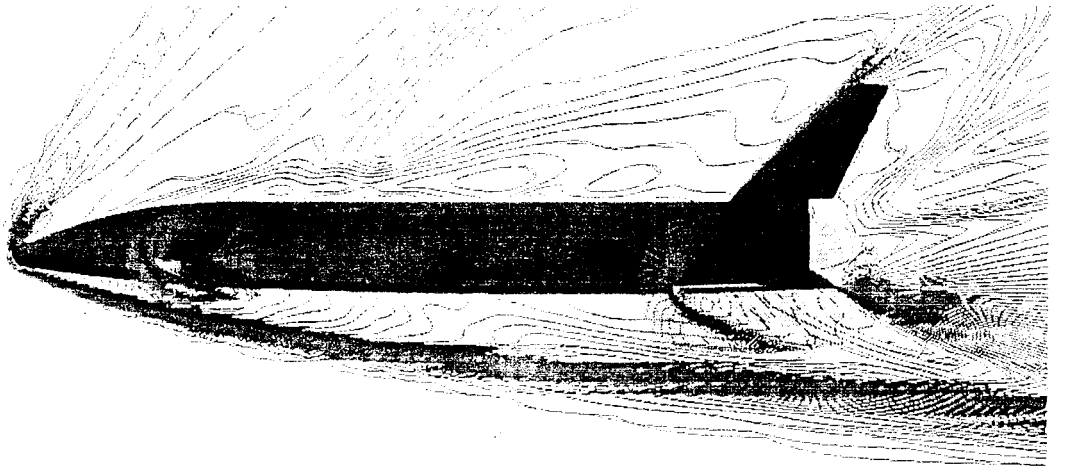
AIAA 2003-3788
Automated CFD Database Generation for a 2nd Generation Glide-Back Booster

Neal M. Chaderjian, Stuart E. Rogers,
Michael J. Aftosmis and Shishir A. Pandya
NASA Ames Research Center
Moffett Field, California

Jasim U. Ahmad and Edward Tejnil
ELORET Institute
Moffett Field, California



21st AIAA Applied Aerodynamics Conference
23-26 June, 2003 / Orlando, Florida

Automated CFD Database Generation for a 2nd Generation Glide-Back Booster

Neal M. Chaderjian*, Stuart E. Rogers†, Michael J. Aftosmis‡, Shishir A. Pandya§
NASA Ames Research Center, Moffett Field, California 94035

Jasim U. Ahmad¶ and Edward Tejnil**
ELORET, Moffett Field, California 94035

Abstract

A new software tool, AeroDB, is used to compute thousands of Euler and Navier-Stokes solutions for a 2nd generation glide-back booster in one week. The solution process exploits a common job-submission grid environment using 13 computers located at 4 different geographical sites. Process automation and web-based access to the database greatly reduces the user workload, removing much of the tedium and tendency for user input errors. The database consists of forces, moments, and solution files obtained by varying the Mach number, angle of attack, and sideslip angle. The forces and moments compare well with experimental data. Stability derivatives are also computed using a monotone cubic spline procedure. Flow visualization and three-dimensional surface plots are used to interpret and characterize the nature of computed flow fields.

Introduction

The past two decades have seen a sustained increase in the use of Computational Fluid Dynamics (CFD) in basic research, aircraft design, and the analysis of post-design issues. One can use a range of fidelities to model the flow field, e.g., engineering databases and panel methods to the Euler and Reynolds-averaged Navier-Stokes (RANS) equations. As the fidelity of a CFD method increases the number of cases that can be readily and affordably computed greatly diminishes. However, computer speeds now exceed 2 GHz, hundreds of processors are currently available and more affordable, and advances in parallel CFD algorithms scale more readily with large numbers of processors. All of these factors make it feasible to compute thousands of high fidelity cases. Applications that were previously thought of as unrealistic are now possible. For example, a sizeable CFD database can be used by itself or in combination with other databases to evaluate an aircraft's stability derivatives, or explore its handling qualities in a flight simulator with a pilot-in-loop.

Even when one is able to generate many solutions, there are other practical issues. How does one execute, monitor, and post-process thousands of cases? Software is needed to reduce the tedious nature of the problem, personnel workload, and tendency for human error that can easily overwhelm a team of engineers. One example of process automation is the ILab software.[1] This software provides a general purpose capability for creating and launching parameter studies. Due to its generality, significant user input is needed to customize it for a particular application. Chaderjian et al.[2] is another example of process automation. In this example, PERL scripts and a PERL Graphical User Interface (GUI) were used to generate a database of solutions and post-process the results for a Harrier in ground effect using a time-dependent RANS method. These scripts greatly reduced the user workload, but used only one CFD code and one geographical site.

A major focus of the Computing, Networking, and Information Systems (CNIS) Project, which is part of the NASA Computing, Information and Communication Technologies (CICT) Program is to facilitate the use of distributed heterogeneous computer systems (grid computing)[3] for generating large numbers of CFD solutions. Grid computing is based on the concept that one could gain significant increases in computational throughput by accessing any number of remote computer nodes through a common job-submission mechanism. The Globus[4] software provides a common job-submission mechanism, secure services for user authentication, remote shell execution, and secure file transfers over an open network.

The objective of the current work is to build and demonstrate a prototype software system (AeroDB) that is based on Globus and will automate the process of running CFD jobs on grid resources. This software is used to meet a CNIS milestone of generating at least 100 RANS solutions and 1000 Euler solutions in one week for a 2nd generation Langley Glide-Back Booster (LGBB) design. A companion paper by Rogers et al.[5]

---

* Research Scientist, Associate Fellow AIAA.
† Aerospace Engineer, Associate Fellow AIAA.
‡ Research Scientist, Senior Member AIAA.
§ Research Scientist, Member AIAA.
¶ Senior Research Scientist.
** Research Scientist.

process is also fast (over 1 million cells per minute). Because the grid generation process is fast and robust, grids are usually created on-demand in the run script on the remote host rather than stored on the mass storage system, thus reducing network transfer time. The Cart3D subsonic grids consist of about 1.4 million cells, and the supersonic grids consist of about 0.8 millions cells. Solution convergence is typically obtained in 200-300 cycles, depending on the flow conditions.

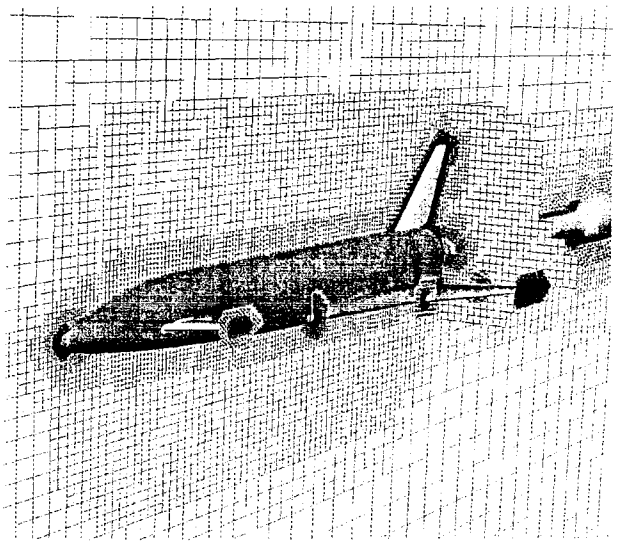


Fig. 2 Cart3D Cartesian grid system.

The flow-solver algorithm is based on an explicit multi-stage procedure with strong multigrid acceleration.[8] Cart3D has been parallelized to efficiently run on shared-memory computers using standard OpenMP directives, and has demonstrated parallel speedups of 496 on 512 processors.

Navier-Stokes Flow Solver

The steady RANS equations are solved with the Overflow[9,10] code, which uses overset structured grids to model complex geometries. Figure 3 shows the overset surface grids for the LGBB geometry. The volume grid system consists of 34 zones. Outer box grids are used to extend the viscous body grids to the far field. These box grids are solved with the Euler equations. The subsonic grid system has far-field boundaries located 10 body lengths from the vehicle in all three coordinate directions, and consists of over 8.5 million grid points. The supersonic grid system has its far-field boundaries located closer to the vehicle (about 1 body length away). This does not affect the solution accuracy due to domain-of-influence effects, and reduces the computational cost because of the smaller outer box grids. The grid system was generated by using an automated script system using the Chimera Grid Tools[11,12] software and the Pegasus[13] overset grid-joining program.

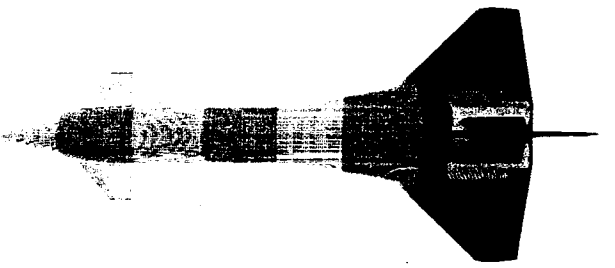


Fig. 3 Planform view of overset surface grid system.

Within the Overflow code, the implicit, approximate-factorization, diagonal algorithm due to Pulliam and Chaussee[14] is used to solve the steady RANS equations with central differencing, together with the Spalart-Allmaras[15] turbulence model. This algorithm was selected because of it computational efficiency. The Overflow solution procedure starts from impulsive conditions using large values of scalar dissipation, and is run for 800 steps using full multi-grid sequencing on three levels. The computation is then run to steady-state convergence using matrix dissipation and three-level multi-grid acceleration. The matrix dissipation provides lower levels of spatial damping while maintaining code stability. The matrix dissipation values suggested by Olsen and Prabhu[16] where used in the present computations. The AeroDB software automates this entire run process.

Two different parallel versions of the Overflow code are used in the solution process. On distributed memory machines, a Message-Passing Interface[17] (MPI) version of Overflow is used. This version of the code explicitly passes inter-zonal boundary condition data between zones as a message between the CPU's using the MPI standard. Load balancing is obtained by distributing the zones among all the CPU's. Since the zones can be significantly different in size, a CPU may be given just one zone, or multiple zones, or just part of a zone.

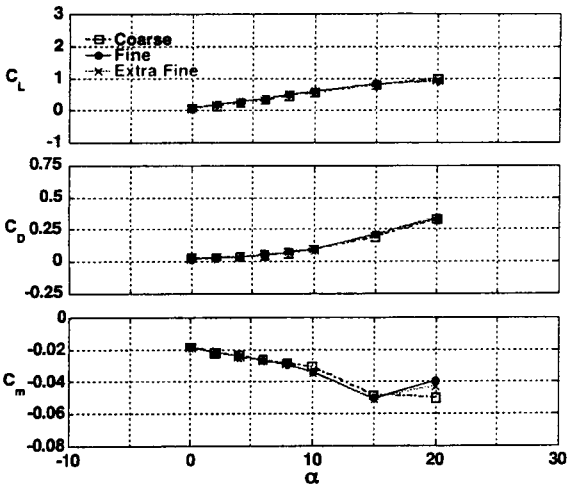
Improved scalability is obtained on single-image computers, e.g. NASA's SGI Origin 2000 and 3000 machines, using multi-level parallelization (MLP). Overflow-MLP[18] utilizes both fine-grain (do loops) and coarse-grain (domain decomposition) approaches to improve parallel performance. Domain decomposition groups together several zones and assigns several processors to each group. The user simply selects the number of groups and processors, and the domain decomposition algorithm in the code decides how many processors to allocate to each group. It seeks to distribute the workload evenly among the available processors by taking into account the number of grid points in each group, and the fidelities assigned to each zone, i.e. Euler, thin-layer, or full Navier-Stokes equations.

the database, representative cases are chosen to validate the computed results. The Overflow Navier-Stokes solutions are computed at full flight Reynolds number (Re) conditions, according to a flight trajectory scenario. This Reynolds number is based on the fuselage length.

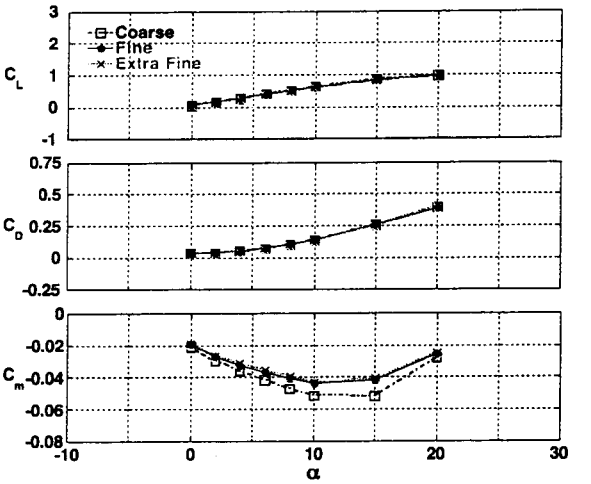
An Overflow grid refinement study is carried out using three different grids for compressible subsonic and transonic flow conditions. A coarse grid consists of about 2 million grid points. A fine grid consists of about 8.5 million grid points and has twice the resolution of the coarse grid in the two body-surface directions. An extra-fine grid consists of about 16 million grid points and has twice the resolution of the fine grid in the two body-surface directions.

Figure 6a compares the Overflow lift, drag, and pitching moment coefficients computed on all three grids with M=0.6. All three grids show virtually identical lift and drag coefficients up to 20 degrees angle of attack. The pitching moment coefficients compare very well with each other using the fine and extra fine grids; however, the coarse grid shows some differences when the angle of attack is above 8 degrees.

In a similar manner, Fig. 6b compares the Overflow lift, drag, and pitching moment coefficients for all three grids under transonic conditions (M=0.9). Once again, the computed lift and drag coefficients compare very well with each other on all three grids. The pitching moment coefficients for the fine and extra-fine grids also compare very well with each other; however, the coarse grid pitching moment coefficients are markedly different and consistently under-predict the values computed on the fine and extra-fine grids. Overall, the fine grid has adequate grid support and is used for computing the present database.



(a) M=0.6, Re=115 million.



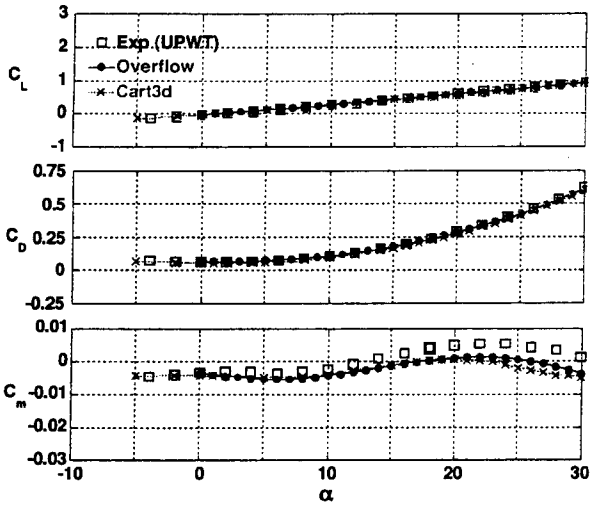
(b) M=0.9, Re=60 million.

Fig. 6 Comparison of Overflow lift, drag, and pitching moment coefficients using three different grid systems.

The inviscid Cart3D code[7, 8] utilizes an advanced Cartesian grid generation algorithm with automated grid refinement as described in the Euler Flow Solver section. This procedure has been applied to a wide variety of vehicles and flow conditions, and its automated grid-refinement algorithm is used to obtain proper grid support for the present database computations.

Both Euler and Navier-Stokes solutions are compared with subsonic data that is available from NASA Marshal Space Flight Center[19] for angles of attack up to 10 degrees, and zero sideslip. Supersonic data is available from NASA Langley Research Center[19] for angles of attack up to 30 degrees, and zero sideslip.

Figure 7 compares computed lift, drag, and pitching moment coefficients with wind tunnel data for several subsonic flow conditions. The Overflow lift, drag and pitching moment coefficients compare very well with the experiment. The Cart3D lift and drag coefficients compare equally well with the experiment, but under predict the pitching moment somewhat. Cart3D computations with and without a sting (not shown here) indicate the sting has very little effect on the computed forces and moments. So the differences in moment coefficients are attributed to viscous effects. Overall, the subsonic computed results compare well with the experimental data.



(c) M=3.0, Re=32 million.

Fig. 8 Comparison of computed lift, drag, and pitching moments with supersonic wind tunnel data.

Visualization

Visualization of two of the viscous cases are now presented. Figure 9 shows the Overflow pressure coefficient (Cp) contours at the symmetry plane and on the surface of the LGBB, for M=1.6, α=20°, and Re=75 million. Shocks near the vehicle nose, along the vertical tail, and the aft-body region are evident. Influence of the canard shock can also be observed at the flow symmetry plane. The Cp contours at the tail end of the fuselage indicate separated flow.

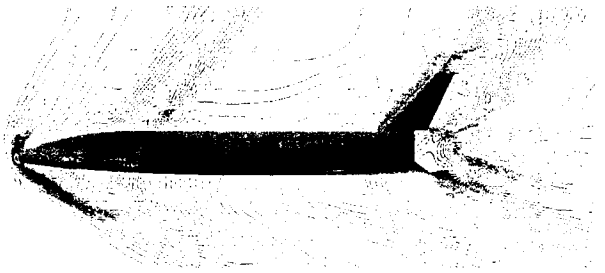


Fig. 9 Pressure coefficient contours (blue ⇒ low Cp, green ⇒ higher Cp). M=1.6, α=20°, Re=75 million.

Figure 10 shows the surface flow topology on the aft portion of the LGBB, and the LEX vortex (shown in red). The LEX vortex provides additional lift at the higher angles of attack. The wing surface flow indicates separated flow. However, due to domain-of-influence (DOI) effects, these separated regions are confined to very close to the body. This helps explain why there are very little viscous effects shown in the previous section for the supersonic Euler and Navier-Stokes lift and drag coefficients, but more significant effects for the pitching moments. This also explains why the flows were mostly found to be steady.

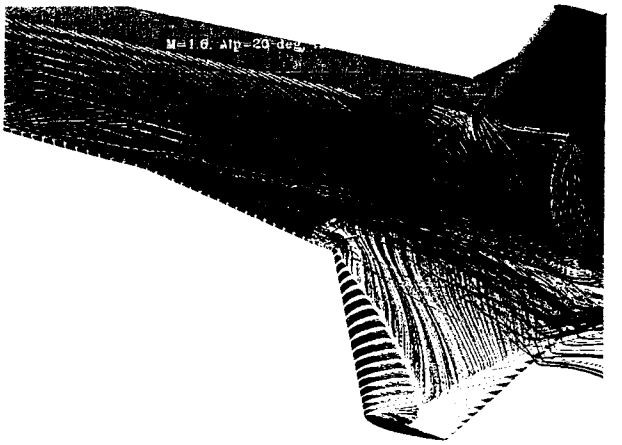


Fig. 10 Surface pressure coefficients, surface flow topology (white), and the off-surface LEX vortex (red). Wing colored green to highlight surface flow. M=1.6, α=20°, Re=75 million.

Figure 11 shows the Overflow pressure coefficient contours at the symmetry plane and on the surface of the LGBB, for M=3.0, α=30°, and Re=32 million. The shock structures are similar to those shown in Fig. 9. However, the nose shock is closer to the underside of the vehicle due to the higher Mach number and angle of attack. Cutting planes through the canard and wing sections also reveal leading and trailing edge shocks. The Cp contours at the tail end of the fuselage also indicate separated flow.

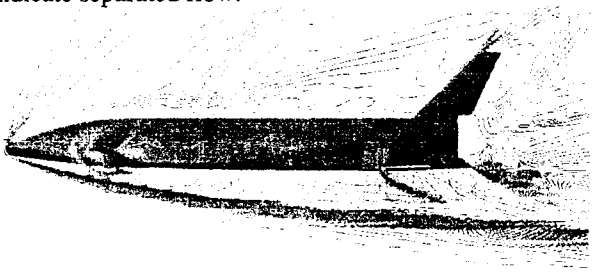


Fig. 11 Pressure coefficient contours (blue ⇒ low Cp, green ⇒ higher Cp). M=3.0, α=30°, Re=32 million.

Figure 12 shows the LGBB viscous surface pressure coefficient, surface flow topology, and off-surface vortical flows highlighted by helicity-density contours (yellow). It is remarkable how complex the surface-flow topology is; and as was the case in Fig. 10, the separated regions are confined to regions very close to the body and wing. This is not the case, however, for the fuselage vortex shown in yellow. The separation and reattachment lines follow the topological rules for physical flows. Primary and secondary separation and reattachment lines are also evident underneath the canard, indicating the matrix dissipation is sufficiently low so as not to wash out these flow structures.

vehicle in a virtual flight simulator before a full-sized flight vehicle is built.

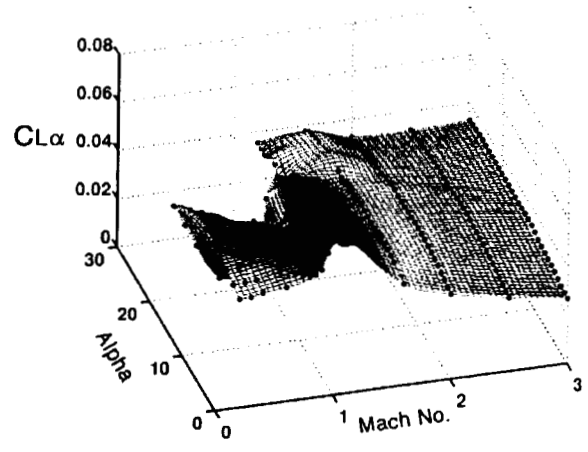


Fig. 16 Overflow lift coefficient stability derivative.

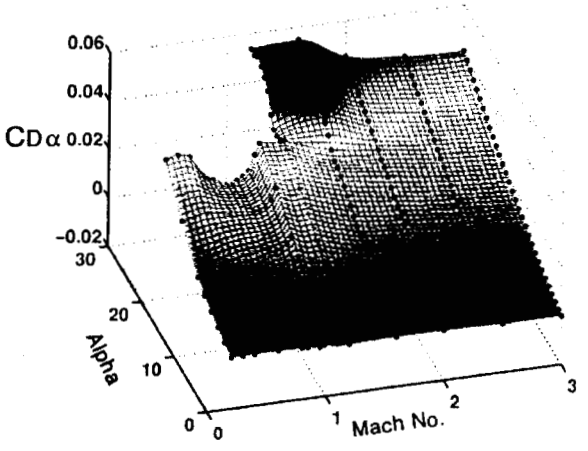


Fig. 17 Overflow drag coefficient stability derivative.

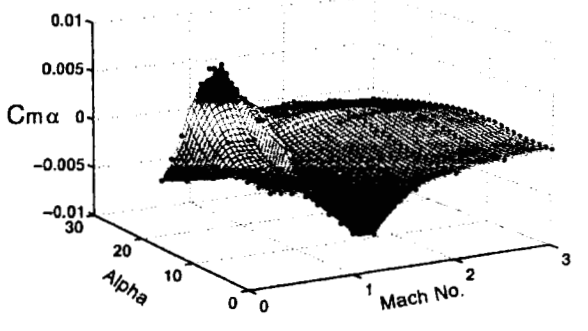


Fig. 18 Overflow pitching moment coefficient stability derivative.

Figures 19-21 show the Cart3D (inviscid) variation of lift, drag, and pitching moment with Mach number and angle of attack. These figures show the same trends as the viscous cases (i.e., Figs. 13-15), but also provide a larger view of the parameter space. The angle of attack range is extended from –5 degrees to 30 degrees for all Mach numbers, and the Mach number range is extended up to M=6. The inviscid and viscous solutions differ primarily in the transonic pitching moment coefficients.

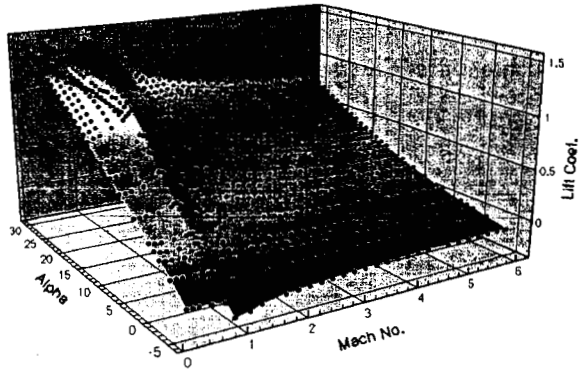


Fig. 19 Inviscid Cart3D lift coefficient.

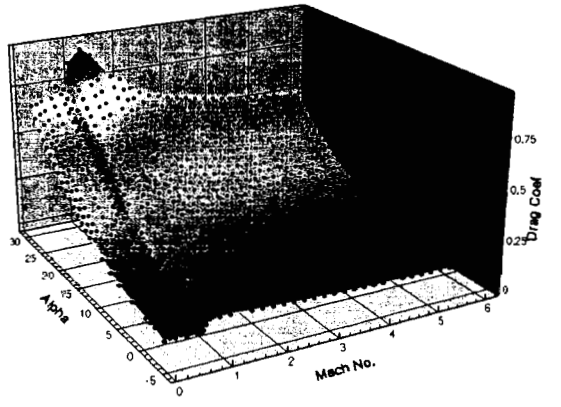


Fig. 20 Inviscid Cart3D drag coefficient.

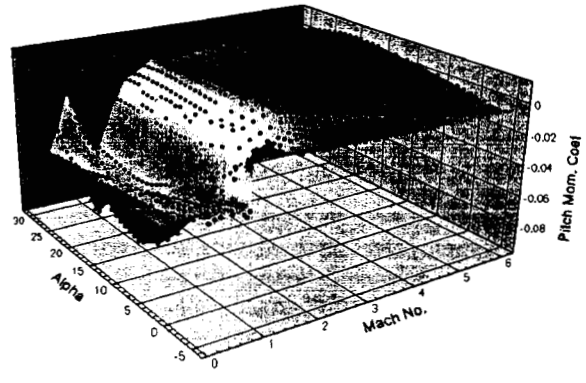


Fig. 21 Inviscid Cart3D pitching moment coefficient.

[12]Chan, W. M., "The Overgrid Interface for Computational Simulations on Overset Grids," AIAA Paper 2002-3188, June 2002.

[13]Suhs, N. E., Rogers, S. E., and Dietz, W. E. "PEGASUS 5: An Automated Pre-processor for Overset-Grid CFD," *AIAA Journal*, Vol. 41, No. 6, June 2003, pp. 1037-1045.

[14]Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, No. 2, 1981, pp. 347-363.

[15]Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, January 1992.

[16]Olsen, M. E., and Prabhu, D. K., "Application of Overflow to Hypersonic Perfect Gas Flowfields," AIAA Paper 2001-2664, June 2001.

[17]Jespersen, D. J., "Parallelism and Overflow," NAS Technical Report NAS-98-013, October 1998. http://www.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-98-013/

[18]Taft, J. R., "Multi-Level Parallelism, "A Simple Highly Scalable Approach to Parallelism for CFD," HPCCP/CAS Workshop Proceedings, (C. Schulbach, ed.), 1998.

[19]Pamadi, B. N., Covell, P. F., Murphy, K. J., Borderlon Jr, W. J., and Frost., A. L., "Aerodynamic Characteristics of Langley Glide Back Booster", to be presented at the AIAA Aerosciences Meeting, Reno, Nevada, January 2004.

[20]Butland, J., and Fritsch, F.N., "A Method for Constructing Local Monotone Piecewise Cubic Interpolants," *SIAM Journal on Scientific and Statistical Computing*, Vol. 5, No. 2, June 1984, pp. 300-304.