JOURNAL OF AIRCRAFT
Vol. 37, No. 6, November–December 2000

New Fully Automated Procedure for the Prediction of Store Trajectory

Seungsoo Lee* and Minwoo Park†
Agency for Defence Development, Taejon 305-600, Republic of Korea
Kum Won Cho‡
KORDIC Supercomputing Center, Taejon 305-600, Republic of Korea
and
Jang Hyuk Kwon§
Korea Advanced Institute of Science and Technology, Taejon 305-701, Republic of Korea

A new fully automated store-trajectory simulation code has been developed. The method couples a highly robust computational fluid dynamics method and a cut-paste algorithm for Chimera domain decomposition to enhance the efficiency of store-trajectory simulation. The time-step size limitation caused by grid movement is relaxed with the interpolation of solutions in the previous time step. A treatment for orphan cells is also devised to improve the robustness of the method. The computational results show that the current method is capable of simulating store separation without user interruption and that a time-step size of 0.005 s, which is more than three times larger than ever reported, can be used for the Eglin Wing/Pylon/Store separation problem.

Nomenclature

C_p	= pressure coefficient
C_x, C_y, C_m	= axial, normal, and pitching moment coefficients
$\boldsymbol{D}$	= diagonal matrix, $(\Delta\tau/\Delta t + 1)\,\boldsymbol{I}$
$\hat{\boldsymbol{F}}$	= generalized flux vector, $\hat{i}\boldsymbol{E} + \hat{j}\boldsymbol{F} + \hat{k}\boldsymbol{G} - \boldsymbol{Q}\xi$
$\tilde{\boldsymbol{F}}$	= numerical flux vector
$\boldsymbol{I}$	= identity matrix
I_{xx}, I_{yy}, I_{zz}	= inertia of the store in x, y, z directions
i, j, k	= grid indices
k_r	= blanking index
L	= characteristic length of the store
$\boldsymbol{L}_i$, $\boldsymbol{L}_j$, $\boldsymbol{L}_k$	= implicit operators
m	= mass
$\hat{n}$	= outward normal vector
$\boldsymbol{Q}$	= conservative variables
$\boldsymbol{R}$	= residual
t	= time
U	= normal velocity components to ∂V in the moving coordinate, $\hat{n}\cdot(\boldsymbol{v} - \boldsymbol{\xi})$
u, v, w	= velocity components in x, y, z directions
V	= control volume
$\boldsymbol{v}$	= velocity vector
x, y, z	= spatial coordinates
∂V	= surface of the control volume
θ	= pitch angle
$\boldsymbol{\xi}$	= grid velocity
ρ, p, e	= density, pressure, total energy
τ	= fictitious time
ϕ	= circumferential angle of the three-dimensional store

Subscripts

t	= time
∞	= freestream

Introduction

SINCE the Chimera grid technique[1] was introduced to the computational fluid dynamics (CFD) community, it has been gaining in popularity not only because grid generation over complex geometries is relatively simple compared to the multiblock grid generation technique but because the Chimera method is the most successful method of handling the relative movement of multiple bodies. The Chimera grid technique is naturally suitable for moving-body problems. One moving-body problem frequently encountered in aerodynamic applications is the trajectory prediction of a store separating from a parent aircraft.

There have been several reports that successfully predict the trajectory of the well-known Eglin Wing/Pylon/Store problem.[2–5] Recently, ripple release simulations of multiple stores have been reported.[6–8] However, with CFD trajectory prediction is not a routine engineering application. The major stumbling block in the path of this goal comes from the fact that most Chimera methods rely on interactive approaches for Chimera hole cutting.[9,10] The computations, furthermore, are often hindered by the lack of efficiency and robustness of the CFD tools. In this paper efforts are presented toward this goal. These efforts include implementing an automated Chimera hole cutting method, a cut-paste algorithm,[11] combined with a two-step donor cell search. The cut-paste algorithm is the advancing front technique for Chimera hole cutting. Solid walls or nonpenetrable boundaries are used as the initial hole-cutting boundary so that user input is minimized. With this method Chimera hole cutting can be done automatically for unsteady applications. Also, a two-step donor cell search ensures a fast and reliable search. As stated by Lijewski,[4] computation often fails because of the presence of orphan cells. A proper handling of orphan cells without loss of accuracy is a must for full unsteady simulations. Furthermore, the time-step size is limited by the presence of cells that are initially hole cells and become normal cells as the grid moves without being updated. Methods are devised to handle such problems to enhance the robustness of the Chimera method. In addition, a highly robust CFD method[11] is used for aerodynamic forces and moment calculations.

The objective of the paper is to enhance the efficiency and robustness of the computational and Chimera methods so that the store

Presented as Paper 99-3131 at the AIAA 17th Applied Aerodynamics Conference, Norfolk, VA, 28 June–1 July 1999; received 8 October 1999; revision received 25 February 2000; accepted for publication 2 March 2000.

*Senior Researcher, P.O. Box 35-3, Yusong. Member AIAA.
†Principal Researcher, P.O. Box 35-3, Yusong. Member AIAA.
‡Senior Researcher, P.O. Box 122, Yusong.
§Associate Professor, Department of Aerospace Engineering, 373-1 Kusong-Dong, Yusong-Gu. Senior Member AIAA.

prediction method can be used as an engineering tool in the near future. The trajectory computations of the Eglin Wing/Pylon/Store and ripple release from TER (Triple Ejection Rack) are presented showing that the current method satisfies the objective of this study.

Chimera Hole Cutting

Most existing Chimera methods need user-specified hole cutting surfaces for the Chimera hole cutting. In PEGASUS,[9] for example, the user must specify the hole-cutting surfaces as well as the hole grid, whereas in DCF3D[10] the user has to define the hole-cutting surfaces by combinations of simple analytical shapes like cylinders and spheres. Recently, research has focused on the automation of the Chimera hole cutting. The Beggar code,[12] for example, automatically determines the hole-cutting surfaces from user-specified solid walls. Also, Wey developed an automated method[13] in which the initial hole-cutting surfaces advance outward to the final hole-cutting surfaces, which meet certain criteria. The advancing distance is computed based on the relative distances between the fronts in the domain and ranges of the grid points.

In this paper a new Chimera domain decomposition algorithm[11] is developed. It is a two-step method, consisting of the initial hole cutting and the cut-paste algorithm based on Wey's method. Solid walls or nonpenetrable boundary condition, which are also used for the flow solver, are used for the initial hole cutting. The hole cutting is performed by applying a zone of interference algorithm and a hole map algorithm. During the second step, the hole-cutting surfaces march outward along the grid line until there is no cell to be cut. Then, the paste phase adds two layers of interpolation cells outside the fringe cells. This allows not only minimization of the overlapped region, but also control over how much the grids are overlapped. Because the advantage of the discrete nature of the grid is considered, the number of iterations needed is less than that of Wey's algorithm. With the iterative process, moreover, the overlapped region is placed midway between bodies where the solution gradient is small compared to the region near the solid wall, which can reduce the error associated with trilinear interpolation especially for viscous flow computations. There are two alternatives in applying the method: a manual mode in which only the first step is applied and an automatic mode in which both the two steps are applied. A detailed description of the cut-paste algorithm can be found in Ref. 11.

Donor Cell Search

For the donor cell search a two-step search method, a stencil walk[14] followed by a gradient search,[15] is adopted. The stencil walk finds the nearest cell to a given interpolation cell by comparing the distance between the interpolation cell and the possible donor candidate cells. When the grid is severely skewed, the stencil walk fails to find the correct donor cells. The gradient search, on the other hand, checks whether the interpolation cell actually lies inside the cube that is made with the eight donor candidate cells by computing the isoparametric coordinates. This two-step search, therefore, guarantees the success of the donor cell search. The gradient search rarely needs more than two iterations. Furthermore, the interpolation coefficients are the byproduct of the gradient search. This donor cell search can be performed across the block interface boundaries between grids. For the multiple overlapped region the donor cell is determined from among possible donor cells according to a user-specified preference list. For unsteady problems the search cost can be greatly reduced by using the donor cells at the previous time step as the initial search guess.

Interpolation and Orphan Cells

Trilinear interpolation is used to communicate solutions across the grid system. The interpolation coefficients, as mentioned in the preceding section, are computed during the second step of the donor cell search, the gradient search. The interpolation is done after updating the solutions of every grid component. One or some of the donor cells for a given interpolation cell are often found to be hole cells, which are known as orphan cells. It is possible to remove the orphan cells by adding more grid points. However, this can increase the number of grid points inhibiting computations. Furthermore, one cannot know beforehand whether or not the assembled grid system has orphan cells. The Beggar code,[12] for example, does not allow orphan cells to occur. Once an orphan cell does occur, one must add more grid points near the problematic region or regenerate the grid system and then start the solution process all over again. In this paper, on the other hand, the occurrence of a limited number of orphan cells is allowed. The solutions of the hole cell are replaced with those of the nearest cell among eight donor cells. If all of the donor cells are found to be hole cells, the solutions of the nearest normal cell are used to replace the solutions of the eight donor cells. This procedure lowers the formal order of accuracy of the trilinear interpolation. Because the number of orphan cells is usually far less than that of the valid interpolation cells, however, the loss of accuracy is found to be minimal.

Flow Solver

The compressible Euler equations can be written in integral form over a control volume $V(t)$ moving with velocity $\boldsymbol{\xi}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} Q\,\mathrm{d}V + \int_{\partial V(t)} \hat{F}\,\mathrm{d}S = 0 \tag{1}$$

with

$$Q = (\rho, \rho u, \rho v, \rho w, e)^T \tag{2}$$

$$\hat{\boldsymbol{F}} = (\hat{i}E + \hat{j}F + \hat{k}G - Q\boldsymbol{\xi})\cdot\hat{n} = \begin{bmatrix} \rho k \\ \rho u k + p n_x \\ \rho v k + p n_y \\ \rho w k + p n_z \\ (e+p)k + p\xi_t \end{bmatrix} \tag{3}$$

also,

$$k = (\boldsymbol{v} - \boldsymbol{\xi}) = -\xi_t + n_x u + n_y v + n_z w \tag{4}$$

$$\xi_t = \boldsymbol{\xi}\cdot\hat{n} = \xi_x n_x + \xi_y n_y + \xi_z n_z \tag{5}$$

where Q, $\boldsymbol{\xi}$, and $\hat{n}$ are the conservative variables, the grid velocity, and the outward normal vector, respectively.

This system of Euler equations is discretized using the finite volume method in conjunction with Roe's approximated Riemann solver.[16] MUSCL extrapolation[17] of primitive variables is used to obtain second-order spatial accuracy, whereas Van Albada's limiter[18] or the minmod limiter is used to maintain the total-variation-diminishing property near shocks. Also, the entropy fix

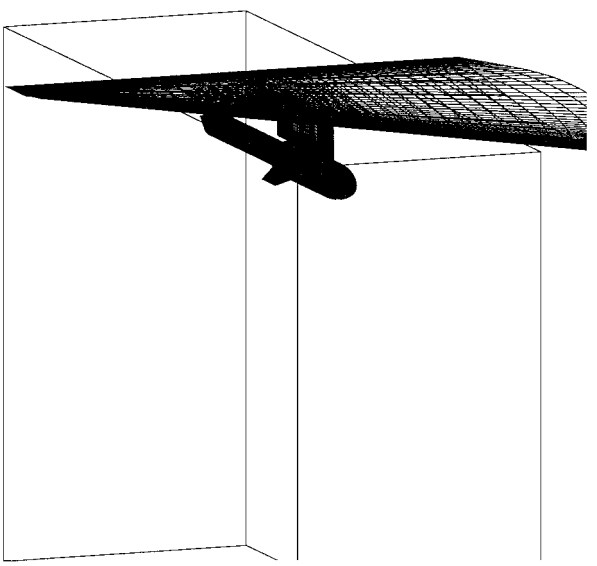

Fig. 1 Surface grid of Eglin Wing/Pylon/Store case.

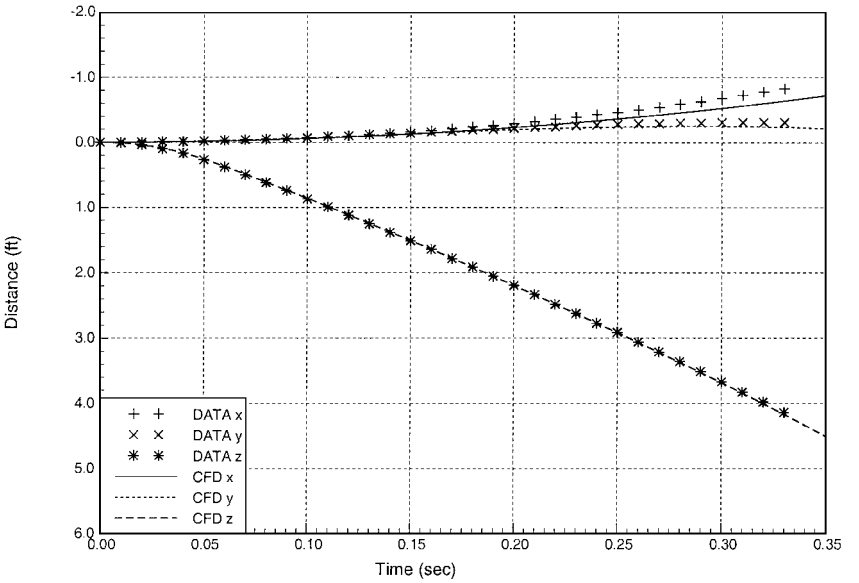

Fig. 2 Linear trajectory (validation case); M_∞ = 0.95, and α = 0 deg.

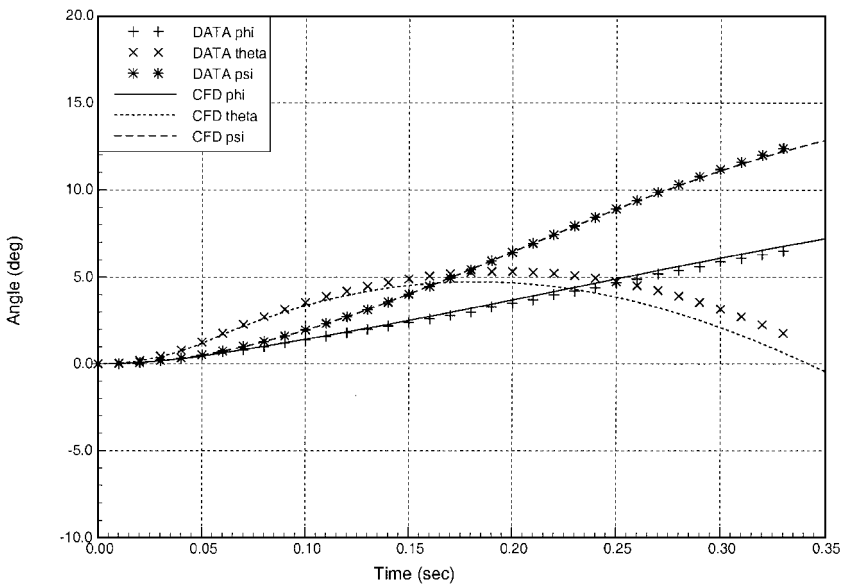

Fig. 3 Angular trajectory (validation case); M_∞ = 0.95, and α = 0 deg.

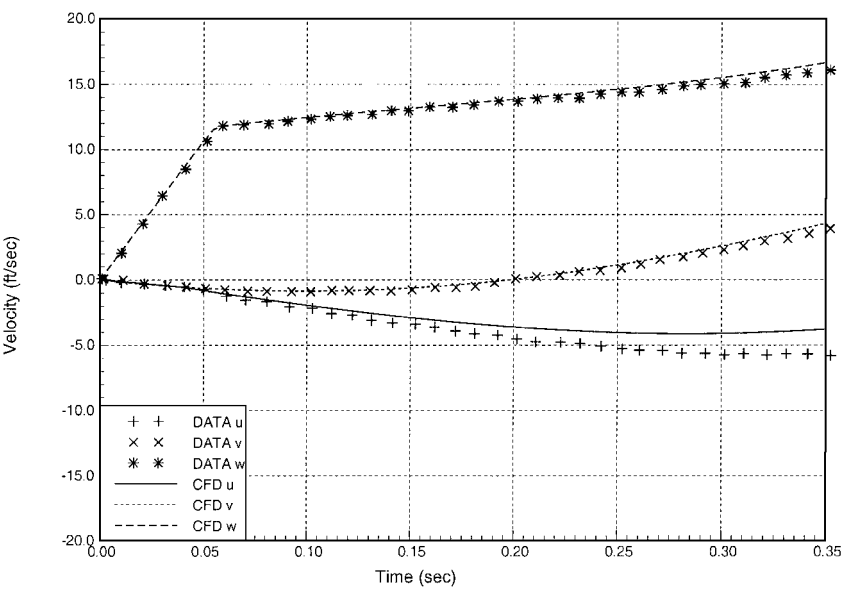

Fig. 4 Linear velocity (validation case); M_∞ = 0.95, and α = 0 deg.

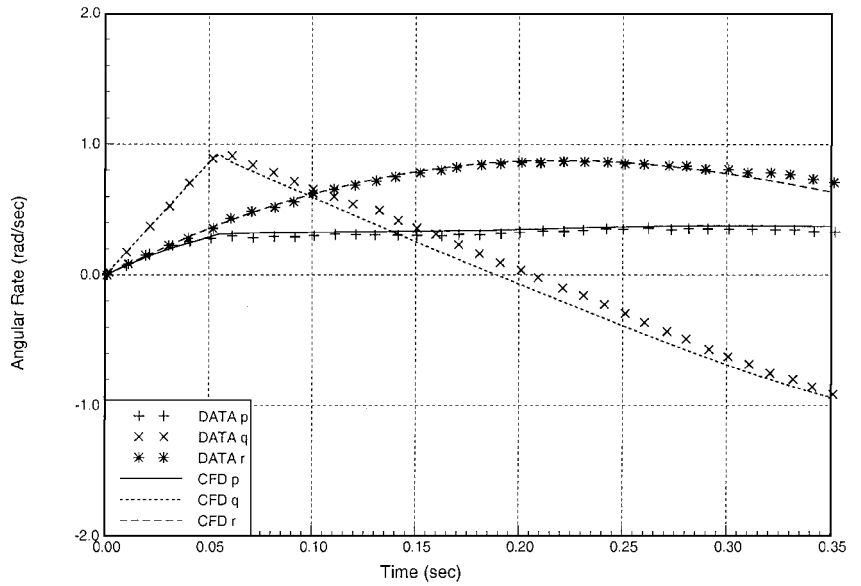

Fig. 5 Angular velocity (validation case); M_∞ = 0.95, and α = 0 deg.

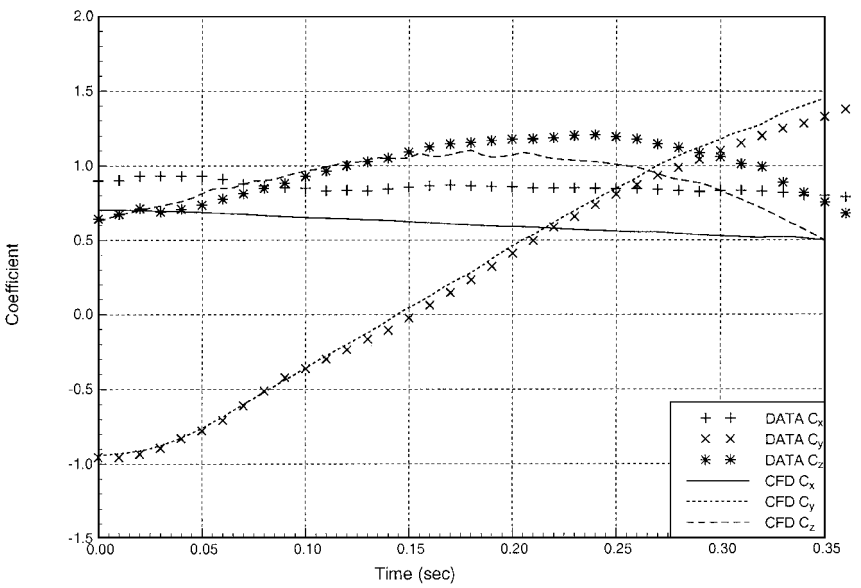

Fig. 6 Force coefficients (validation case); M_∞ = 0.95, and α = 0 deg.

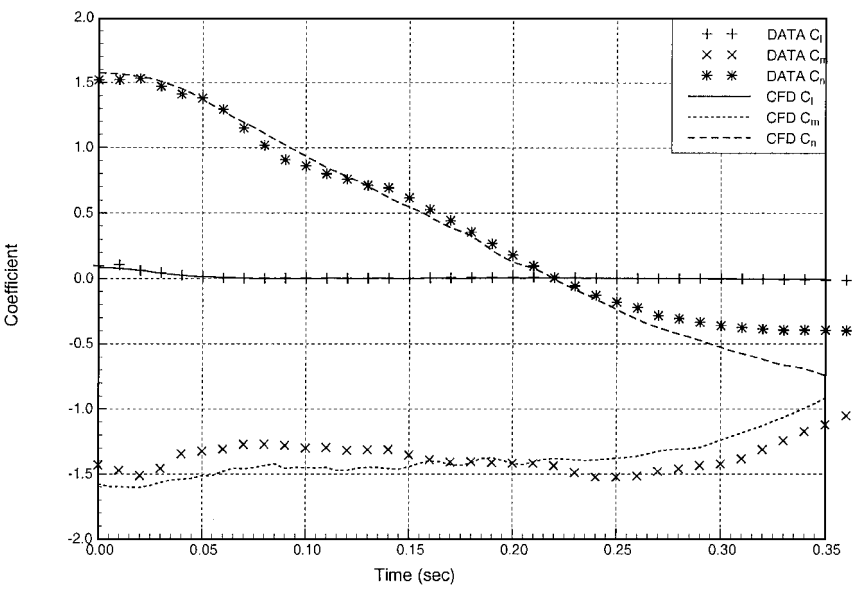

Fig. 7 Moment coefficients (validation case); M_∞ = 0.95, and α = 0 deg.

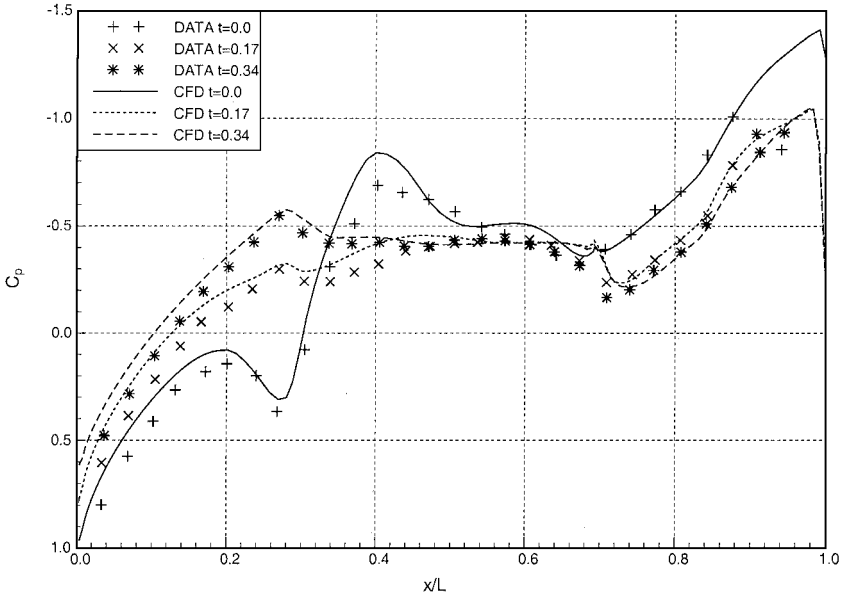

Fig. 8 Surface pressure coefficients; ϕ = 5 deg.

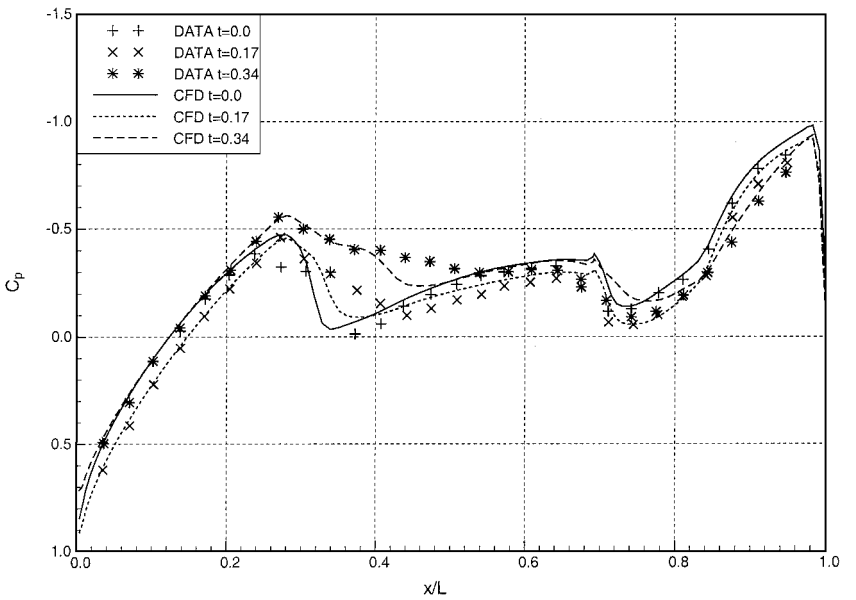

Fig. 9 Surface pressure coefficients; ϕ = 185 deg.

suggested by Harten[19] is applied to avoid expansion shock. Dual time stepping[20] with alternating direction implicit is used to advance the solution in time. This allows one not only to use a large time increment but to maintain temporal accuracy. The dual time stepping eliminates factorization and linearization errors by iterating the solutions along an artificial time. The resulting discretized form of Eq. (1) can be written as

$$L_i D^{-1} L_j D^{-1} L_k \Delta Q = -k_r \Delta \tau R \tag{6}$$

where

$$\boldsymbol{R} = (Q^n - Q)/\Delta t + (1/V)\Big[(\tilde{F}\Delta S)_{i+\frac{1}{2}} - (\tilde{F}\Delta S)_{i-\frac{1}{2}} + (\tilde{F}\Delta S)_{j+\frac{1}{2}} - (\tilde{F}\Delta S)_{j-\frac{1}{2}} + (\tilde{F}\Delta S)_{k+\frac{1}{2}} - (\tilde{F}\Delta S)_{k-\frac{1}{2}}\Big]^n$$

$$\boldsymbol{D} = (\Delta\tau/\Delta t + 1)I \tag{7}$$

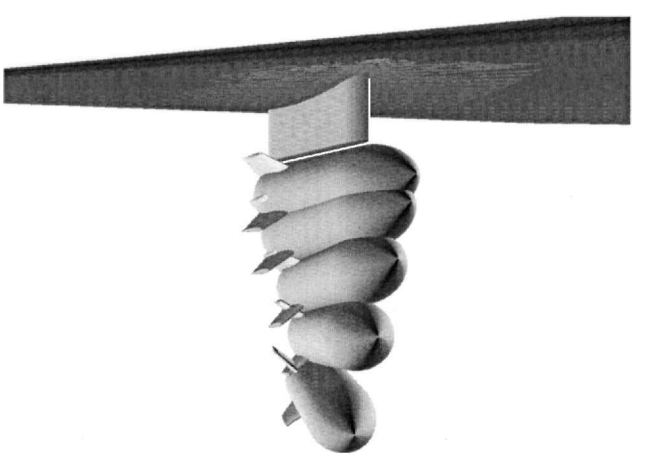

Fig. 10 Separation sequence at every 0.1 s.

Also, the index n denotes the dual time step. The i-directional factor is given by

$$\boldsymbol{L}_i = \left\{ D + k_r \frac{\Delta\tau}{V} \left(\left(\frac{\partial \tilde{F}_{i+\frac{1}{2}}}{\partial Q_{i+1}} + \frac{\partial \tilde{F}_{i+\frac{1}{2}}}{\partial Q_i} \right) \Delta S_{i+\frac{1}{2}} - \left(\frac{\partial \tilde{F}_{i-\frac{1}{2}}}{\partial Q_i} + \frac{\partial \tilde{F}_{i-\frac{1}{2}}}{\partial Q_{i-1}} \right) \Delta S_{i-\frac{1}{2}} \right) \right\} \tag{8}$$

Other factors take similar forms. The number of dual time iterations is found to be around 20, sufficient for accurate simulations through numerical experiments. A detailed description of the solver can be found in Ref. 21.

Table 1 Store mass properties and ejector characteristics of single release

Properties	Eglin Wing/Pylon/Store case
Center of gravity	4.65 ft aft of store nose
Forward ejector location	4.06 ft aft of store nose
Rear ejector location	5.73 ft aft of store nose
Forward ejector force	2,400 lb
Rear ejector force	9,600 lb
Ejector stoke length	0.33 ft
I_{xx}	20 slug-ft^2
I_{yy}	360 slug-ft^2
I_{zz}	360 slug-ft^2
Weight	2,000 lb
Freestream mach number	0.95
Angle of attack	0
Altitude	26,000 ft

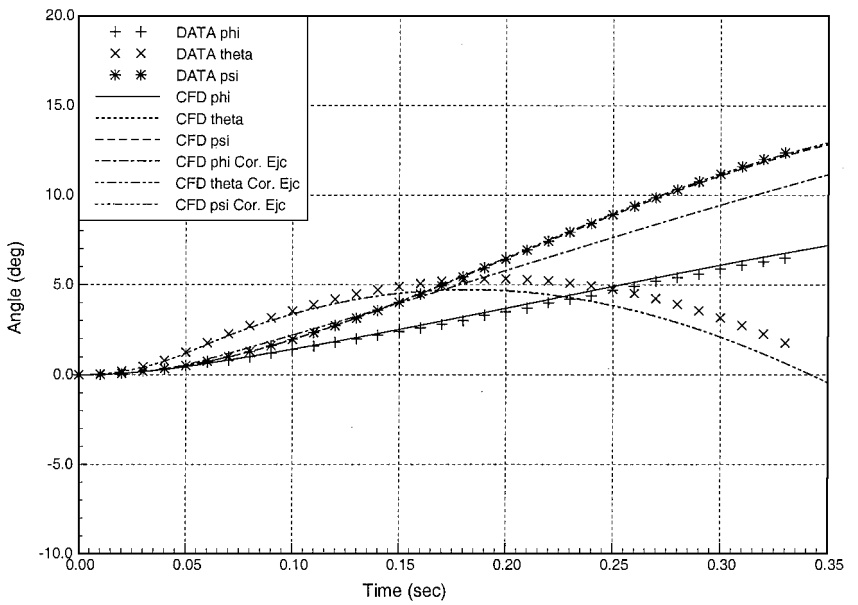

Fig. 11 Angular trajectory, comparision of ejector models.

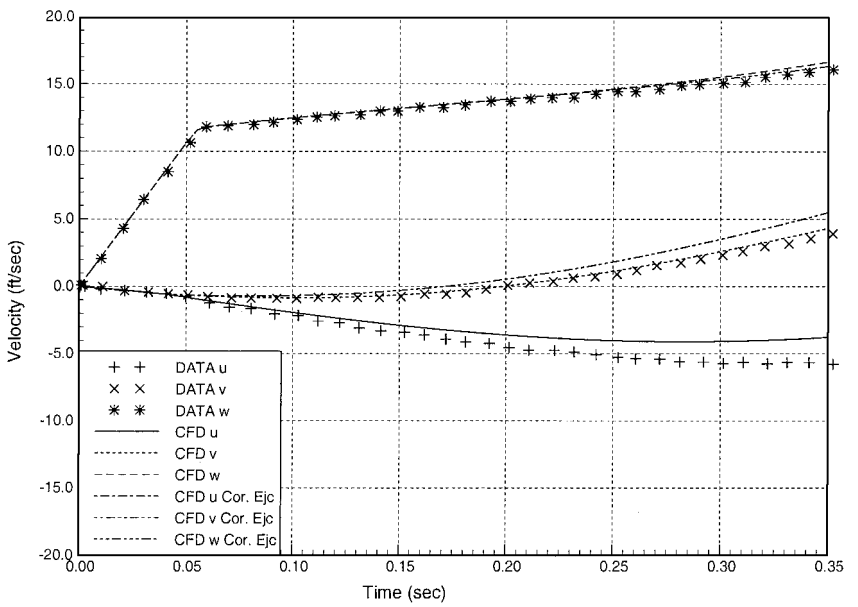

Fig. 12 Linear velocity, comparision of ejector models.

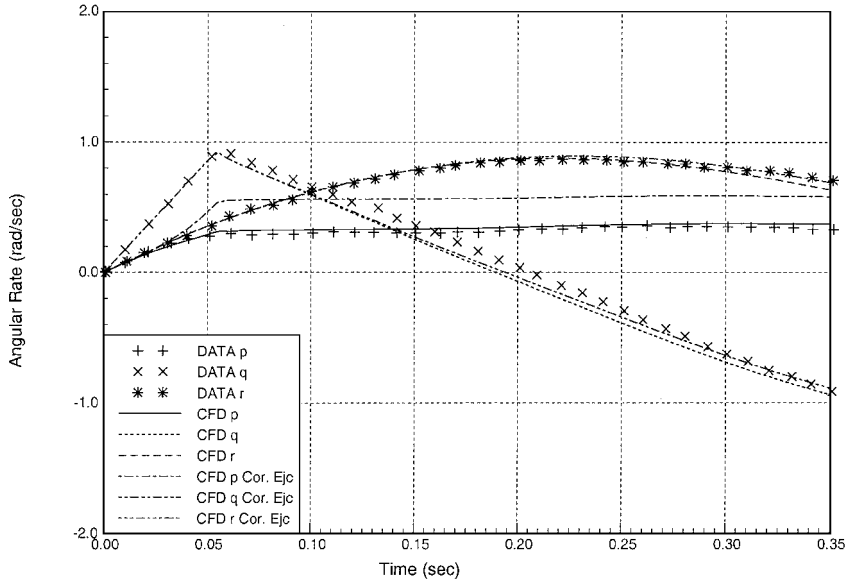

Fig. 13 Angular velocity, comparision of ejector models.

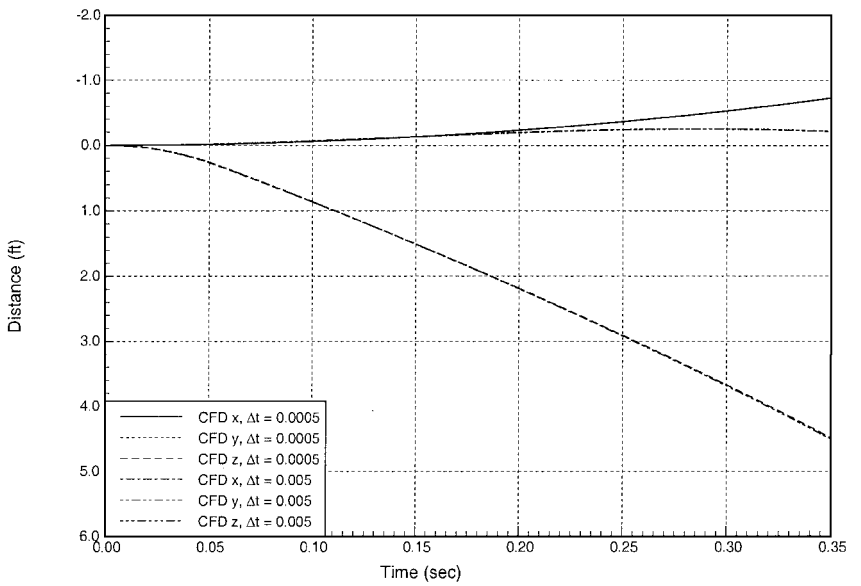

Fig. 14 Linear trajectory, time-step-size study; $\Delta t = 0.0005, 0.005$.

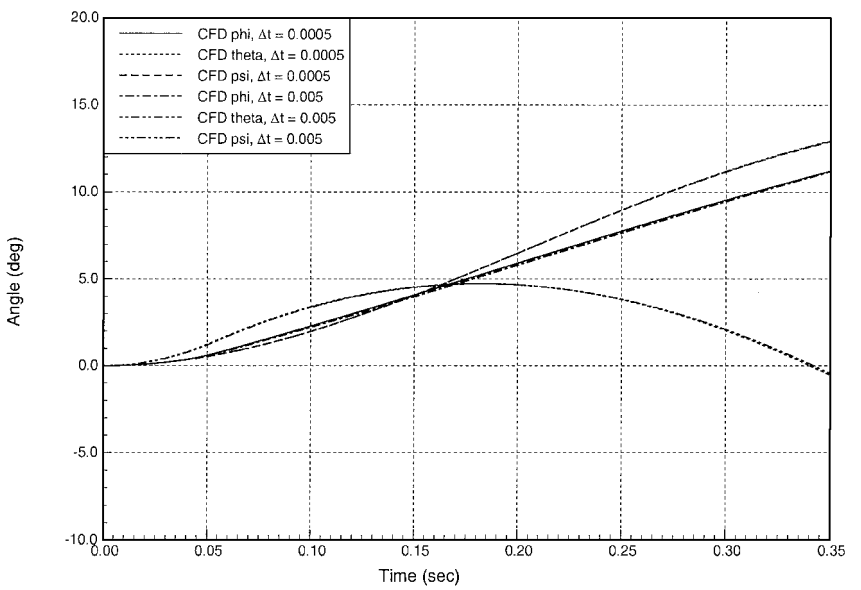

Fig. 15 Angular trajectory, time-step-size study; $\Delta t = 0.0005, 0.005$.

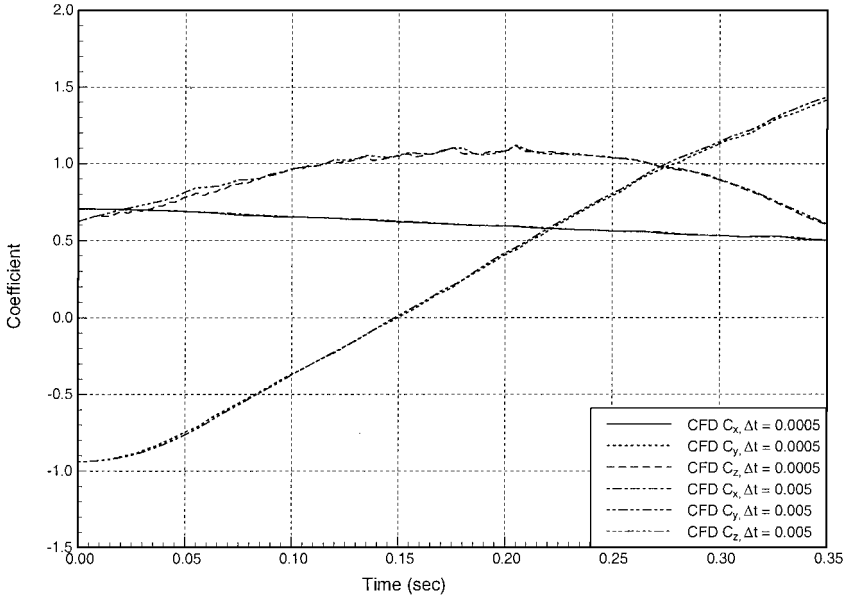

Fig. 16 Force coefficients, time-step-size study; $\Delta t = 0.0005, 0.005$.

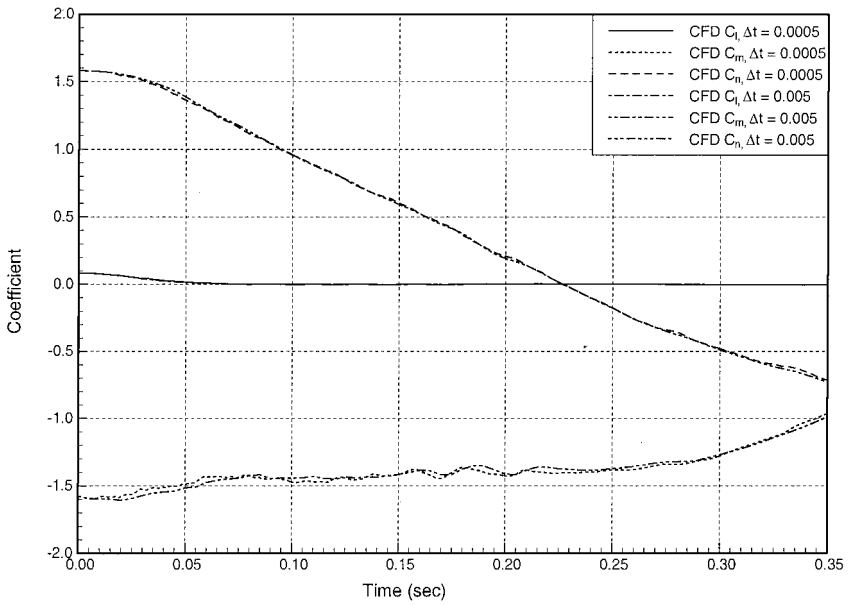

Fig. 17 Moment coefficients, time-step-size study; $\Delta t = 0.0005, 0.005$.

Time-Step Size Limitation

The time-step size in time-accurate computations using Chimera grids is limited not only by the numerical instability and accuracy of the CFD method but also by the time-step size limitation. The time-step size limitation can be understood from Eq. (7). The first term on the right-hand side of the residual is not correctly calculated when a hole cell in the previous time step becomes a normal cell without even being an interpolation cell. The previous time-step solutions of such a cell are not valid. The time-step sizes required to accurately resolve the physical processes being simulated are generally much smaller than the time-step sizes limited by the grid movement.[22] However, with the large time-step sizes being used here, this was not the case. Instead of using a smaller time-step size, the solutions of those cells at the previous time step were interpolated.

Numerical Examples

Eglin Wing/Pylon/Store

The validity of the current Chimera method and the solution procedure is tested against known captive trajectory system (CTS) results.[2] The initial c.g. position and the mass properties of the store and the ejector characteristics are summarized in Table 1. A C-H type grid (129 × 44 × 49) generated over the wing and an O-O type (121 × 27 × 26) over the pylon are overlapped and used as a major grid system together with the H-H grid (105 × 51 × 85) to ease interpolation. A grid system of four block grids (130 × 32 × 25 each) is used to discretize the domain near the missile and is the only grid system allowed to move freely. The total number of grids is approximately 1.2 million points. Figure 1 shows the surface grid of the test case and the outline of the H-H grid system. As can be seen in Fig. 1, a sting is modeled as in the wind-tunnel test.

The initial steady-state solution was obtained in 3000 iterations with local time stepping for fast convergence. The code runs at 11.7 μs per iteration per grid point on CRAY T916. The initial grid assembly took 0.73 μs CPU time per interpolation cell with the manual mode and 7.81 μs per interpolation cell with the automatic mode. The numbers of interpolation cells for the manual mode and the automatic mode are 124,014 and 143,663, respectively. The equations of motion of the store are integrated with the four-stage Runge–Kutta method for a larger time-step size. Chimera domain decomposition

was done at every time step, including the intermediate Runge–Kutta steps, which means four applications of the Chimera domain decomposition in every time step. The time-step size used was 0.005 s. This time-step size is over three times larger than what was used in Ref. 3. The time step required to resolve the dynamics of uncontrolled stores is far larger than that needed to maintain the stability and accuracy of the flow solver. The robustness and accuracy of the solver is believed to be the main reason for using such a large time step. The unsteady grid assembly took 0.17 and 3.94 μs for the manual mode and the automatic mode on average, respectively. The total CPU time for the initial condition and the unsteady computation is about 42 h with the manual mode compared to about 102 h with the automatic mode. Because of the inviscid nature of the flow, the solutions of both modes show little difference. Therefore, only the solutions of the automatic mode are shown here. The advantage of the automatic mode, however, will be evident in viscous computation because the interpolations will occur away from the boundary-layer region.

Figures 2 and 3 show the trajectories and Euler angles of the store compared with the CTS data, showing good agreement with the data, whereas Figs. 4 and 5 show the linear and angular velocities. Figures 6 and 7 show the force and moment coefficients of the store. The computational results match well with the data except for the axial force coefficient, which cannot be predicted correctly with any Euler solvers. In Figs. 8 and 9 the surface pressure coefficients at the initial condition and during the separation are presented, showing excellent agreement with the experimental results. Figure 10 shows the separation sequence at every 0.1 s.

As indicated in Ref. 5, the ejector forces were incorrectly modeled in the CTS simulation. The ejector forces should be modeled in a nonmoving frame so that they can generate rolling and yawing moments on the store. As can be seen in Figs. 11–13, the rolling velocities are almost doubled with the correct ejector model. This is exactly the opposite of the results of Ref. 5. The leading-edge sweep of the wing generates an asymmetric pressure distribution on the store and causes the c.g. to shift inboard immediately after the store is released. Because of the offset of the c.g., the ejector forces cause positive rolling moments on the store resulting in large positive rolling velocities. Similar rolling characteristics can be seen in Fig. 23 in Ref. 4. Figures 14 and 15 present the time histories of the linear and angular position of the store with different time steps. The computations were performed with the correct ejector model. The two results are almost identical indicating that $\Delta t = 0.005$ s is enough for accurate computations. The force and moment coefficient histories (Figs. 16 and 17) confirm the argument.

Ripple Release from TER

As an example of the versatility of the present method, a ripple release from TER was chosen. The simulation was made with the corrected ejector model. The same initial orientation of three stores given in Ref. 6 are used in these computations. Figure 18 depicts

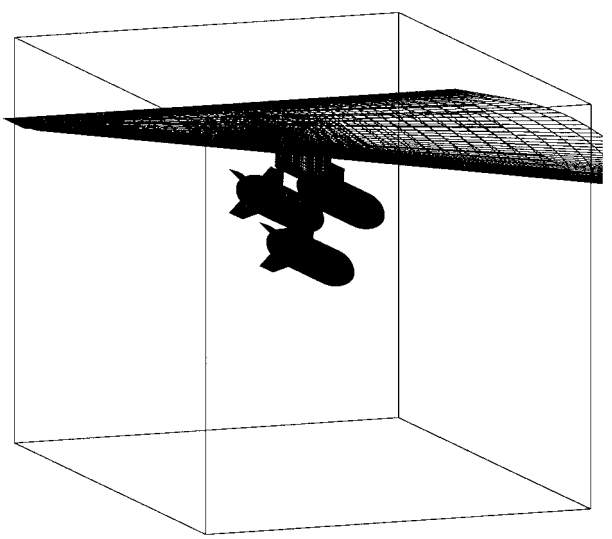

Fig. 18 Surface grid of TER case.

Table 2 Store mass properties and ejector characteristics of the TER case

Properties	TER case
Center of gravity	4.65 ft aft of store nose
Forward ejector location	4.06 ft aft of store nose
Rear ejector location	5.73 ft aft of store nose
Forward ejector force	1,800 lb
Rear ejector force	7,200 lb
Ejector stoke length	0.33 ft
I_{xx}	10 slug-ft^2
I_{yy}	180 slug-ft^2
I_{zz}	180 slug-ft^2
Weight	1,000 lb
Freestream mach number	0.95
Angle of attack	0
Altitude	26,000 ft

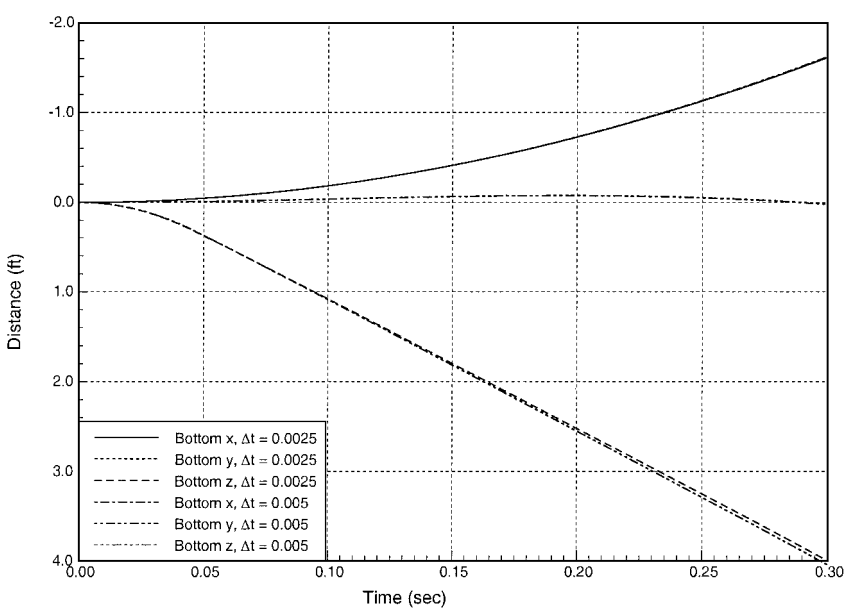

Fig. 19 Linear trajectory, bottom store, TER case.

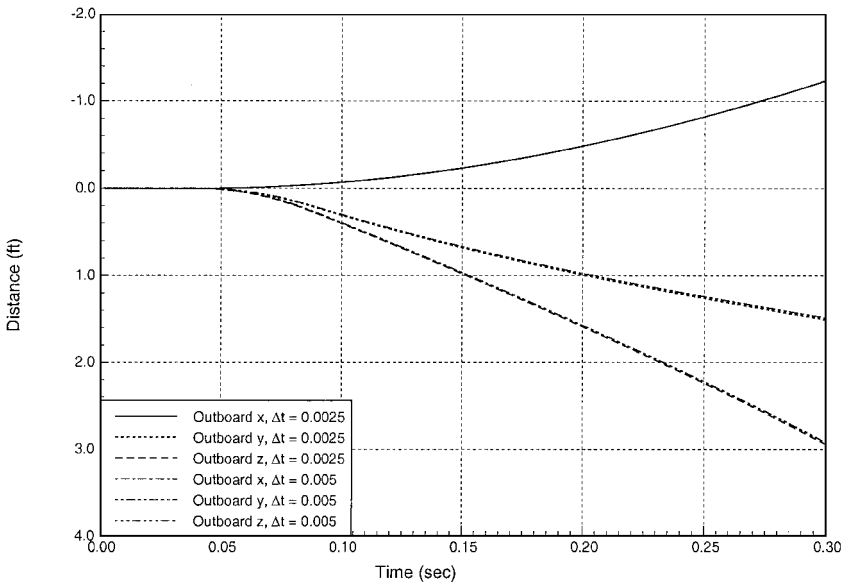

Fig. 20 Linear trajectory, outboard store, TER case.

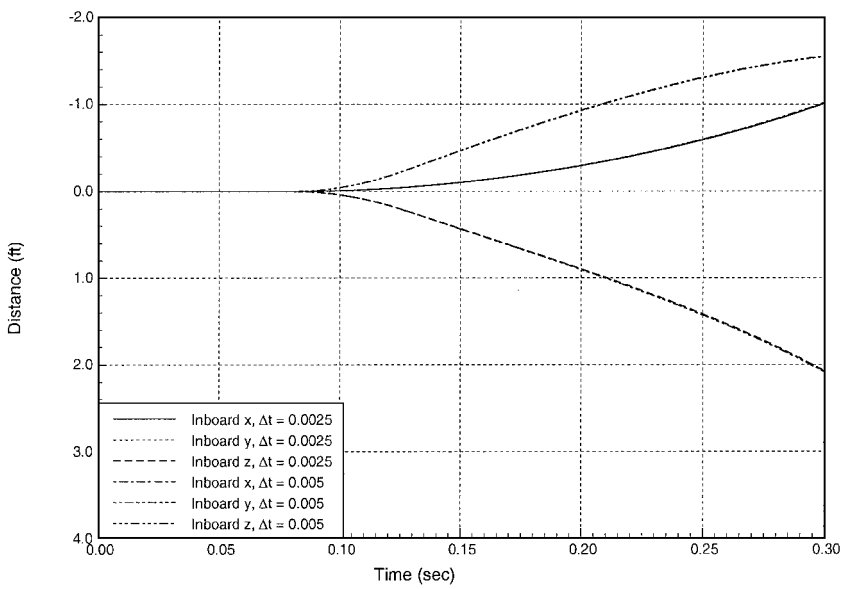

Fig. 21 Linear trajectory, inboard store, TER case.

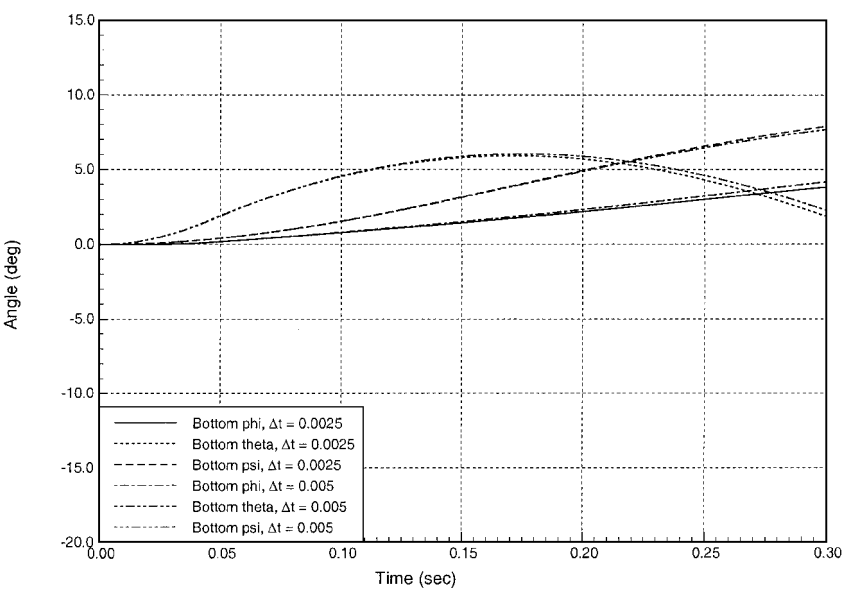

Fig. 22 Angular trajectory, bottom store, TER case.

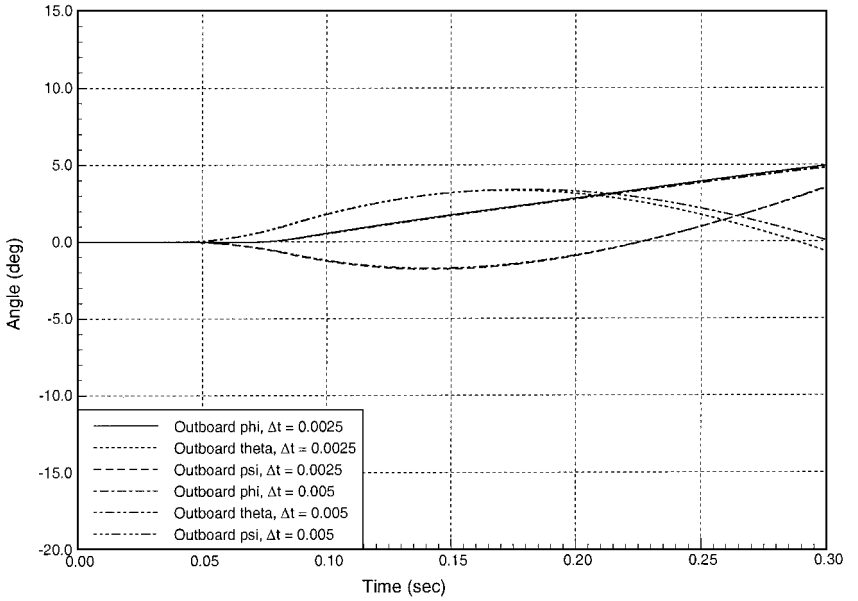

Fig. 23 Angular trajectory, outboard store, TER case.

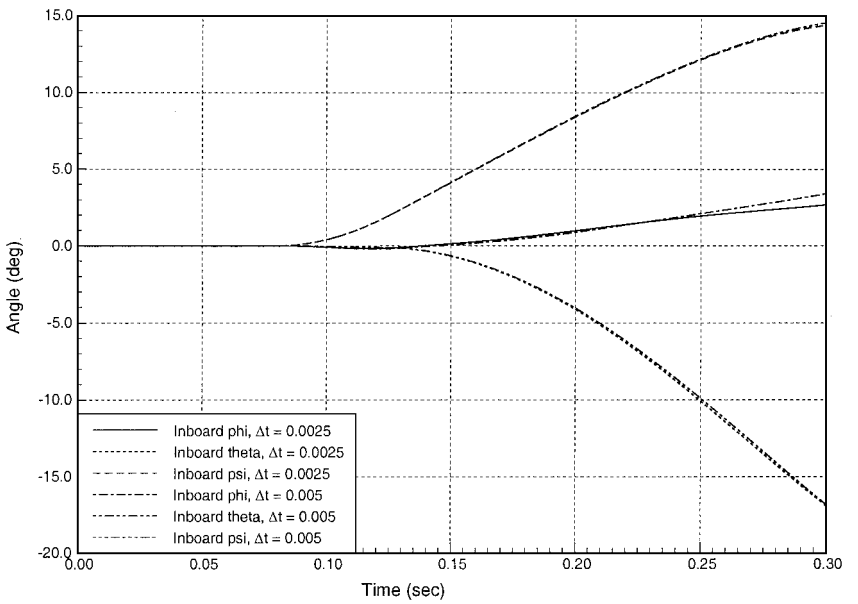

Fig. 24 Angular trajectory, inboard store, TER case.

the surface grid of the current case. The number of grids used to discretize the computational domain is 15 including an H-H grid ($110 \times 81 \times 85$) for improved interpolation. The outline of the H-H grid can also be found in Fig. 18. The total number of grid points is over 2.1 million. The flight conditions are the same as those in the single store release. The store mass properties and the ejector characteristics are the same as used in Ref. 6 and given in Table 2. Because the mass and inertia are reduced by half, the influence of aerodynamic forces and moment are more significant in the TER case than in the earlier release case. Because of CPU intensive computation, the simulation was carried out with the manual mode only. The initial grid assembly took 0.89 μs per interpolation cell, whereas the unsteady grid assembly took 0.22 μs on average. The two time-step sizes of 0.0025 and 0.005 s are chosen to compute the trajectories. Figures 19–21 show the linear trajectories of three stores, whereas Figs. 22–24 show the angular trajectories. These figures indicate that the results with two time-step sizes are similar and that enough resolution can be obtained with $\Delta t = 0.005$ s. Figure 25 shows the separation sequence at every 0.05 s.

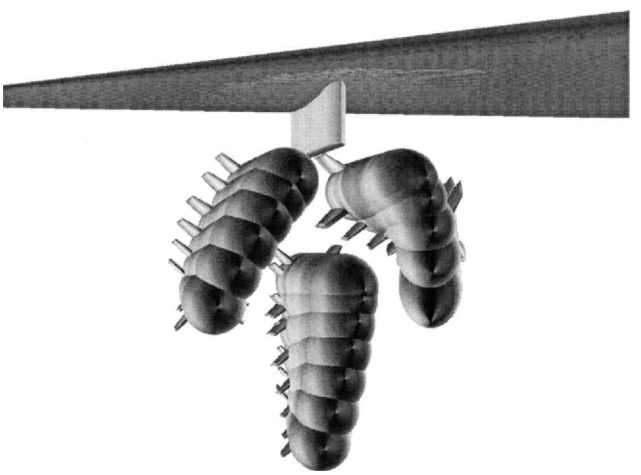

Fig. 25 Separation sequence at every 0.05 s.

Conclusions

A fully automated store trajectory simulation code has been developed. The code has been named Multibody Separation Analysis Program. The code uses a cut-paste algorithm for Chimera hole cutting, a two-step donor cell search, and a highly robust Euler solver. Through numerical examples the demonstration was made that, with the new Chimera method, the single or multiple release of stores can be simulated with a relatively large time-step size. Also, the time-step sizes limited by the grid movement can be larger than those required to resolve the physics of the flow being studied, and this limitation can be relaxed by the interpolation of the previous time-step solution.

References

[1]Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia, ASME-FED-5, American Society of Mechanical Engineers, New York, 1983, pp. 59–69.

[2]Lijewski, L. E., and Suhs, N. E., "Time-Accurate Computational Fluid Dynamics Approach to Transonic Store Separation Trajectory Prediction," *Journal of Aircraft*, Vol. 31, No. 4, 1994, pp. 886–891.

[3]Tramel, R. W., and Nichols, R. H., "A Highly Efficient Numerical Method for Overset-Mesh Moving-Body Problems," AIAA Paper 97-2040, June 1997.

[4]Lijewski, L. E., "Comparison of Transonic Store Separation Trajectory Predictions Using the PEGASUS/DXEAGLE and Beggar Codes," AIAA Paper 97-2022, June 1997.

[5]Hall, L. H., and Parthasarathy, V., "Validation of an Automated Chimera/6-DOF Methodology for Multiple Moving Body Problems," AIAA Paper 98-0753, Jan. 1998.

[6]Thoms, R. D., and Jordan, J. K., "Investigation of Multiple-Body Trajectory Prediction Using Time-Accurate Computational Fluid Dynamics," AIAA Paper 95-1870, June 1995.

[7]Nichols, R. H., and Tramel, R. W., "Applications of a Highly Efficient Numerical Method for Overset-Mesh Moving Body Problems," AIAA Paper 97-2255, June 1997.

[8]Prewitt, N. C., Belk, D. M., and Maple, R. C., "Multiple Body Trajectory Calculations Using the Beggar Code," *Journal of Aircraft*, Vol. 36, No. 5, 1999, pp. 802–808.

[9]Suhs, N. E., and Tramel, R. W., "PEGASUS 4.0 User's Manual," Arnold Engineering Development Center, AEDC-TR-91-8, Arnold AFB, TN, June 1991.

[10]Meakin, R. L., "A New Method for Establishing Intergrid Communication Among Systems of Overset Grids," AIAA Paper 91-1586, June 1991.

[11]Cho, K. W., Kwon, J. H., and Lee, S., "Development of a Fully Systemized Chimera Methodology for Steady/Unsteady Problems," *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 973–980.

[12]Belk, D. M., and Maple, R. C., "Automated Assembly of Structured Grids for Moving Body Problems," AIAA Paper 95-1680, June 1995.

[13]Wey, T. C., "Development of a Mesh Interface Generator for Overlapped Structured Grids," AIAA Paper 94-1924, June 1994.

[14]Dougherty, F. C., "Development of a Chimera Grid Scheme with Applications to Unsteady Problems," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, June 1985.

[15]Meakin, R. L., *Handbook of Grid Generation: Composite Overset Structured Grids*, CRC Press, Boca Raton, FL, 1997, pp. 1–25.

[16]Roe, R. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[17]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, No. 1, 1976, pp. 101–136.

[18]Anderson, W. K., Thomas, J. L., and Rumsey, C. L., "Extension and Application of Flux-Vector Splitting to Calculation on Dynamic Meshes," *AIAA Journal*, Vol. 27, No. 6, 1989, pp. 673, 674.

[19]Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, No. 3, 1983, pp. 357–393.

[20]Merkle, C. L., and Athavale, M., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper 87-1137, *Proceedings of AIAA 8th Computational Fluid Dynamics Conference*, AIAA, New York, June 1987, pp. 397–407.

[21]Lee, S., "A Study on Unsteady Flows with Dynamically Moving Meshes," ASDC-501-960430, ADD, Taejon, Korea, 1996 (in Korean).

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