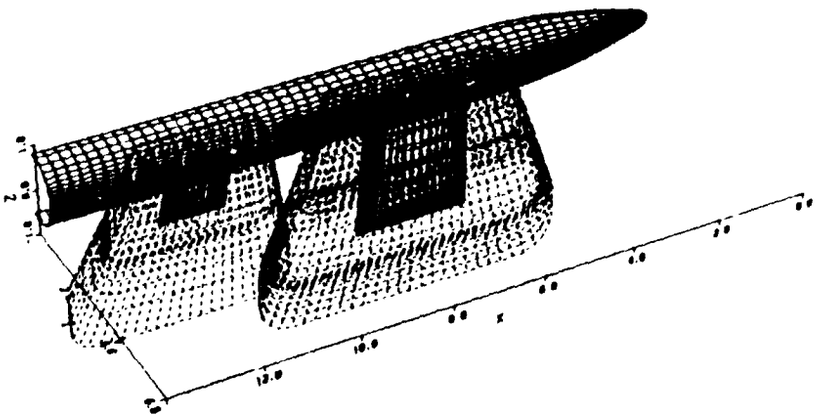

Figure 15. Three-dimensional wing/body/tail overset grid configuration.

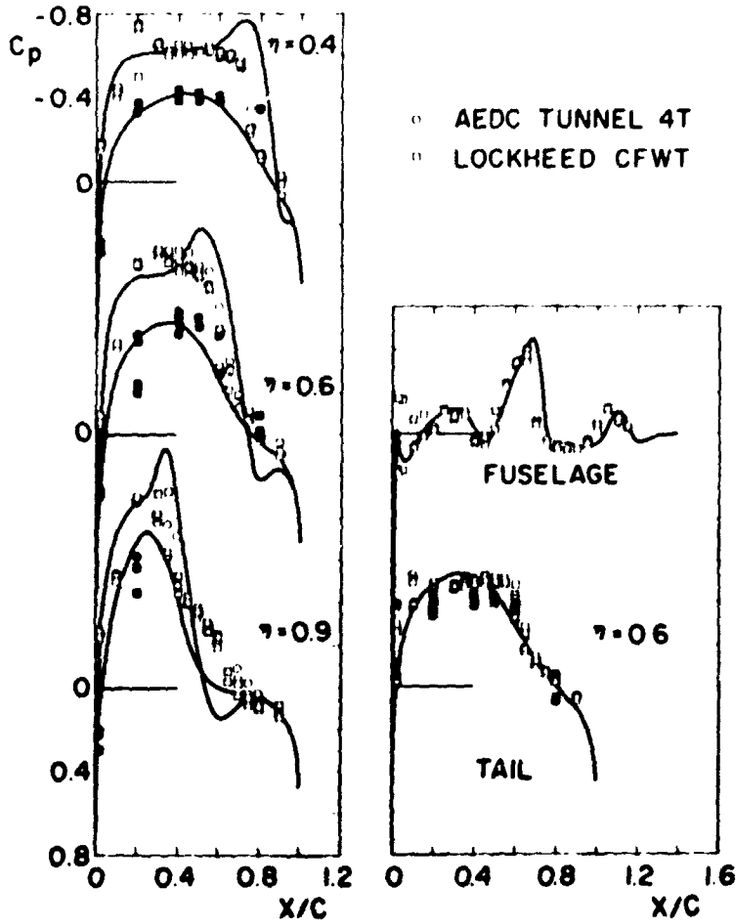

Figure 16. Wing/body/tail solution at $M_\infty = 0.9$ and $\alpha = 2°$ (open symbols, upper surface; solid symbols, lower surface).

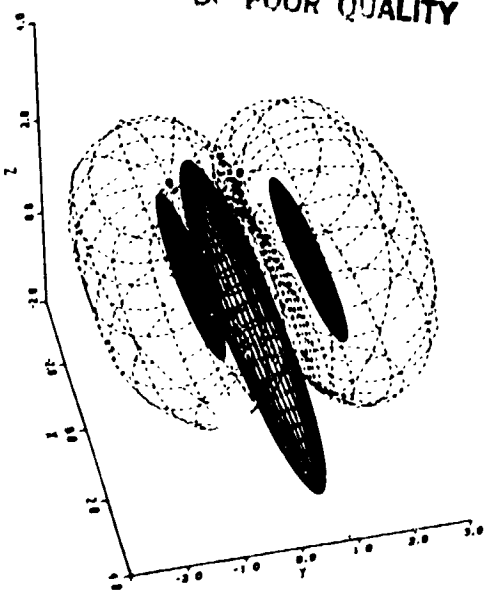

Figure 17. Three-dimensional multiple ellipsoids with overset grids.

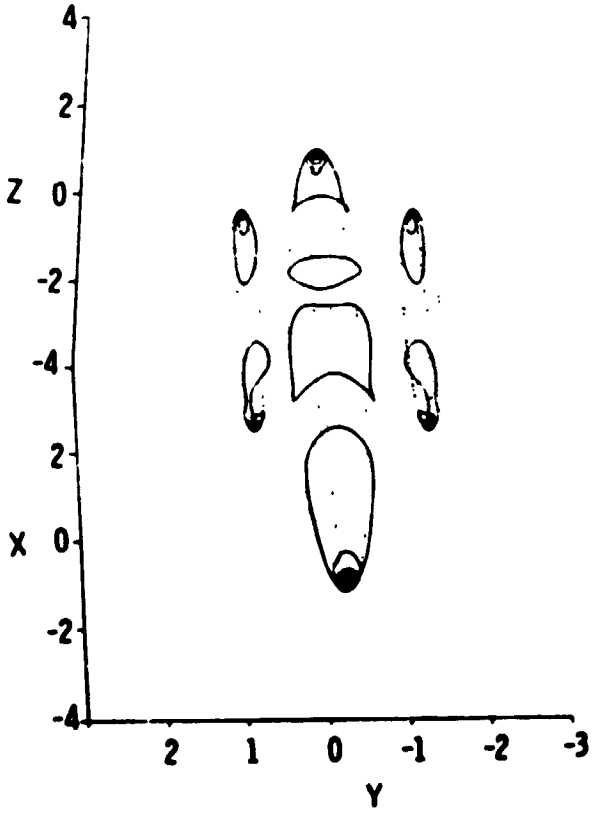

Figure 18. Mach number contours on the surface of the ellipsoids. (M_∞ = .8, α = -2°)

1 Report No.	2. Government Accession No.	3 Recipient's Catalog No.
NASA TM-88193		

4. Title and Subtitle	5. Report Date
ON APPLICATIONS OF CHIMERA GRID SCHEMES TO STORE SEPARATION	October 1985
	6. Performing Organization Code

7 Author(s)	8. Performing Organization Report No.
F. Carroll Dougherty, John A. Benek (Calspan Field Services, Arnold Air Force Station, TN), and Joseph L. Steger	86045
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Ames Research Center Moffett Field, CA 94035	
	13. Type of Report and Period Covered
12 Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14 Sponsoring Agency Code 505-31-01

15 Supplementary Notes

Point of Contact: F. Carroll Dougherty, Ames Research Center, MS 202A-1, Moffett Field, CA 94035, (415) 694-6415 or FTS 464-6415

16 Abstract

A finite-difference scheme which uses multiple overset meshes to simulate the aerodynamics of aircraft/store interaction and store separation is described. In this chimera, or multiple mesh, scheme, a complex configuration is mapped using a major grid about the main component of the configuration, and minor overset meshes are used to map each additional component such as a store. As a first step in modeling the aerodynamics of store separation, two-dimensional inviscid flow calculations have been carried out in which one of the minor meshes is allowed to move with respect to the major grid. Solutions of calibrated two-dimensional problems indicate that allowing one mesh to move with respect to another does not adversely affect the time accuracy of an unsteady solution. Steady, inviscid three-dimensional computations demonstrate the capability to simulate complex configurations, including closely-packed multiple bodies.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Store separation Computational fluid dynamics Overset grid schemes Unsteady Euler computations	Unlimited Subject category - 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	13	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161