Parametric Study of a YAV-8B Harrier in Ground Effect Using Time-Dependent Navier-Stokes Computations

Shishir Pandya[*] and Neal Chaderjian[†]
NASA Ames Research Center
Moffett Field, CA 94035

Jasim Ahmad[‡]
ELORET
Moffett Field, CA 94035

1 Abstract

A process is described which enables the generation of 35 time-dependent viscous solutions for a YAV-8B Harrier in ground effect in one week. Overset grids are used to model the complex geometry of the Harrier aircraft and the interaction of its jets with the ground plane and low-speed ambient flow. The time required to complete this parametric study is drastically reduced through the use of process automation, modern computational platforms, and parallel computing. Moreover, a dual-time-stepping algorithm is described which improves solution robustness. Unsteady flow visualization and a frequency-domain analysis are also used to identify and correlated key flow structures with the time variation of lift.

2 Introduction

Flow simulations using the time-dependent, Reynolds-averaged, Navier-Stokes (RANS) equations continue to be a challenge for powered-lift vehicles such as the YAV-8B Harrier in ground effect. Low-speed flight very close to the ground results in complex time-dependent flow phenomena such as ground vortices, jet-fountains that impact the undercarriage, loss of powered-lift due to Hot Gas Ingestion (HGI) by the inlets, and the "suck-down" effect where high-speed jet flows along the ground plane induce low pressures underneath the vehicle. Many of these phenomena are difficult to accurately model, and occur at very low frequencies. These low frequencies, together with algorithm stiffness due to low-speed ambient flow conditions, contribute to very long compute times.

Various researchers, using simplified geometries, have carried out a number of computational investigations. Examples include single and multiple jets in crossflow[1][2][3] and delta wings with jet nozzles directed towards the ground.[4][5][6] Smith et.al.[7] reported on a RANS solution for a simplified Harrier (wing, fuselage, inlets and nozzles). All of these investigators site two main problems in computing these flows: 1) The need for more accurate solution methods; and, 2) The need for faster solution process. We view the need for a faster solution process as key to improving the solution accuracy, which is the focus of this paper. One can hardly explore the use of different turbulence models and refined grids when a single solution can take many weeks.

A parametric study to generate a database of forces and moments for the Harrier aircraft in near-hover conditions has been discussed in previous papers.[8][9] The Chimera overset grid approach is used to address the geometric complexity of the vehicle. An overset mesh with 67 zones is generated around the Harrier to compute the flow field in the vicinity of the aircraft (see Fig. 1) and the ground plane. The flow field with two high-speed rear jets and two front jets impacting the ground from an aircraft fixed at heights between 10 and 30 feet is found to be complex and time-varying. At each height, computations are carried out at several angles of attack to generate a database of force and moment coefficients. The OVERFLOW code[10] is used to compute the time dependent, viscous flows. Each computation in the parametric study is costly due to the long run times to simulate unsteady flow with low dominant frequencies (~1Hz). In a first attempt at a similar parametric study, it took more than one month to generate one solution without taking grid generation into account. The bottlenecks included the slow decay of numerical transients associated with the solution start-up process, and the lack of scalability to more than 8 processors.

In our previous paper,[8] the emphasis is put on reduction of the time to generate a solution. The computations in the earlier paper used a 52-zone overset grid system and a first order accurate method in time. Small time steps had to be used in order to maintain algorithm stability to capture the low-speed ambient flow. The current paper discusses the use of a dual-time-stepping method to further improve the process. The method allows the use of larger time steps and is more robust. The robustness allows the solution to be started from free stream conditions (impulsive start) in a time-accurate manner. This reduces artificially induced transients allowing for the evaluation of temporal statistics earlier in the computation. The algorithm is now formally second order accurate in
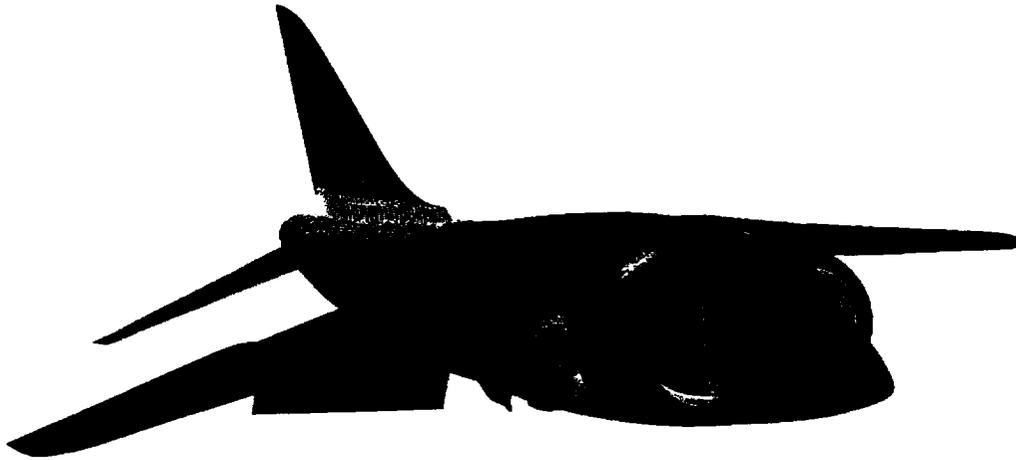

Figure 1: The Harrier vehicle and grid system.

time and maintains solution accuracy with larger time steps. The dual-time-stepping method also eliminates the factorization and linearization errors by iterating between time steps. This dual-time-stepping technique and its advantages are briefly described in section 3.2 .

Reduction of computation time can also be achieved by improving the parallel efficiency. The MLP (Multi-level parallelism)[11] version of the OVERFLOW computational fluid dynamics (CFD) code provides a method of grouping mesh zones into groups so that each group requires approximately the same amount of computational work. However, if too few groups are used on a large number of processors, every group is spread over many processors. Since only loop-level parallelism is responsible for parallel performance within each group, the goal is to keep the number of processors per group to a minimum in order to get the best fine grain parallel efficiency. In the current work, the original 52-zone mesh is split up into 67 zones to ensure both an even distribution of work in domain decomposition and better loop-level parallel efficiency.

Silicon Graphics Origin 3000 computer systems also provide a faster platform contributing to the speed-up. This platform has four CPUs per node (four CPUs which share the same memory) instead of two on the origin 2000 systems used for the previous paper. This removes some of the communication overhead which makes it possible to assign more CPUs to a single group without loss of efficiency. The loop-level parallel efficiency is also improved on the origin 3000 computer due to improvements in the memory access speed, and the cache size and speed. These combined make it possible to scale the problem up to more processors. The present set of cases typically used 112 CPUs per case.

Post-processing the many unsteady solutions to determine mean forces and moments can be an arduous task. The large amounts of data generated during a parametric calculation are automatically stored on a mass storage system. To simplify post-processing a user interface that allows a user to access and process stored data in an automated manner is improved and utilized. An earlier version of the post-processing tool is discussed in reference [9]. Addition of a statistical analysis of the lift coefficient history is discussed briefly.

The post processing is used to gain further understanding of the complex flow field of the Harrier near the ground (see Fig. 2). The low-speed flight combined with the high-speed jet exhaust gives rise to two dominant features in the flow field. The ground vortex is generated due to the interaction of the opposing flows from the jet exhaust impacting the ground and the free stream. A jet fountain flow also moves forward near the symmetry plane to form a swirling fountain vortex. These vortices change in size and behavior with respect to changes in the height and angle of attack.
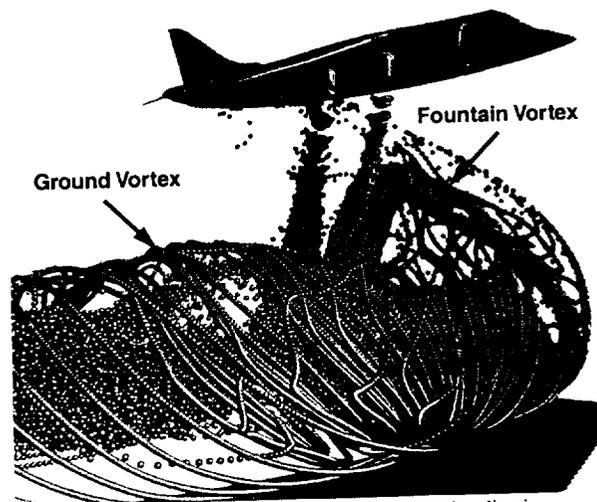

Figure 2: Single frame from an unsteady flow visualization animation.

The mean values of lift are affected by these changes in the dominant features of the flow. The computed values of lift are extended using a monotone cubic spline procedure. Unsteady flow visualization is used to identify and correlate the changes in the gross features of the flow with the dominant frequency in the lift history.

3 Solution Procedure

The procedure to generate many time-dependent solutions has been described in references [8] and [9]. While the Spalart-Allmaras turbulence model[12] is still used and the boundary condition treatment remains unchanged, the grid system, and the numerical algorithm have been changed.

These changes are described in this section along with the process used for a parametric study of unsteady viscous solutions.

3.1 Grid System

The grid system used in reference [8] is modified to improve the load balance for using a large number of CPUs on an SGI Origin platform. In the MLP paradigm, a large number of CPUs for the computation requires that some groups be assigned more than the ideal number of CPUs. For best fine-grain parallel efficiency, each group must be large enough to keep all processors busy. There are several ways to increase the loop-level parallel efficiency of each group. Changing the order of the loops, or the stride are not attempted as these are problem dependent. Instead, it is decided to reduce the number of CPUs assigned to each group by splitting up the large meshes into several smaller grids.

In reference [8] a 52-zone mesh is used for all computations of the Harrier geometry. Reference [9] discusses the improvement gained from splitting some meshes into several pieces. A 67-zone mesh, the result of mesh splitting, is concluded to be more efficient. The goal is to make all meshes similar in size. The mesh splitting is achieved using the OVERGRID code[13] and its scripting feature. The improvement in parallel efficiency due to grid splitting is discussed in section 4 .

The load balancing algorithm in OVERFLOW is also modified to treat meshes according to the work required. The viscous meshes generally require more CPU time per grid point than the inviscid meshes. The viscous meshes are also not equal in terms of CPU time required per grid point. This is because the meshes close to the aircraft surface are computed with the thin-layer approximation while other meshes (i.e., jet grids) require full viscous modelling. A relative weighting system is used where each mesh is given a weight according to the work done by that mesh. If an Euler mesh is the baseline with a weight of 1.0, a thin-layer mesh is given a weight of 1.15. A mesh where all 3 directions are viscous is given a

weight of 1.4. These weights combined with the existing method of computing a load-balance provides a more accurate representation of the work done by the meshes and thus improves parallel performance.

3.2 Dual-time-stepping Algorithm

The dual-time-stepping algorithm in OVERFLOW is based on the dual-time-stepping methods presented in references [14] and [15]. The Navier-Stokes equations can be written in conservative form as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = L(Q) \qquad (1)$$

where $Q = \begin{bmatrix} \rho & \rho u & \rho v & \rho w & e \end{bmatrix}^T$, E, F, and G are the inviscid fluxes and $L(Q)$ represents the viscous terms.

An artificial time term is introduced to the governing equations in order to provide a relaxation (sub-iteration) procedure between physical time steps.

$$\frac{\partial Q}{\partial \tau} + \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = L(Q) \qquad (2)$$

In generalized coordinates this equation is discretized with first order accurate Euler implicit discretization for the artificial time term, second order backwards difference discretization for the physical time terms and central difference discretization for the spatial terms to obtain

$$\frac{\hat{Q}^{k+1} - \hat{Q}^{k}}{\Delta \tau} + \frac{3\hat{Q}^{k+1} - 4\hat{Q}^{n} + \hat{Q}^{n-1}}{2\Delta t} + \delta_{\xi}(\hat{E}^{k} + \hat{A}^{k}\Delta\hat{Q}) \qquad (3)$$

$$+ \delta_{\eta}(\hat{F}^{k} + \hat{B}^{k}\Delta\hat{Q}) + \delta_{\zeta}(\hat{G}^{k} + \hat{C}^{k}\Delta\hat{Q}) = L$$

where $\hat{Q} = J^{-1}Q$; and A, B, and C are the flux Jacobians

$$\left\{ I + \Delta\tau b(\hat{A}^{k}\delta_{\xi} + \hat{B}^{k}\delta_{\eta} + \hat{C}^{k}\delta_{\zeta}) \right\}\Delta\hat{Q} = bR^{k} \qquad (4)$$

where $\frac{1}{b} = 1 + \frac{3\Delta\tau}{2\Delta t}$, and the residual is given by

$$R^{k} = -\Delta\tau\left(\frac{3\hat{Q}^{k} - 4\hat{Q}^{n} + \hat{Q}^{n-1}}{2\Delta t} + \delta_{\xi}\hat{E}^{k} + \delta_{\eta}\hat{F}^{k} + \delta_{\zeta}\hat{G}^{k} - L(Q) \right)$$

The variable n is the time step counter while the variable k is the sub-iteration counter. Finally, $\Delta Q = Q^{k+1} - Q^{k}$. A diagonalized approximate factorization algorithm[16] is used for the solution of this equation. When converged in artificial time, this method is formally second order accurate.

The dual-time-stepping algorithm has three main advantages. The first is its robustness, which allows the computation to

start in a time-accurate manner. It also minimizes artificially generated transients. The transients with the dual-time-stepping method are closer to the physical transients than when the solution is started with a local time stepping procedure. In a previous paper,[8] the solution is briefly started in a steady-state manner using multigrid and local time stepping before a transition to a time-accurate calculation. The dual-time-stepping algorithm eliminates this step allowing for meaningful and efficient temporal evolution.

The second advantage of the dual-time-stepping method is that it allows the use of larger time steps. Stability and accuracy restrictions, especially at low-Mach numbers, limit the time step to unacceptably low values. Time steps can be increased by an order of magnitude. Furthermore, the current unsteady compressible formulation in OVERFLOW is not suitable for very small Mach numbers. The dual-time-stepping method can be modified to implement an unsteady pre-conditioner which addresses the accuracy issues at low Mach numbers. However, preconditioner is not used in the current calculations.

The third and final advantage of the dual-time-stepping method is that the sub-iteration procedure eliminates the factorization error in the diagonalized approximate factorization method. It also eliminates the linearization error.

3.3 Process Automation

A common challenge in a parametric study is the logistics of running many jobs at once and keeping track of them. The goal of this process is to automate the tasks of submitting jobs and archiving the resulting data. An object-oriented Perl module is employed to connect to, store files on, and retrieve files from a mass storage system.[9] This is necessary because the resulting data is far greater in size than is available on a local hard disk array. The Perl module is called by a Perl script that runs a single case. The script first sets up the parameters and determines which case it needs to run (Height, and angle of attack are varied in this study). It then edits a standard OVERFLOW input file to setup the current case conditions. The next step is to copy the appropriate grid and solution (restart) files from the mass storage system to the compute host. The OVERFLOW code is then run for up to 8 hours at a time and the newly generated force/moment history and solution files are stored on the mass storage system.

To run multiple solutions at once and to automate the resubmitting process, another Perl script is used which finds the set of available nodes from the operating system and sets up which cases will be run on which CPUs. The script also sets up the parameters for each case to be run. After invoking as many single runs as it can, given the number of CPUs available, the script proceeds to monitor each job. When a job is completed it puts an end of completion entry in the standard output. When all jobs are completed the script determines if the job should be resubmitted.

The solution process can be further automated by including a procedure in the script which decides when a case is finished. In the present procedure the decision to stop a job is made based on two criteria, convergence of the mean forces and moments, and the presence of a dominant frequency. Currently, this decision is made by the user with the help of a post-processing tool described below.

3.4 Post Processing

Convergence of the mean forces and moments can be determined with a confidence interval test based on a standard student t-distribution.[17] The dominant frequency is more difficult to determine because it may not be constant, the solution may prefer to vary between two frequencies, or the flow may be steady. The dominant frequency is computed with a Fast Fourier Transform (FFT) through the user interface shown in Fig. 3. The automation aspect of this method as well as the precise criteria for deciding to stop the run are currently under investigation.

A statistical analysis has been added to the post-processing tool discussed in [9]. Figure 3 shows the user interface to the post processing tool. The FFT analysis window is shown in the upper right corner of the figure. A lift history along with its mean is shown in the lower right corner of the figure. The initial start-up transient is automatically removed from the temporal analysis. The time-accurate data used for the temporal statistics is highlighted with a thick black line. This tool gives the user access to the solutions without having to know anything about how the solutions are stored on the mass storage machine. The user is asked to choose a parametric study and the parameters to investigate in the main window (see top left of Fig. 3). The user can then retrieve grid and solution files and view them interactively. Default views and rakes are setup to make the process simpler for the user. Three criteria are used to suggest convergence status to the user. At the moment these criteria are under investigation and the user makes the final determination.

The post processing tool also allows the user to plot lift, drag, and pitching moment histories. The FFT analysis can be performed on a force/moment history to obtain mean and confidence interval information along with a power spectrum to reveal the dominant frequencies. The user is also given the option to recompute the vehicle forces and moments based on the OVERFLOW output. This ability is especially helpful when the run is in progress.

4 Results

A parametric study of 35 unsteady flow solutions for the YAV-8B Harrier vehicle near hover conditions is computed in ground effect with a crossflow of 33 knots (M=0.05). The
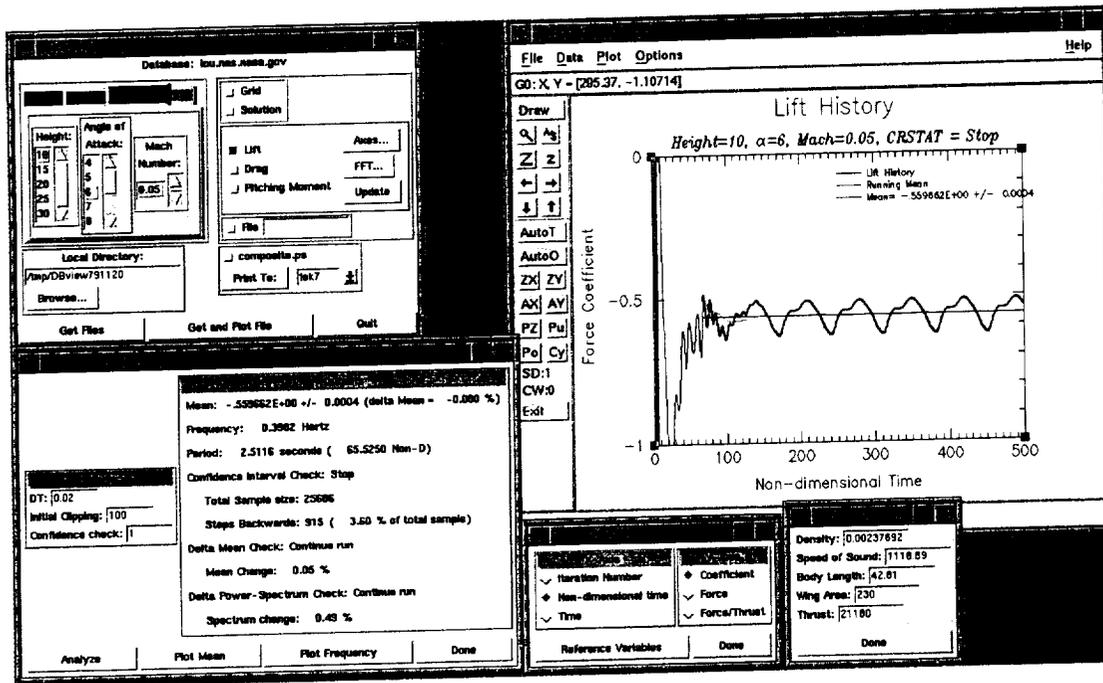
Figure 3: Post-processing interface.

process of completing such a complex study in one week is briefly presented and computational efficiency issues are discussed. A scalability study is presented in section 4.1 to show the improvement in parallel efficiency. The total speed-up achieved is demonstrated by comparing the solution time to that of previous solutions.

The issue of time accuracy of the unsteady solutions is discussed in section 4.2. To address this issue, a time step refinement study is conducted along with a study that uses more subiterations in the dual-time scheme to compute the solution.

The mean lift coefficients obtained from the solution of 35 cases are presented in section 4.3 along with a method for extending the database using monotone cubic splines. All 35 solutions are generated on the SGI Origin 2000/3000 machines using the MLP version of OVERFLOW.

Finally, some of the trends in these solutions are examined with unsteady flow visualization in section 4.4.

4.1 Solution of 35 Cases in One Week on 952 Processors

A parametric study is performed using 840 processors of two SGI Origin 3000 computers and 112 CPUs of an Origin 2000 computer. Two important issues need to be addressed: 1) the logistics of running 35 jobs; and, 2) parallel efficiency necessary to meet the one week deadline.

A change in the existing zonal mesh is responsible for increasing the overall parallel efficiency which resulted in

improved scalability for up to 112 CPUs/case. As discussed in reference [9], a 67-zone mesh delivers better parallel performance due to the improvements in the loop level parallel efficiency and a more balanced CPU load than a 52-zone mesh. Thus the 67-zone mesh is employed.

In order to verify the speed-up, a test case is run on an Origin 2000 at NASA Ames Research Center with the 52-zone mesh and the 67- zone mesh. Figure 4 shows the parallel speed-up of both the 52 zone mesh and the 67 zone mesh with the number of CPUs. The 16 CPU case which has an overall efficiency of 99%[9] is used as the base from which to measure
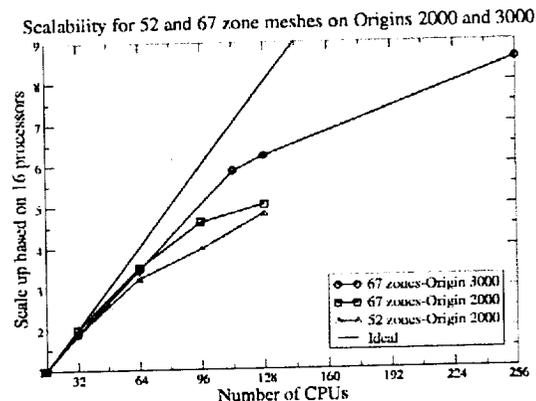

Figure 4: Scalability comparison of 52 and 67 zone solutions.

-5-

parallel speed-up. The 67-zone grid system is more efficient than the 52-zone grid system.

The 67-zone solution scales approximately linearly up to 32 processors. It does not scale linearly beyond that because of low work load for each processor due to the small problem size of 3.8 million grid points. However, the scale up is excellent up to 64 processors where the solution takes only 15% more time than the linear speed-up value. At 96 CPUs, the solution takes 30% more time and is competitive with many implicit parallel CFD codes.

The parallel speed-up improves even further on the origin 3000 system due to the higher number of CPUs per node, the faster access to memory and the larger cache size. The 67-zone case is computed on the Origin 3000 machine to study the speed-up. Once again, the timing for 16 CPUs is used as a baseline. Figure 4 shows the Origin 3000 results for up to 256 CPUs. The figure shows that the parallel speed-up for 32 CPUs is close to ideal. From 32 CPUs to 112 CPUs the speed-up is no longer ideal. However, the loss in efficiency is only 15%. As noted above, the 15% loss in efficiency was reached at 64 CPUs on the Origin 2000. On more than 112 CPUs the performance degrades further and is not acceptable for efficient use of the available resources.

Seven cases are simultaneously run on the Origin 3000 for a total of 784 CPUs. One case is run on the 112 CPUs of Origin 2000. Two other cases are run with 24 and 32 CPUs to fully utilize all available processors. The computations are carried to an average of 15 seconds of physical flight time in order to capture the low dominant frequencies in the solution.

The improvement of solution time over the past 17 months is shown in Fig. 5. All points in the figure are scaled to 3.8 million grid points to indicate actual procedure speed-up. The first data point corresponds to the time-dependent solution of
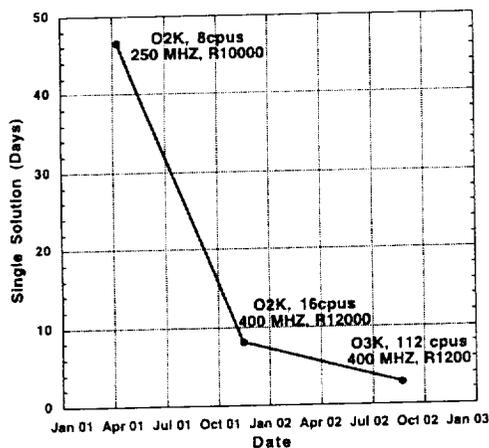
a simplified Harrier geometry. Only the fuselage, wing, inlet and jet exits are modelled along with the ground plane. The half body meshes are made of an average of 2.5 million grid points in 37 zones. The solution was performed on a 128 CPU (250MHz MIPS R10000) SGI Origin 2000. This was the production machine in March/April 2000 when this baseline was computed.

The second data point in Fig. 5 corresponds to the full geometry computation. The full geometry includes the empennage and a deployed wing flap. The mesh consists of 3.1 million grid points at the lowest vehicle height of 10 feet and 3.6 million grid points at the highest vehicle height of 30 feet. These grid points are distributed in 52 overset zones. The computations are performed on the Origin2000 at NASA Ames research center, which has 512 MIPS R12000 CPUs running at 400MHz. Larger grid systems are required at higher heights to capture the larger space between the aircraft and the ground plane. All grids are automatically generated from a base grid system.[9]

The last data point in Fig. 5 corresponds to the same geometry as the one run on Lomax, but the mesh was split into 67 zones to achieve better scalability as noted in section 4.1. This computation was carried out on an Origin 3000 on 112 CPUs per case. Each CPU is a 400MHz MIPS R12000 as is the case for Lomax. The machine based speed-up comes from the faster communication, more/faster cache, and a new node board arrangement where each node holds 4 CPUs that share one memory bank instead of the 2 CPUs on Lomax.

4.2 Time Accuracy

A time step refinement study is presented to show solution behavior with respect to changes in the time step (see Fig. 6). The study involves restarting from an existing solution for 15 sub-iterations with one-half, one-quarter and one-eighth the

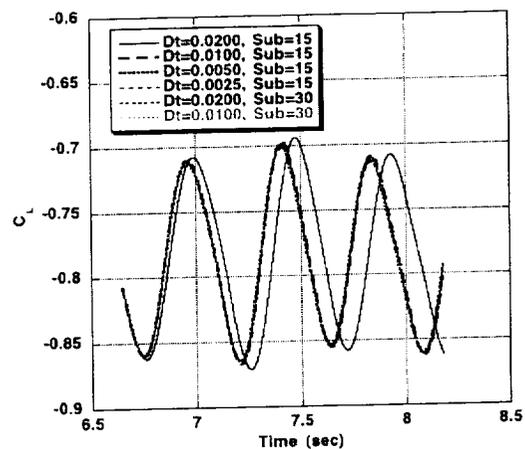Figure 5: Process speed-up for a parametric study.

Figure 6: Comparison of time histories with respect to changes in time step and inner iterations.

time steps to show time step convergence. Figure 6 shows the lift coefficient as a function of time. The solid black line shows the lift history for the time step (0.02) and subiterations (15) used for the current parametric study. The mean value of lift changes by approximately 1% when the time step is reduced by a factor of eight. The change in the dominant frequency due to time step refinement is approximately 6%. All other time steps agree with the value of lift and the dominant frequency computed for the refined time step.

The number of inner iterations in the dual-time-stepping procedure is also doubled for two of the time steps to show that the solution is fully converged between time steps. Figure 6 shows that the solutions with more sub-iterations are in agreement with the solutions with refined time steps. Both time step refinement and an increase in sub-iterations show that the solution obtained is within engineering accuracy for the computation of the mean value of lift.

Similar results are obtained for drag, and pitching moment. These results follow similar trends and support the above conclusion.

4.3 The Lift

The initial 35 solutions are represented in Fig. 7 by the surface plot of the mean coefficient of lift plotted against vehicle height and angle of attack. The size of the database is increased from 35 CFD solutions to over 2500 cases by using a local monotone cubic spline procedure.[18] This is accomplished by applying a one-dimensional interpolation operator successively in each parametric direction, where the angle-of-attack resolution is increased from one degree to 0.1 degrees, and the height resolution is increased from five feet to 0.5 feet. The amount of refinement will depend on the database requirements, and its accuracy will depend on an adequate parametric resolution of the base CFD solutions.
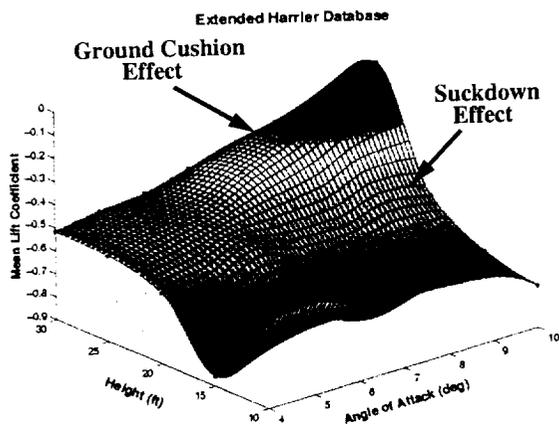
The monotone property is also very important if one wishes to extract meaningful stability derivative information from the interpolated data.

At the higher heights the lift coefficient increases with increasing angle of attack. This expected behavior occurs because the jet flows and ground/fountain vortices do not impact the vehicle and thus have little influence on the aerodynamic lift. Thus, the lift is based mostly on the performance of the wing.

When the vehicle is at a low height, the high-speed jet flow impacting the ground results in a low pressure region under the vehicle. This suck-down effect is responsible for the lower lift. At increasing heights the suck-down effect reduces and thus lift increases. This behavior lasts as long as the vehicle is in ground effect. The drop in lift above 25 ft. is associated with the loss of the ground cushion effect.

4.4 Unsteady Flow Visualization

Unsteady flow visualization is performed on several cases in in an attempt to explain the variations in flow properties. A software tool, Graphics Encapsulation Library (GEL), was developed at NASA Ames research center and relies on "out-of-core" visualization technology.[19][20] This method is an excellent tool to visualize the large amount of time dependent data from one simulation on a workstation. With limited memory this is usually a difficult task. However, "out-of-core" algorithms can predict which part of the data will be needed. The unused data can reside on disk instead of loading the entire data file in memory.

Unsteady flow visualization is used to correlate the dominant frequency to a flow feature. For example, at a height of 10ft and angle of attack of 6° (lift history shown in Fig. 8), the dominant frequency in the lift history correlates very well with the movement of the fountain vortex.

Figure 7: Lift as a function of vehicle height and angle of attack. Monotone cubic spline interpolation of 35 CFD solutions.
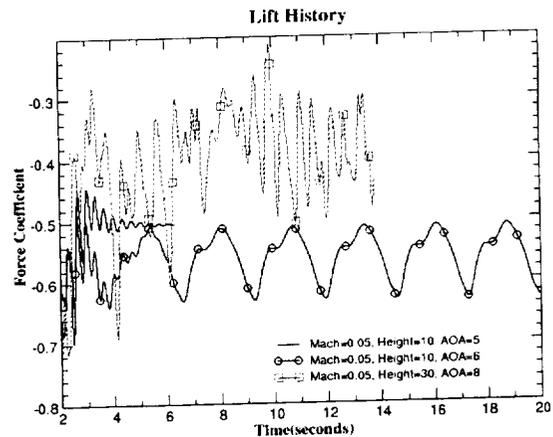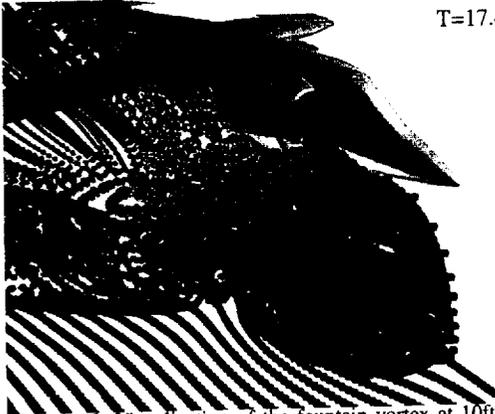

Figure 8: Typical lift histories.

The dominant frequency in the flow field, 0.398Hz, corresponds to a period of approximately 2.5 seconds. Figures 9, 10, and 11 show the extent of the fountain vortex at 16.2, 17.45 and 18.7 seconds respectively. Figure 9 shows that the fountain vortex is tall, covering all of the inlet face. However,

Figure 9: Visualization of the fountain vortex at 10ft height and 6° angle of attack at time of 16.2 seconds.

Figure 10: Visualization of the fountain vortex at 10ft height and 6° angle of attack at time of 17.45 seconds.

Figure 11: Visualization of the fountain vortex at 10ft height and 6° angle of attack at time of 18.7 seconds.

when 1.25 seconds has passed, the same fountain vortex is shorter and wider as indicated in Fig. 10.

Figure 11 shows the same tall vortex another 1.25 seconds later when it has grown to its full size again. This process repeats itself, and the 2.5 second period corresponds to the dominant frequency seen in the variation of the lift force. Unsteady flow visualization also reveals that the fountain vortex traverses a circular footprint along the ground. The ground vortex and the jets are mostly steady. Clearly, hot gas ingestion is a problem at this height.

At higher heights (20ft and above), the fountain vortex is much smaller and no longer exhibits this unsteady behavior. The lift history for a 30 ft. case at an angle of attack of 8° is also shown in Fig. 8. The dominant frequency for the lift history at this condition is 1.75Hz, and much higher than the 10 ft. case. This corresponds to a period of approximately one half of a second. The unsteady flow feature that corresponds to this frequency is shown in Figs. 12 and 13 on the underside of the wing in the region where the flap has been deployed. This separation and attachment cycle underneath the wing is responsible for the major changes in lift. The flow turns
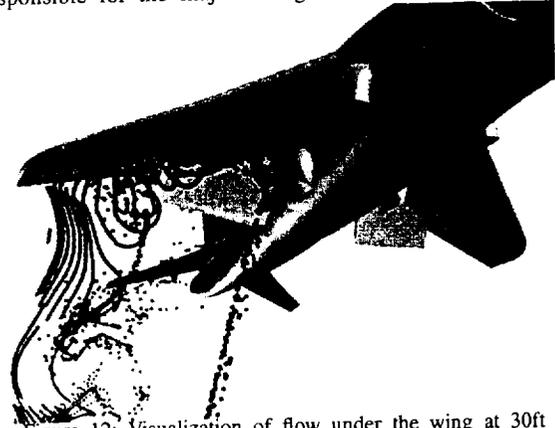

Figure 12: Visualization of flow under the wing at 30ft height and 8° angle of attack at time of 10.47 seconds.
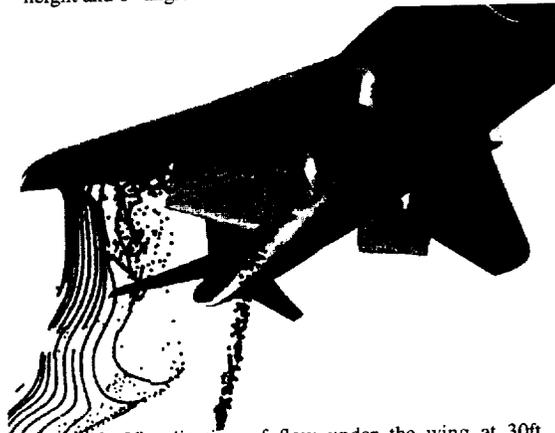

Figure 13: Visualization of flow under the wing at 30ft height and 8° angle of attack at time of 10.76 seconds.

downward and separates because of the adverse pressure gradient associated with the deployed flap. Some of the separated flow is also entrained into the rear jet flow. The flow from the jets is not shown in Figs. 12 and 13 to highlight the separated flow.

All cases are treated as unsteady because it is not possible to predict apriori which cases may exhibit steady behavior. In fact a few cases close to the ground at low angles of attack are found to be steady. For example, at a height of 10 ft. and angle of attack of 5°, the initial transients in lift rapidly damp to a steady value as shown in Fig. 8. The fountain vortex (see Fig. 14) doesn't change its size or location with time, and the jet exhaust flows and ground vortex are also found to be steady. On the other hand, it was previously shown that at an angle of attack of 6° the fountain vortex was unsteady, changing its size and location with time (see Figs.9-11).
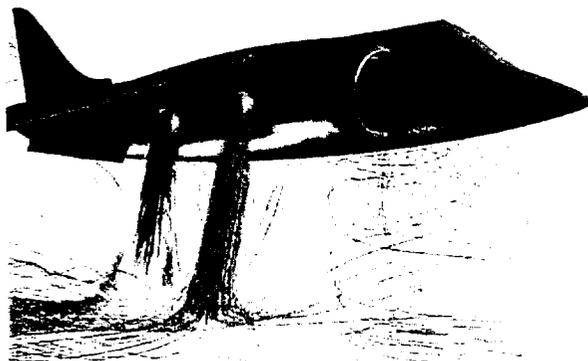

Figure 14: Visualization of flow at 10ft height and 5° angle of attack at time of 6.3 seconds.

5 Summary and Conclusions

A parametric study of 35 time dependent viscous solutions is completed in one week. The results exhibit many of the aerodynamic characteristics common to powered-lift vehicles in ground effect, such as Hot Gas Ingestion, suck-down effect, and ground-cushion effect. Unsteady flow features, such as moving fountain vortices and separated flows, correlate with the dominant frequency of lift the history. The improvement in process speed for running a large number of unsteady viscous solutions is attributed to modern computational platforms and improved algorithms. A speed-up of over 17x is achieved in a span of 17 months. The force/moment database is extended to 2500 solutions by a monotone cubic spline procedure. A new tool (GEL) is used for unsteady flow visualization to create movies of the time-dependent flows so the dominant frequencies can be correlated to the dominant features of the flow. The time consuming post-processing step is further improved with a better user interface and the addition

of statistical analysis to compute mean forces and moments and to examine the dominant frequencies in the flow.

This research has reduced the computation time of high fidelity time dependent viscous solutions using improved computing technologies and state-of-the-art computing platforms. The issues of accuracy and turbulence modelling can now be more readily addressed. The use of statistical analysis to determine when a run is complete remains a topic of further research.

6 Acknowledgement

References

[1] Van Dalsem, W. R., Panaras, A. G., and Steger, J. L., "Numerical Investigation of a Jet in Ground Effect with a Crossflow," SAE Paper 872344, December 1987.

[2] Barata, J., Durao, D., and McGuirk, J., "Numerical Study of Single Impinging Jets Through a Crossflow," AIAA Paper 89-0449, January 1989.

[3] Barata, J., "Numerical and Experimental Study of Fountain Flows Produced by Multijet Impingement on a Ground Plane," AIAA Paper 91-1806, June 1991.

[4] Van Dalsem, W. R., Chawla, K., Smith, M. H., and Abeloff, P. A., "Numerical Simulation of Powered-Lift Flows," International Powered Lift Conference Proceedings, The Royal Aeronautical Society, August 1990, pp. III.16.1-14.

[5] Chawla, K., and Van Dalsem, W. R., "Numerical Simulation of a Powered-Lift Landing," AGARD-CP-534, April 1993, pp.32-1-32-10.

[6] Roth, K. R., "Comparison of Computation with Experiment for a Geometrically Simplified Powered-Lift Model," AIAA Journal of Aircraft, Vol. 34, No. 2, March-April 1997, pp. 160-167.

[7] Smith, M. H., Chawla, K., and Van Dalsem, W. R., "Numerical Simulation of a Complete STOVL Aircraft in Ground Effect," AIAA 91-3293-CP, September 1991.

[8] Chaderjian, N. M., Pandya, S., Ahmad, J., and Murman, S. M., "Parametric Time-Dependent Navier-Stokes Computations for a YAV-8B Harrier in Ground Effect," AIAA Paper No. 2002-0950, AIAA 40th Aerospace Sciences Meeting & Exhibit, Reno, NV, January 14-17, 2002.

[9] Murman, S. M., Chaderjian, N. M., and Pandya, S., "Automation of a Navier-Stokes S&C Database Genera-

tion for the Harrier in Ground Effect," AIAA Paper No. 2002-0259, AIAA 40th Aerospace Sciences Meeting & Exhibit, Reno, NV, January 14-17, 2002.

[10] Buning, P. G., Jespersen, D. C., Pulliam, T. H., Chan, W. M., Slotnick, J. P., Krist, S. E., and Renze, K. J. "OVE-FLOW User's manual," NASA.

[11] Taft, J. R., "Multi-Level Parallelism, a simple highly scalable approach to parallelism for CFD," HPCCP/CAS workshop Proceedings, (C. Schulbach, editor), 1998.

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

[13] Chan, W. M., "The OVERGRID interface for Computational Simulations on Overset Grids," AIAA Paper No. 2002-3188.

[14] Pulliam, Thomas H., "Time accuracy and the use of implicit methods," AIAA Paper No. 93-3360.

[15] S. Venkateswaran and Merkle, C. L., "Dual time stepping and preconditioning for unsteady computations," AIAA Paper No. 95-0078.

[16] Pulliam, T. H., and Chausee, D. S., "A diagonal form of an implicit approximate-factorization algorithm," Journal of Computational Physics, Vol. 39, No. 2, 1981, pp. 347-363.

[17] Bendat, J. S., and Piersol, A. G., "Random Data: Analysis and Measurement Procedures", Wiley-Interscience, New York, 1971.

[18] 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.

[19] Sandstrom, T. A., Chaderjian, N. M. "The Power of Unsteady Flow Visualization," Gridpoints, Vol. 2, No. 2, Summer, 2001, pp. 16-17,20.

[20] Cox, M. B., Ellsworth, D., "Application-controlled demand paging for out-of-core visualization," IEEE Visualization '97 (Roni Yagel and Hans Hagen editors) pp.235-244.