AIAA

AIAA 2005–5093

SIMULATION OF UNSTEADY FLOWS USING AN UNSTRUCTURED NAVIER-STOKES SOLVER ON MOVING AND STATIONARY GRIDS

Robert T. Biedron
NASA Langley Research Center,
Hampton, VA 23681
Veer N. Vatsa
NASA Langley Research Center,
Hampton, VA 23681 and
Harold L. Atkins
NASA Langley Research Center,
Hampton, VA 23681

23rd AIAA Applied Aerodynamics Conference
June 6-9, 2005
Toronto, Ontario, Canada

Simulation of Unsteady Flows Using an Unstructured Navier-Stokes Solver on Moving and Stationary Grids

Robert T. Biedron,[*] Veer N. Vatsa,[†] and Harold L. Atkins[‡]

NASA Langley Research Center, Hampton, VA 23681

Abstract

We apply an unsteady Reynolds-averaged Navier-Stokes (URANS) solver for unstructured grids to time-dependent problems on both moving and stationary grids. Example problems considered are relevant to active flow control and stability and control. Computational results are presented using the Spalart-Allmaras turbulence model and are compared to experimental data. The effect of grid and time-step refinement are examined.

Introduction

Unsteady flows are being studied with increasing interest both computationally and experimentally. There are many reasons for this, but they certainly include the growing realization that unsteady flows exhibit properties that can be exploited for control of advanced flight vehicles. In addition, there is simply the fact that all flight vehicles must maneuver at one point or another and as a result experience unsteady conditions. Motivating the current work are NASA research programs focusing on synthetic jets for flow control and predictive methods for obtaining stability and control (S&C) derivatives.

Thomas, Choudhari, and Joslin[1] have identified several active mechanisms, including synthetic jets, as feasible for improving performance and controllability. Potential benefits from the use of synthetic jets for flow control include reducing drag, increasing lift, and enhancing control forces. In an era of increasing costs of experimental testing at a decreasing number of available facilities, there is a growing emphasis on validation of these flow-control concepts via computational fluid dynamics (CFD). Recently, the CFDVAL2004 workshop,[2] had as its primary objective an assessment of the the state of the art for measuring and computing aerodynamic flows in the presence of synthetic jets. The workshop was comprised of three configurations deemed to be of interest for synthetic jet evaluation. This paper will present a CFD simulation of one of the three configurations, designated by the workshop as Case 2.[2] This case represents an isolated synthetic jet formed by a single diaphragm-mounted piston actuator exhausting into a nominally uniform cross-flow. Multiple measurement techniques, including PIV, LDV, and hot-wire probes were used to generate a large body of experimental data for this configuration. References 2 and 3 describe the details of the experimental setup and geometric configuration.

In the area of stability and control, the emergence of novel aircraft configurations is placing an increased need for predictive methods for S&C derivatives beyond traditional methods. Indeed, the classic view of dynamic system linearization about a reference state with a resulting set of stability derivatives may be insufficient to deliver the required information for either the design of these new configurations or flight simulators for training. In September, 2003, NASA Langley Research Center hosted the first of several planned COMSAC symposia.[4] The inaugural symposium sought to bring together researchers from the stability and control S&C community (typically non experts in CFD) and the CFD community (typically non experts in S&C) to discuss issues that currently limit the use of CFD for S&C applications and to formulate plans for expanded use of CFD in this area. Subsequent to the symposium, NASA Langley Research Center embarked on a multi-year project to evaluate state-of-the-art CFD codes for calculation of both static and dynamic stability and control characteristics of a Blended Wing Body (BWB) configuration. This paper highlights the application of one of the CFD codes under evaluation to problems of forced pitching oscillation, including the BWB.

Governing Equations

The unsteady, compressible, Navier-Stokes equations may be written for a moving or stationary control volume in integral conservation form as

$$\frac{\partial}{\partial t} \int_V \mathbf{q} \, dV + \oint_{\partial V} (\mathbf{F}^* - \mathbf{F}_v) \cdot \hat{\mathbf{n}} \, dS = 0$$

---

[*] Senior Research Scientist, Member AIAA
[†] Senior Research Scientist, Senior Member AIAA
[‡] Senior Research Scientist, Senior Member AIAA

where V is the control volume, bounded by control surface ∂V, with local control volume face speed $|\mathbf{W}|$. The vector $\mathbf{q}$ represents the conserved variables; the vectors $\mathbf{F}^*$ and $\mathbf{F_v}$ represent the convective and diffusive fluxes of the conserved variables, respectively. For the case of a moving control volume, the convective fluxes must account for the relative reduction or enhancement of the flux through the control surface owing to the local control surface velocity. Compared with a stationary control volume with flux vector $\mathbf{F}$, $\mathbf{F}^* = \mathbf{F} - \mathbf{qW}$. Introducing a volume average of $\mathbf{q}$

$$\mathbf{Q} = \frac{\int_V \mathbf{q}\, dV}{V}$$

the conservation equations are

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \oint_{\partial V} (\mathbf{F}^* - \mathbf{F}_v) \cdot \hat{\mathbf{n}}\, dS = 0 \tag{1}$$

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}$$

$$\mathbf{F}^* = \begin{bmatrix} \rho(u - W_x) \\ \rho u(u - W_x) + p \\ \rho v(u - W_x) \\ \rho w(u - W_x) \\ (E + p)(u - W_x) + W_x p \end{bmatrix} \hat{\mathbf{i}} + \begin{bmatrix} \rho(v - W_y) \\ \rho u(v - W_y) \\ \rho v(v - W_y) + p \\ \rho w(v - W_y) \\ (E + p)(v - W_y) + W_y p \end{bmatrix} \hat{\mathbf{j}} + \begin{bmatrix} \rho(w - W_z) \\ \rho u(w - W_z) \\ \rho v(w - W_z) \\ \rho w(w - W_z) + p \\ (E + p)(w - W_z) + W_z p \end{bmatrix} \hat{\mathbf{k}}$$

$$\mathbf{F_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \hat{\mathbf{i}} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \hat{\mathbf{j}} + \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix} \hat{\mathbf{k}}$$

The components of the stress tensor (with the Stokes hypothesis) and heat flux vector are given by

$$\tau_{xx} = \frac{2}{3} \frac{M_\infty}{Re} (\mu + \mu_t) \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right)$$

$$\tau_{yy} = \frac{2}{3} \frac{M_\infty}{Re} (\mu + \mu_t) \left(2\frac{\partial y}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right)$$

$$\tau_{zz} = \frac{2}{3} \frac{M_\infty}{Re} (\mu + \mu_t) \left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right)$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t) \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)$$

$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t) \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)$$

$$q_x = -\frac{M_\infty}{Re(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial T}{\partial x}$$

$$q_y = -\frac{M_\infty}{Re(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial T}{\partial y}$$

$$q_z = -\frac{M_\infty}{Re(\gamma - 1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial z}$$

where μ_t is a turbulent eddy viscosity determined via a suitable turbulence model. The mean flow equations are closed by the equation of state for a perfect gas

$$p = (\gamma - 1)\left(E - \rho\frac{(u^2 + v^2 + w^2)}{2}\right)$$

where p is static pressure and E the total energy per unit volume. The laminar viscosity is obtained via Sutherland's law

$$\mu = \frac{1 + C^*}{T + C^*}T^{3/2}$$

where $C^* = 198.6/T_\infty$. The governing equations above have been made dimensionless using the free-stream density $\tilde{\rho}_\infty$ (where $\tilde{\ }$ denotes a dimensional quantity), speed of sound $\tilde{a}_\infty$, molecular viscosity $\tilde{\mu}_\infty$, thermal conductivity $\tilde{k}_\infty$, and a scaled reference length $\tilde{L}_{ref}$ such that the Reynolds number appearing in the stress tensor and heat flux vector is a Reynolds number per unit length in the mesh.

Solution Algorithm

The unstructured mesh solver used for this study is FUN3D.[5] Within the code, Eq. 1 is discretized over the median dual volume surrounding each mesh point, balancing the time rate of change of the averaged conserved variables in each dual volume with the flux of mass, momentum, and energy through the instantaneous surface of the control volume.

Spatial Discretization

Details of the spatial discretization have been presented in Ref. 5, and will only be summarized here. Within FUN3D, the convective fluxes are computed with a flux splitting scheme, and for second order accuracy the values at dual-cell interfaces are reconstructed using gradients at mesh nodes computed using a least-squares technique. Limiting of the reconstructed values may be employed for flows with strong shocks. For all results presented in this paper, the convective flux scheme used is Roe's flux difference splitting[6] with second order unlimited reconstruction. For tetrahedral meshes, the full viscous fluxes are discretized using a finite-volume formulation in which the required velocity gradients on the dual faces are computed using the Green-Gauss theorem. On tetrahedral meshes this is equivalent to a Galerkin type approximation. For non-tetrahedral meshes, the same Green-Gauss approach can lead to odd-even decoupling. A pure edge-based approach can be used to circumvent the odd-even decoupling issue, but yields only approximate viscous terms. Thus for non-tetrahedral meshes, the edge-based gradients are combined with Green-Gauss gradients, which improves the h-ellipticity of the operator, and allows the complete viscous stresses to be evaluated. For turbulent flows, both the one-equation model of Spalart and Allmaras[7] (SA) and the two-equation SST model of Menter[8] are available. The SA model may be solved loosely coupled to the mean-flow equations or tightly coupled to the mean-flow equations. For all results presented in this paper, the one equation SA model is employed, solved in a loosely coupled fashion.

Few modifications are needed to the spatial discretization summarized above to handle moving meshes. Specifically, the dual face speeds need to be included in the convective flux routines (including boundary routines); the diffusive flux routines remain unaltered. For viscous flows, the no-slip condition at the wall is satisfied by setting the fluid velocity equal to the velocity of the wall itself; this in turn adds a kinetic energy term to the boundary condition for the energy equation at the wall that does not normally appear for stationary meshes.

Temporal Discretization

In this paper we restrict ourselves to the class of problems in which the grid either moves as a rigid body or is stationary. Thus the control volumes are invariant in time and Eq. 1 may be written as

$$V\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{R}$$

Evaluating this equation at time level n+1, and writing the time derivative as a series expansion of successive levels backward in time gives

$$\frac{V}{\Delta t}\left(\phi_{n+1}\mathbf{Q}^{n+1} + \phi_n\mathbf{Q}^n + \phi_{n-1}\mathbf{Q}^{n-1} + \phi_{n-2}\mathbf{Q}^{n-2} + \ldots\right) = \mathbf{R}^{n+1} \qquad (2)$$

3

The sequence $\{\,\phi_n\,\}$ governs the accuracy of the temporal discretization; several choices are described below. At this stage the right-hand side of Eq. 2 could be linearized about time level n+1, resulting in an implicit scheme for $\Delta\mathbf{Q}^n = \mathbf{Q}^{n+1} - \mathbf{Q}^n$. However, the linearization introduces an additional error into the result that is time-step dependent. Furthermore, $\mathbf{R}(\mathbf{Q})$ is difficult to linearize exactly, so that in practice approximate linearizations are often used, introducing another level of error.

To mitigate these linearization errors, as in Reference 9 a pseudo-time term may be added to Eq. 2

$$V\frac{\partial\mathbf{Q}}{\partial\tau} + \frac{V}{\Delta t}(\phi_{n+1}\mathbf{Q}^{n+1} + \phi_n\mathbf{Q}^n + \phi_{n-1}\mathbf{Q}^{n-1} + \phi_{n-2}\mathbf{Q}^{n-2} + \ldots) = \mathbf{R}^{n+1}$$

where the pseudo time is denoted by τ and $0 < \tau < \infty$ during each physical time step. Provided that $\partial\mathbf{Q}/\partial\tau$ vanishes for large τ, during each time step, Eq. 2 is recovered. Discretizing this pseudo-time term with a first order backward difference about pseudo-time level m+1, and noting that $\mathbf{Q}^{m+1} \to \mathbf{Q}^{n+1}$ as $m \to \infty$ gives

$$V\frac{(\mathbf{Q}^{m+1} - \mathbf{Q}^m)}{\Delta\tau} + \frac{V}{\Delta t}(\phi_{n+1}\mathbf{Q}^{m+1} + \phi_n\mathbf{Q}^n + \phi_{n-1}\mathbf{Q}^{n-1} + \phi_{n-2}\mathbf{Q}^{n-2} + \ldots) = \mathbf{R}^{m+1}$$

Defining $\Delta\mathbf{Q}^m = \mathbf{Q}^{m+1} - \mathbf{Q}^m$ and linearizing the right-hand side about pseudo-time level m, yields, after some algebraic manipulation

$$\left[\left(\frac{V}{\Delta\tau} + \frac{V\phi_{n+1}}{\Delta t}\right)\mathbf{I} - \frac{\partial\mathbf{R}^m}{\partial\mathbf{Q}}\right]\Delta\mathbf{Q}^m = \mathbf{R}^m - \frac{V\phi_{n+1}}{\Delta t}(\mathbf{Q}^m - \mathbf{Q}^n) - \frac{V\phi_{n-1}}{\Delta t}(\mathbf{Q}^{n-1} - \mathbf{Q}^n) - \frac{V\phi_{n-2}}{\Delta t}(\mathbf{Q}^{n-2} - \mathbf{Q}^n) - \ldots \quad (3)$$

where the consistency requirement $\Sigma\phi_n = 0$ has been invoked. Equation 3 is the means by which the solution is advanced in time in FUN3D; with the physical time step Δt set to ∞, Eq. 3 reverts to the standard "steady state" scheme described in Ref. 5. At each subiteration m, the linear system represented by Eq. 3 is iteratively solved using a user-specified number of point Jacobi or point Gauss-Seidel sweeps — typically the latter as it converges the linear system approximately twice as fast as point Jacobi. More recently a line solver has been added,[10] allowing points near solid walls to be solved in a more closely coupled fashion. Within the nonlinear iteration process between time steps, the equations are advanced in pseudo time with local time stepping (spatially varying time step $\Delta\tau$ based on a constant CFL number) to accelerate the solution to a steady state in pseudo time, with the physical time step Δt being held constant over the entire mesh. The CFL number may be ramped during the subiterations in order to accelerate the convergence in pseudo time. To reduce computational time, the evaluation of the flux jacobian $\partial\mathbf{R}^m/\partial\mathbf{Q}$ may be periodically frozen during pseudo-time stepping.

Returning to the choice of $\{\,\phi\,\}_n$, if one retains terms to ϕ_{n-2}, various backward-difference schemes up to third order in time may be constructed, as summarized in the following table

order	ϕ_{n+1}	ϕ_n	ϕ_{n-1}	ϕ_{n-2}
1st	1	-1	0	0
2nd	3/2	-2	1/2	0
3rd	11/6	-3	3/2	$-1/3$
$BDF2_{opt}$	$3/2 - \phi_{n-2}$	$-2 + 3\phi_{n-2}$	$1/2 - 3\phi_{n-2}$	$-0.58/3$

Table 1 Coefficients for BDF schemes up to third order

The coefficients for the first-, second- and third-order schemes listed above are well known, with the first- and second-order schemes being stable for any time step. The third-order scheme, while usually found to be stable in practice, is not guaranteed so. The coefficients for the scheme labeled $BDF2_{opt}$ are from Ref. 11, and are a linear combination of second and third order coefficients that lead to a scheme that is stable for any time step, with an order of accuracy that is in between second order and third order.

Temporal Error Control

The subiterative scheme outlined above has been implemented in a number of flow solvers. One issue that consistently arises is the choice of the number of subiterations to use. Converging the subiteration residual at each time step (as measured by the complete right-hand side of Eq. 3) to machine zero is generally prohibitive in terms of computational cost. So the issue becomes a balance between computational cost and performing just enough subiterations to "get the right answer". For the BDF schemes, we can estimate the temporal error incurred at each time step by examining the difference in residual contribution

with two different levels of approximations of time derivatives.[12] The temporal error norm can be used as an exit criteria for terminating the subiteration loop of the dual time stepping process. Basically, we terminate this subiteration process when the residuals (algebraic errors) drop below a specified fraction of the temporal error norm. For all cases presented in this paper, the specified fraction was 0.1, and was chosen based on prior experience. Such a strategy results in uniform temporal accuracy for all time steps, and eliminates the guess work for selecting iteration count or arbitrary level of residual reduction.

Mesh Movement

The initial implementation of time-varying meshes within the FUN3D code were restricted to translation or rotation as a solid body. Solid-body translation is trivial, with cell face speeds directly following from the specified translation velocity. Rigid rotation is also straightforward, though some details are warranted.

Consider a rotation, parallel to the y-axis, of a vector $\mathbf{A}$ lying in the x-z plane, through an angle $\Delta\theta$, to a new position where the vector is denoted by $\mathbf{A}'$. One can show that

$$A_{x'} = A_x cos\Delta\theta + A_z sin\Delta\theta \tag{4}$$
$$A_{z'} = A_z cos\Delta\theta - A_x sin\Delta\theta$$

Transformations for rotation about the x or z axes are obtained by a cyclic permutation of the coordinate directions in Eq. 4. By differentiating Eq. 4 with respect to time, one can show that the corresponding velocity components are

$$\dot{A}_{x'} = A_{z'}\dot{\theta} \tag{5}$$
$$\dot{A}_{z'} = -A_{x'}\dot{\theta}$$

When applied to a position vector, Eq. 4 may be used to obtain expressions for the new mesh point locations and velocities. For example, for rotation parallel to the y-axis, $\mathbf{A} = \mathbf{r} = (x - x_0)\hat{\mathbf{i}} + (z - z_0)\hat{\mathbf{k}}$, with (x_0, z_0) the rotation point, gives the required expressions for the new position $(x' - x_0, z' - z_0)$.

Because the dual-cell volumes are fixed in time, they may be precomputed and stored. Likewise, dual-face areas do not change with time. With rotational motion, dual-face normals do change direction with each time step, but are easily recalculated using Eq. 4. The dual-face speeds, needed to evaluate the fluxes via Eq. 1, are calculated from the mesh point velocities obtained via Eq. 5 with the same Green-Gauss approach used to evaluate the dual-cell normals for static meshes.

Boundary Conditions

The boundary conditions required for solving the Navier-Stokes equations, such as the no-slip, adiabatic wall, far-field and extrapolation conditions, are described in Ref. 5, with the above-mentioned extensions of the no-slip condition and the convective-flux evaluation for moving grids.

For the synthetic jet case described below, a special boundary condition was added to simulate the effect of an oscillating piston. This is done with a periodic velocity transpiration condition imposed at the mean piston surface. Further details of this transpiration condition are given in the section describing the synthetic jet results.

Pitching Airfoil

As a first test of the new moving mesh capability within FUN3D, a pitching 0012 airfoil is considered. The flow and geometry motion conditions are chosen to represent AGARD Case 3,[13] for which experimental data are available. The airfoil is pitched about the quarter cord, according to

$$\alpha = \alpha_m + \alpha_0 sin(\tilde{\omega}\tilde{t}) \tag{6}$$

where α is the instantaneous angle of attack, α_m is the mean angle of attack and α_0 is the pitch amplitude. The pitch frequency $\tilde{\omega}$ may be written in terms of a reduced frequency k_r as $\tilde{\omega} = k_r 2\tilde{U}_\infty/\tilde{c}$, where $\tilde{c}$ is the airfoil chord and $\tilde{U}_\infty$ is the free-steam velocity. Case 3 is defined by the parameter set $\alpha_m = 4.86°$, $\alpha_m = 2.44°$, $k_r = 0.01547$, $M_\infty = 0.6$, $Re_{\tilde{c}} = 4.8x10^6$.

Within FUN3D, the angular displacement is specified as

$$\alpha = \alpha_m + \alpha_0 sin(2\pi k_{cfd}t) \tag{7}$$

where k_{cfd} is a reduced frequency and t is the nondimensional time given by

$$t = \frac{\tilde{t}\tilde{a}_\infty}{\tilde{L}_{ref}}$$

Recall that $\tilde{L}_{ref}$ is the dimensional reference length that makes the Reynolds number appearing in the stress tensor and heat flux vector of Eq. 1 a Reynolds number per unit length in the computational grid. To insure that the motion in the computational model tracks the experimental model, the arguments of the sin functions in Eqs. 6 and 7 must be identical. This leads to the value of k_{cfd} in Eq. 7

$$k_{cfd} = k_r \frac{\tilde{L}_{ref}}{\tilde{c}} \frac{M_\infty}{\pi} \tag{8}$$

The mesh used for this study was obtained by converting a structured mesh into an unstructured hexahedral mesh. In the mesh, the airfoil had a chord of unit length, so that $\tilde{L}_{ref} = \tilde{c}$ in Eq. 8. The original mesh contained 609 points around the airfoil and wake cut, and 257 points away from the airfoil. To evaluate the effect of grid refinement, this fine mesh was also coarsened successively in each direction to create meshes of size 305×129 and 153×65. For these simulations, FUN3D was run in 2D mode, in which the equations are solved on a single $y = $ constant plane. This mode allows faster execution time by avoiding many unnecessary evaluations when there is no variation in the y direction. To assess the effect of time step on the solution, several different time steps were used, ranging from 100 time steps per pitch cycle to 800 time steps per pitch cycle. All computations of the 0012 airfoil were performed using the $BDF2_{opt}$ scheme and utilized the temporal error controller with a cutoff of 0.1.

Figure 1 shows the computed lift coefficient vs angle of attack from the computations on the finest mesh, together with the experimental values from Ref. 13. The agreement is generally good, although there are relatively few experimental data points to determine whether the crossover seen in the computed results is seen in the experiment. While some variation is seen in the computation as the time step is refined from 100 to 200 steps per cycle, very little change is seen with further refinement to 400 and 800 steps per cycle, indicating the computation is essentially temporally converged for this mesh.

Figure 2 shows the corresponding pitching moment results. Pitching moment is always more difficult to accurately capture than lift, so it is not surprising to see more differences with the data than with lift. The largest discrepancy between computation and experiment occurs as the airfoil is pitching down (upper section of the curve), with the largest differences at the lower limit near $\alpha = 2.43°$. In fact, as the time step is refined the computations move further away from the data in the pitch down region. However, the changes become smaller with decreasing time step, again indicating the computation is essentially temporally converged for this mesh. Very little variation with time step is seen in the pitch up region (lower section), for any time step.

In Fig. 3 the effect of grid refinement on the lift variation is shown. For each mesh density, the smallest time step (800 steps per cycle) was used. More variation in the result is seen with mesh refinement than with time step refinement. Figure 4 illustrates the effect of grid refinement on pitching moment variation. Not unexpectedly, pitching moment is quite different between the coarse and medium grids. The change in pitching moment as the grid is refined from 305×129 to 609×257 is minimal, suggesting that an essentially mesh-size invariant solution is obtained for the finest mesh when using 800 steps per cycle. The fact that the computed moment curve on the finest mesh using the finest time step does not exactly replicate the experimental moment curve is perhaps not surprising, since the computation is strictly two dimensional, while no experiment can completely eliminate three-dimensional effects.

Pressure coefficients at two points during the pitch cycle were extracted for comparison with the experimental data. The computed pressure data were extracted from the simulation corresponding to 800 time steps per cycle on the 609×257 mesh. Figure 5 shows pressure coefficients over the airfoil upper and lower surfaces as the airfoil is pitching up, at $\alpha = 5.92°$. The computed and measured pressure coefficients are in excellent agreement. Figure 6 shows the comparison as the airfoil is pitching down, at $\alpha = 2.43°$. At this point in the cycle, the agreement is generally very good, except for a slight under prediction of the peak suction as compared to the experiment.

Pitching Blended Wing Body

Next, a three-dimensional geometry is considered. The Blended Wing Body (BWB) is an advanced concept configuration that should offer an increase in L/D through reduced interference drag and a lower wetted area to volume ratio.[14] Several variants of the BWB have been wind-tunnel tested; the configuration considered here includes winglets and three nacelle/pylon installations. Experimental measurements[15] were obtained with a relatively large model (approximately 7 foot wingspan) in the NASA Langley 14x22 subsonic tunnel. The model, with unpowered, flow-through nacelles, was designed for low speed measurements of static and dynamic stability derivatives. Data were taken for a number of flap/slat configurations; only the "clean" (no flaps/slats) geometry is considered here. As with any wind tunnel testing, mounting interference can be important. For the dynamic derivatives, the model was attached to a rather large mounting device that allowed the model to be pitched up and down in a sinusoidal fashion. The wind-tunnel model mounted on the pitch rig is shown in Figure 7. The top-mounted configuration is shown in the figure, however, the the experimental pitch data presented in this paper was taken with the model mounted from the bottom. Neither the mounting device nor the tunnel are modeled in the computation. Companion static force

American Institute of Aeronautics and Astronautics

and moment computations,[16] including comparison with static data from another experimental facility, suggest the mounting system in the 14x22 tunnel imposed significant interference effects on the measured data.

As with the NACA 0012, the pitching motion used in the BWB tests are described by Eq. 6. For the BWB, a range of α_m, α_0, and $\tilde{\omega}$ was tested in the wind tunnel. For the case presented here, $\alpha_m = 0°$, $\alpha_0 = 5°$, and $k_r = \tilde{\omega}\tilde{\bar{c}}/2\tilde{U}_\infty = 0.07$, where $\tilde{\bar{c}}$ is the mean aerodynamic chord. Due to structural limitations, the tunnel dynamic pressure during the forced pitching had to be kept very low, around 2 psf, so that the resulting tunnel Mach number was approximately 0.036, with a corresponding $Re_{\tilde{\bar{c}}} = 0.7x10^6$. For the computational simulation, the Reynolds number was matched to the experimental value, but the Mach number was artificially boosted to 0.1 in order to avoid slow convergence of the compressible equations at very low Mach numbers. The sensitivity of the results to the change in Mach number from 0.036 to 0.1 was assumed to be negligible. It is important to note that this artificially larger Mach number must also be used when determining the reduced frequency k_{cfd} from Eq. 8. Otherwise the ratio of characteristic time of the mean flow (e.g. the time it takes a fluid particle to traverse the body) and the characteristic time of the forced oscillation will be different between computation and experiment.

The grid for this case was generated within the GridEx[17] framework. The mesh was developed in full-scale meters, so that so that $\tilde{L}_{ref} = 1$ meter and $\tilde{\bar{c}} = 27.13228$ meters. The initial mesh contains approximately 2.3 million nodes combined into tetrahedral cells. Source strengths were chosen to give a normal spacing near the wall leading to $y^+ \sim 1$ at the target Reynolds number. The results presented in this paper are for zero yaw angle, so only one half of the configuration is used, with symmetry boundary conditions on the center plane. Figure 8 shows the surface geometry and symmetry plane mesh.

Figure 9 shows the pitching moment variation with angle of attack from both the computation and the experiment. Because the data is proprietary, labels are omitted from the vertical axes. Arrows labeled "+q" and "-q" indicate pitch up and pitch down segments of the curve, respectively. There is a clear offset between the data and the computation. This is very likely due to the significant post interference mentioned previously. The severity of this interference is indicated in the figure by symbols denoting the experimentally measured static pitching moment for a top-mounted model and a bottom-mounted model at the mean angle of attack. The dynamic data were taken in the wind tunnel with model mounted on the bottom. In the computational study, three time steps were used corresponding to 400, 800 and 1600 steps per pitch cycle. These results were all obtained using the second-order temporal scheme with an error controller cutoff of 0.1. Some variation of pitching moment with time step is seen between 400 and 800 steps per cycle, particularly in the range of $\alpha = +2°$ to $\alpha = +5°$. However, very little change is observed in the pitching moment curve for time steps corresponding to 800 and 1600 steps per cycle, indicating that the pitching moment is essentially temporally converged for this mesh with 1600 steps per cycle. Apart from the static offset in the computed and measured pitching moment curves, the overall shape of the curve is in reasonably good agreement between the experiment and computation.

Figure 10 shows the variation of normal force coefficient with angle of attack. Again there is a static offset from the experimental data, but overall the shape of the computational curve compares very favorably with the shape of the experimental curve. There is very little change in the normal force coefficient with time step; the largest change with time step refinement occurs in the range $\alpha = +2°$ to $\alpha = +5°$, as was observed for the pitching moment coefficient.

To assess the effect of mesh refinement on the solution, a second mesh was generated, containing 4.9 million nodes. The mesh was generated as an essentially uniform refinement of the initial grid. The variation of pitching moment coefficient with mesh size is shown in Fig. 11. Although the static offset from the experiment is slightly larger for the finer mesh, the overall shape of the pitching moment curve is somewhat better captured with the finer mesh. The change in solution between the two meshes is large enough that one cannot infer that the solution is mesh converged. Figure 12 shows the effect of mesh refinement on normal force coefficient, which tends to simply rotate the computational curve into better alignment with the experimental curve, without significant alteration to the overall shape of the computational curve.

Next, computations for $\alpha_m = 8°$ and $\alpha_m = 16°$ were carried out on the 4.9 million node mesh, using the time step corresponding to 1600 steps per cycle. It should be noted that the range of C_M is the same in all the plots, to better indicate the relative variation; likewise the range of C_N is the same in all the plots. Figure 13 and 14 show the pitching moment and normal force coefficient variation, respectively, for a mean angle of attack of $8°$. Again, the static values are also indicated in the plot to show the inherent offset between the two sets of experimental data and the computation. However, apart from the offset, the computed pitching moment variation with angle of attack is in excellent agreement with the experimental data. The same is true for the normal force coefficient variation. The corresponding results for a mean angle of attack of $\alpha_m = 16°$ are shown in Figs. 15 and 16. At this higher mean angle, the computed pitching moment variation is not particularly well captured by the computation, though the normal force coefficient variation, apart from the static offset, is still in reasonable agreement with the experimental data. Why the pitching moment curve does not show better agreement with the data is not certain, although flow visualization of the computed solutions show significantly more separation, and of a different nature at $\alpha_m = 16°$ than at $\alpha_m = 8°$. It may well be that more grid points are required to adequately resolve the more complex flow field at $\alpha_m = 16°$ before better agreement with the experimental data can be obtained.

In stability and control applications, the so-called stability derivatives are widely used. These derivatives arise out of the assumption of small deviations from a reference condition of steady flight. Because of the small perturbation assumption, stability derivatives are typically valid over a limited region around the reference condition; in highly nonlinear flight regimes they may not be valid at all. Because these derivatives are so widely used, and because in many cases they are difficult to obtain experimentally, it is desirable to use computational methods to estimate them. For longitudinal dynamic stability assessment, a parameter of great interest is the "lumped pitch damping derivative"

$$\overline{C}_{M\hat{q}} = \frac{\partial \overline{C}_M}{\partial \hat{q}}$$

where $\hat{q} = \tilde{q}\tilde{c}/(2\tilde{U}_\infty)$. The overbar indicates that this derivative contains both pitch rate (q) and angle of attack rate ($\dot{\alpha}$) terms. In a wind tunnel or flight test (or computation) in which the configuration undergoes pitching about a spanwise axis (with associated angle of attack change), it is not possible to separate the q effect from the $\dot{\alpha}$ effect. In fact, the separation of the two effects is another consequence of the linearized, small perturbation assumption. For a stable aircraft, $\overline{C}_{M\hat{q}}$ should be negative, that is, a positive pitch rate (pitch up) should generate a nose-down moment. The most straightforward method for determining $\overline{C}_{M\hat{q}}$ is to use a finite-difference between C_M values as measured (or computed) while the model is pitching up and pitching down, at the reference (mean) angle of attack. Table 2 shows the relative errors between $\overline{C}_{M\hat{q}}$ found by applying finite differences to both the experimental and computational C_M data, at the three mean angles of attack considered above. Owing to the proprietary nature of the data, percent errors, relative to the experimental data, are given. At each value of α_m, $\overline{C}_{M\hat{q}}$ was found to be negative, indicating a dynamically stable vehicle in pitch over this range of angle of attack. It is interesting to note that although the overall agreement between the computed and measured moment curves is better at $\alpha_m = 8°$ than at $\alpha_m = 0°$ (and significantly better than at $\alpha_m = 16°$), the computed value of $\overline{C}_{M\hat{q}}$ at $\alpha_m = 8°$ is much further away from the measured value than at $\alpha_m = 0°$, and only in somewhat better agreement than at $\alpha_m = 16°$. The excellent agreement between computed and measured $\overline{C}_{M\hat{q}}$ at $\alpha_m = 0°$ is almost certainly fortuitous, and illustrates the pitfall of using a single derived value as a measure of the quality of the computational result.

α_m	% Error
0°	−4
8°	+37
16°	−45

Table 2 Errors in computed values of $\overline{C}_{M\hat{q}}$

Synthetic Jet

The final application presented in this paper is a piston-driven synthetic jet. A schematic of the actuator used in the experiment is shown in Figure 17. Owing to the vertical movement of the piston, flow passed in and out of the cavity through an orifice. It should be noted that while the orifice was of circular cross section, the piston consisted of a rigid square plate attached somewhat asymmetrically to the flexible membrane. Thus the flow into and out of the orifice was not completely symmetric about the centerline. The orifice was located on a splitter plate mounted in a wind tunnel. The Mach number in the tunnel was 0.1, and the boundary layer height near the orifice of approximately 21mm. It should be noted that the cavity was quite shallow relative to its width, and the piston movement caused a considerable change in volume as it cycled up and down.

The geometry of the cavity is simplified by modeling the piston/membrane as a flat surface, with the vertical position of the surface taken such that the "at rest" volume of the cavity is unchanged. To reduce the problem size, the domain is divided at the tunnel centerline and only half of the domain is modeled in the computation. The computational domain extends from 8 jet diameters upstream and to the side of jet center, and 16 jet diameters downstream of the jet center.

The grid for this case was also generated using GridEx, and the wall spacing set so that the resulting $y^+ \sim 1$ for the target conditions. Grid spacings inside the cavity were based on preliminary simulations with steady blowing applied at the piston face. Griding sources were placed around the lip of the jet and along the jet centerline in an effort to improve the clustering in these regions. The resulting mesh contained approximately 255,000 nodes. Fig. 18 shows the grid in symmetry plane for this configuration. To assess the effect of mesh size on the computed solutions, a second mesh of approximately 46,000 nodes was constructed as an essentially uniform coarsening of the finer mesh.

In order to match the conditions in the tunnel, FUN3D was modified to allow a velocity profile to be input. A turbulent velocity profile obtained with FUN3D and the SA turbulence model on a flat plate grid with conditions that gave a thickness of approximately 21mm was used as an inflow condition.

American Institute of Aeronautics and Astronautics

Rather than modeling the piston motion, at the mean piston surface a time-dependent velocity was imposed, given in dimensional form by

$$\tilde{u} = 0 \qquad \tilde{v} = 0 \qquad \tilde{w} = [(\tilde{\rho}\tilde{w})_{\max}/\tilde{\rho}]cos(2\pi\tilde{f}\tilde{t})$$

where $\tilde{f}$ is the piston frequency, 150 Hz. Density and pressure on this boundary were left to evolve as part of the solution process. The value of $(\tilde{\rho}\tilde{w})_{\max}$ is chosen as part of a calibration process to approximately match the peak vertical velocity at the outflow of the orifice between computation and experiment. The value of this calibration constant is $(\tilde{\rho}\tilde{w})_{\max} = 0.007\tilde{\rho}_\infty\tilde{a}_\infty$. All results were obtained with a time step corresponding to 720 steps per actuator cycle.

The experimental data for this configuration at several locations, marked on Fig. 19, are available from the experimental study of Schaeffer and Jenkins.[18] Figures 20 and 21 show the comparison of the computed phase-averaged normal and stream-wise velocities with the experimental data at $\tilde{x}$=50.8 and $\tilde{z}$=0.4 mm location, which corresponds to the center of the jet exit. The overall agreement with the experimental data is fair and present results are comparable to numerical results of other researchers who employed similar boundary conditions[2] to simulate this case. Very little change with mesh refinement is observed in the normal velocity with mesh refinement although the streamwise component shows some improvement with the finer mesh for phase angles between approximately 80° and 220°.

The phase-averaged velocities at a distance of 2 jet diameters downstream of the jet centerline ($\tilde{x}$=63.5 mm) are compared with the experimental data in Figs. 22 and 23. The computational results on the finer mesh compare favorably with the experimental data at this location; the features on the coarser mesh are greatly dissipated. Finally, the time-averaged velocity profiles at $\tilde{x}$=63.5 mm are shown in Figs. 24 and 25. The computational results replicate the experimentally observed features and are again comparable to the CFD results of other researchers.[2] The time-averaged normal velocity shows the greatest sensitivity to mesh refinement. The peak velocity is under predicted on the coarser mesh and over predicted on the finer mesh, although the profile on the finer mesh is in better agreement overall with the experimental data.

Summary

A Navier-Stokes solver for unstructured grids has been extended to allow time-accurate computations on grids that move as a rigid body. The time-accurate capability of the solver has been extended to third-order accurate BDF schemes, including a temporal-error controller that eliminates much of the guesswork involved in determining the proper number of nonlinear subiterations required within a given time step. In addition, a transpiration boundary condition for simplified modeling of synthetic jet flows has been developed.

The modified code has been applied to several test cases of interest to the active flow control and stability and control communities. The influence of time step and mesh size on the solutions was assessed. These initial results show reasonable agreement with available experimental data in most cases. Both the BWB and synthetic jet cases presented here involved simplifications of the physical model — neglect of mounting hardware or tunnel walls in the BWB case and use of a transpiration boundary condition rather than a moving piston in the synthetic jet case. Recently the mesh movement within the FUN3D solver has been extended to include deforming meshes, and the computations for these cases will be repeated taking the mounting system into account for the BWB and the moving piston into account for the synthetic jet.

Acknowledgment

References

[1]Thomas, R.H.; Choudhari, M.M.; and Joslin, R.D.: "Flow and Noise Control: Review and Assessment of Future Directions". NASA TM 2002-211631, April 2002.

[2]CFD Validation of Synthetic Jets and Turbulent Separation Control: Langley Research Center Workshop, Williamsburg, VA, March 29-31, 2004. (http://cfdval2004.larc.nasa.gov)

[3]Yao, C.; Chen, F.J.; Neuhart, D.; and Harris, J.: "Synthetic Jet Flow Field Database for CFD Validation". AIAA Paper 2004-2218, June 2004.

[4]Fremaux, C. M. and Hall, R. M. (Compilers): "Computational Methods for Stability and Control". NASA CP-2004-213028 PT1, April 2004.

[5]Anderson, W. K., and Bonhaus, D. L.: "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids". Computers and Fluids, Vol. 23, No. 1, pp. 1-21, 1994.

[6]Roe, P.; "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes". *J. Comp. Phys.* Vol. 43, pp. 357-372, 1981.

[7]Spalart, P., and Allmaras, S.: "A One-Equation Turbulence Model for Aerodynamic Flows". *La Recherche Aerospatiale*, No. 1, pp. 5-21, 1994.

[8]Menter, F.: "Improved Two-Equation k-ω Turbulence Model for Aerodynamic Flows". NASA TM 103975, 1992.

[9]Rumsey, C., Sanetrik, M., Biedron, R., Melson, N., and Parlette, E.: "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations". *Computers and Fluids*, Vol. 25, No. 2, pp. 217-236, 1996.

American Institute of Aeronautics and Astronautics

[10]Nielsen, E., Lu, J., Park, M., and Darmofal, D.: "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids". *Computers and Fluids*, Vol. 33, No. 9, pp. 1131-1155.

[11]Nyukhtikov, M., Smelova, N., Mitchell, B. E., and Holmes, D. G.: "Optimized Dual-Time Stepping Technique For Time-Accurate Navier-Stokes Calculation". Proceedings of the 10th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines, 2003.

[12]Vatsa, V. N., and Carpenter, M. H.: "Higher-Order Temporal Schemes with Error Controllers for Unsteady Navier-Stokes Equations". AIAA Paper 2005-5245, June 2005.

[13]Landon, R. H.:"NACA 0012. Oscillation and Transient Pitching". Appearing in *Compendium of Unsteady Aerodynamic Measurement*. AGARD-R-702, 1982.

[14]Qin, N., Vavalle, A., Le Moigne, A., Laban, M., Hackett, K., and Weinerfelt, P.: "Aerodynamic considerations of blended wing body aircraft". *Progress in Aerospace Sciences*. Vol. 40, No. 6, August 2004, pp. 321-343.

[15]Vicroy, D. D.: *private communication*.

[16]Pao, S. P., Biedron, R. T., Park, M. A., Fremaux, C. M., and Vicroy, D. D.: "Navier-Stokes Computations of Longitudinal Forces and Moments for a Blended Wing Body" AIAA-0045, 2005.

[17]Jones, W. T.: "GridEx - An Integrated Grid Generation Package for CFD". AIAA 2003-3849, 2003.

[18]Schaeffler, N. W., and Jenkins, L. N.: "The Isolated Synthetic Jet in Crossflow: A Benchmark for Flow Control Simulation". AIAA 2004-2219, June-July 2004.

[19]Rumsey, C. L.: "Computation of a Synthetic Jet in a Turbulent Cross-Flow Boundary Layer". NASA/TM-2004-213273, October 2004.

American Institute of Aeronautics and Astronautics
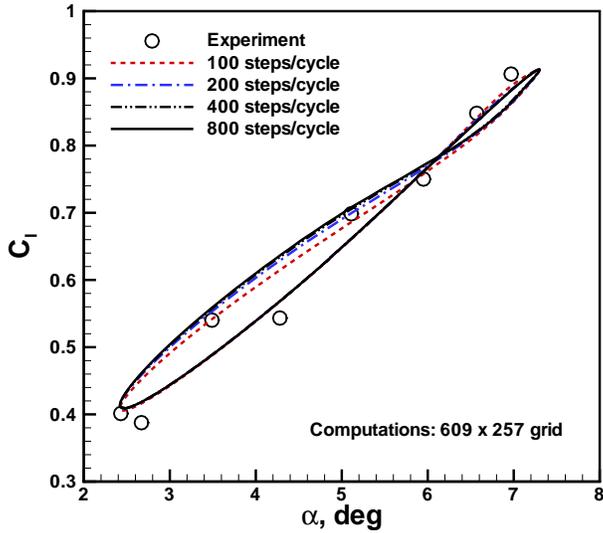
Fig. 1   Effect of time-step refinement on lift coefficient vs. angle of attack, 0012 airfoil, AGARD Case 3.
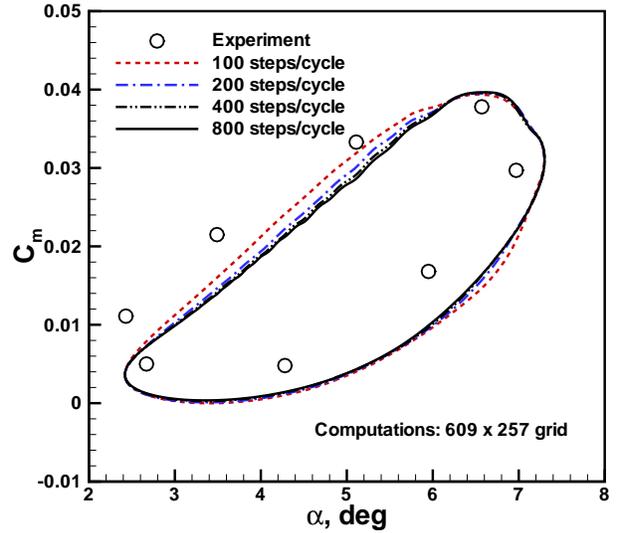
Fig. 2   Effect of time-step refinement on pitching moment coefficient vs. angle of attack, 0012 airfoil, AGARD Case 3.
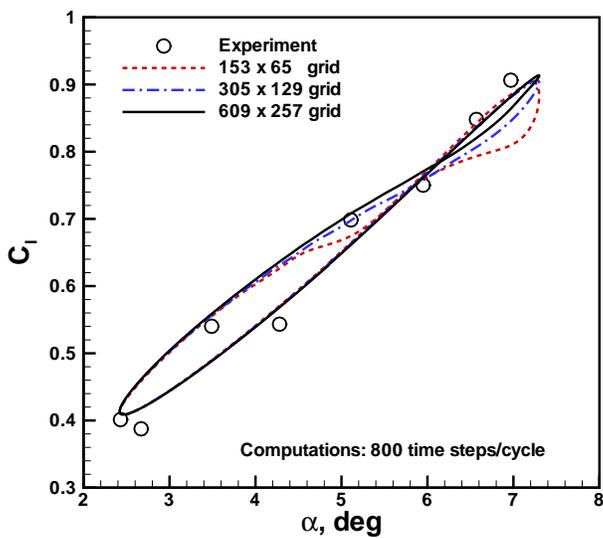
Fig. 3   Effect of grid refinement on lift coefficient vs. angle of attack, 0012 airfoil, AGARD Case 3.
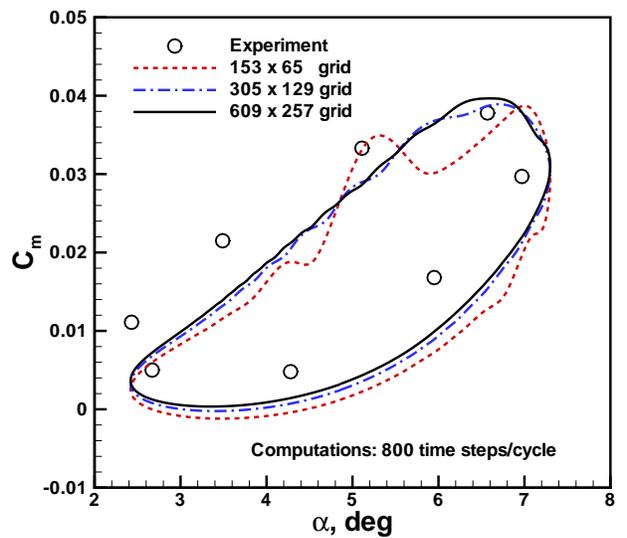
Fig. 4   Effect of grid refinement on pitching moment coefficient vs. angle of attack, 0012 airfoil, AGARD Case 3.

11
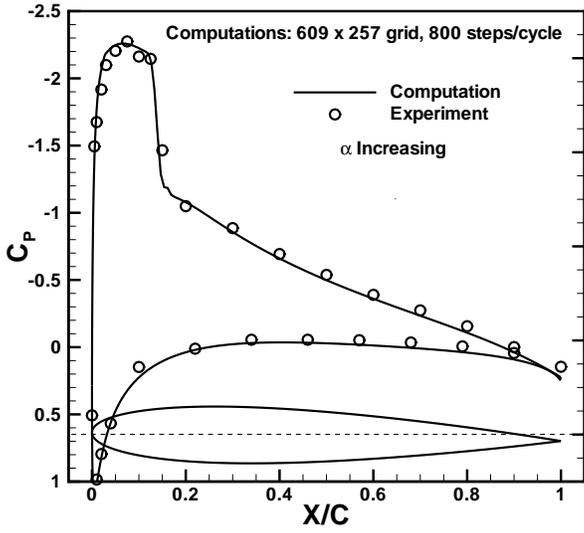
Fig. 5 Pressure coefficient distribution during pitch-up, $\alpha = 5.92°$, 0012 Airfoil, AGARD Case 3.
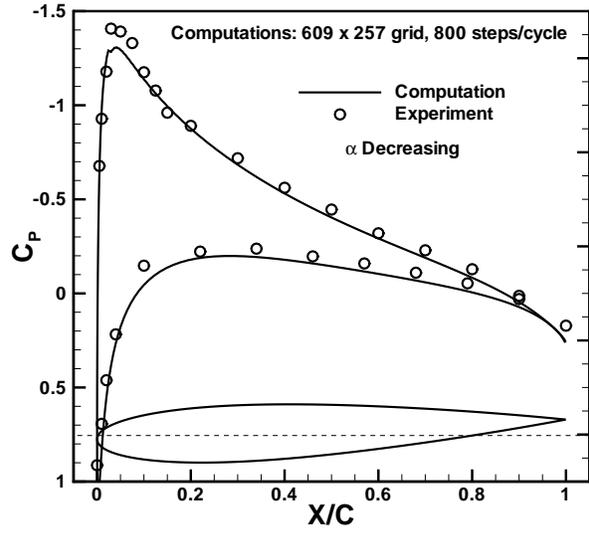
Fig. 6 Pressure coefficient distribution during pitch-down, $\alpha = 2.43°$, 0012 Airfoil, AGARD Case 3.
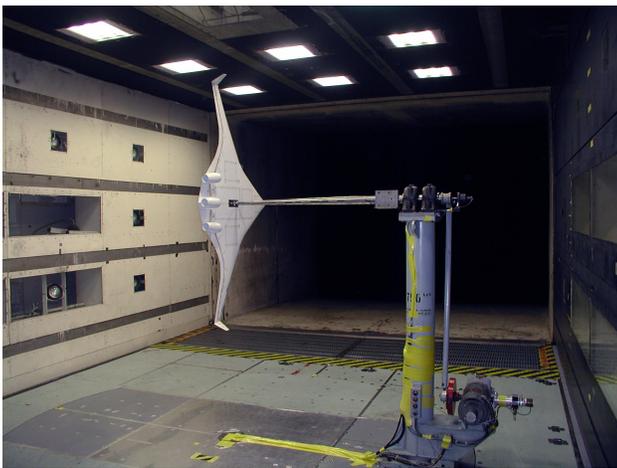
Fig. 7 3% scale model Blended Wing Body on the dynamic test rig in the 14×22 Wind Tunnel; top mounting shown.
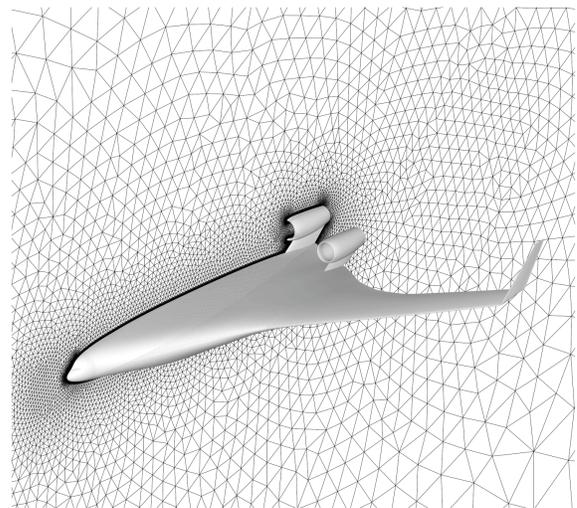
Fig. 8 Computational model of Blended Wing Body; 2.3 million node mesh.

12
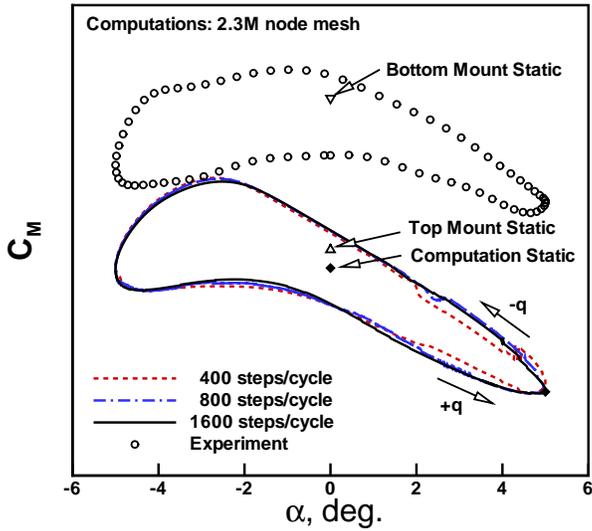
Fig. 9 Effect of time-step refinement on BWB pitching moment coefficient vs. angle of attack, $\alpha_m = 0°$; "+q" and "-q" indicate the pitch-up and pitch-down legs of the oscillation.
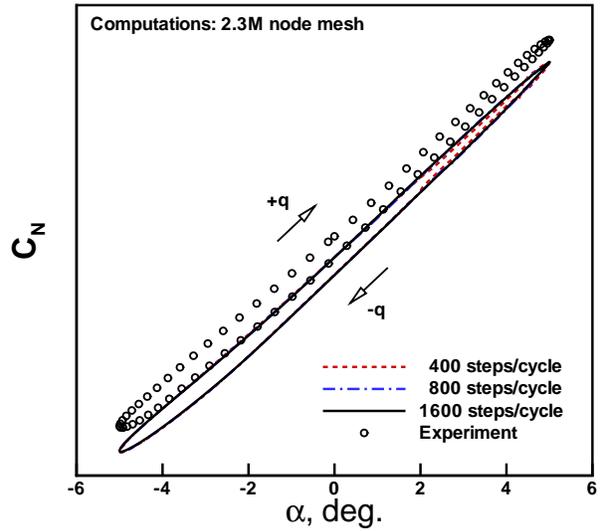
Fig. 10 Effect of time-step refinement on BWB normal force coefficient vs. angle of attack, $\alpha_m = 0°$.
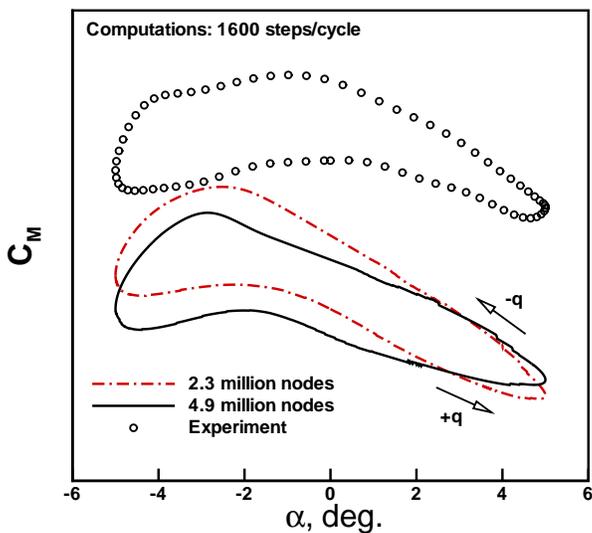
Fig. 11 Effect of grid refinement on BWB pitching moment coefficient vs. angle of attack, $\alpha_m = 0°$.
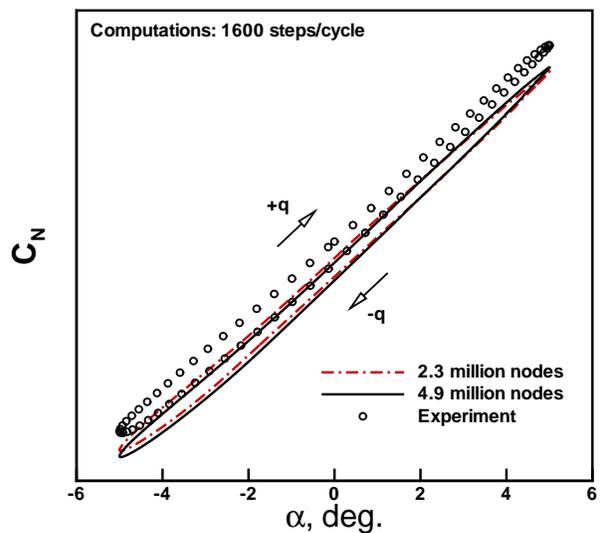
Fig. 12 Effect of grid refinement on BWB normal force coefficient vs. angle of attack, $\alpha_m = 0°$.

13
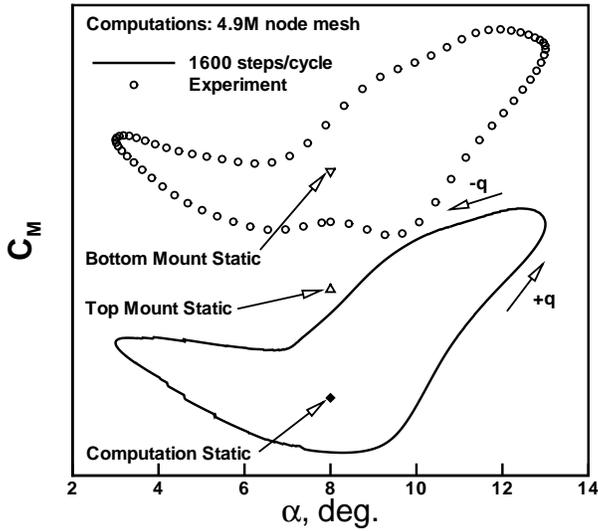
Fig. 13 BWB pitching moment coefficient vs. angle of attack, $\alpha_{\mathrm{m}} = 8°.$
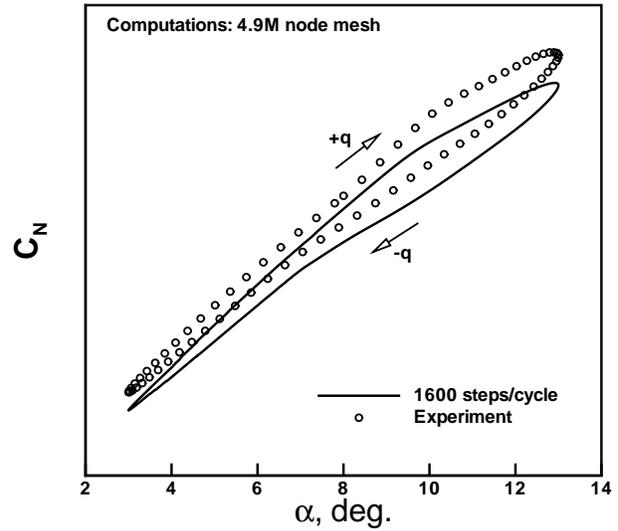
Fig. 14   BWB normal force coefficient vs. angle of attack, $\alpha_{\mathrm{m}} = 8°.$
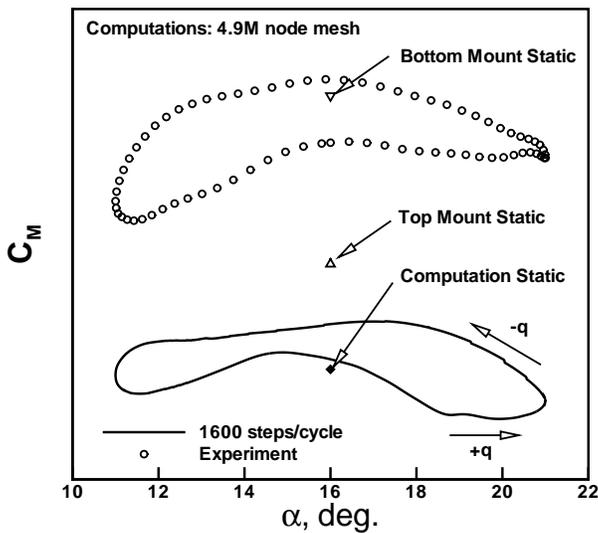
Fig. 15 BWB pitching moment coefficient vs. angle of attack, $\alpha_{\mathrm{m}} = 16°.$

Fig. 16   BWB normal force coefficient vs. angle of attack, $\alpha_{\mathrm{m}} = 16°.$

M = 0.1 →

6.35mm

2.78mm

membrane is attached to all four walls

oscillates +-0.77mm
about neutral position

solid plate - 2mm thick
(approx square 51mm x 50.4mm)
driven up and down
electro-mechanically

flexible membrane
(approx square 76mm x 76.3mm)
Neutral position is raised approx
1.1mm when tunnel is on and
piston is operating

(not drawn to scale)
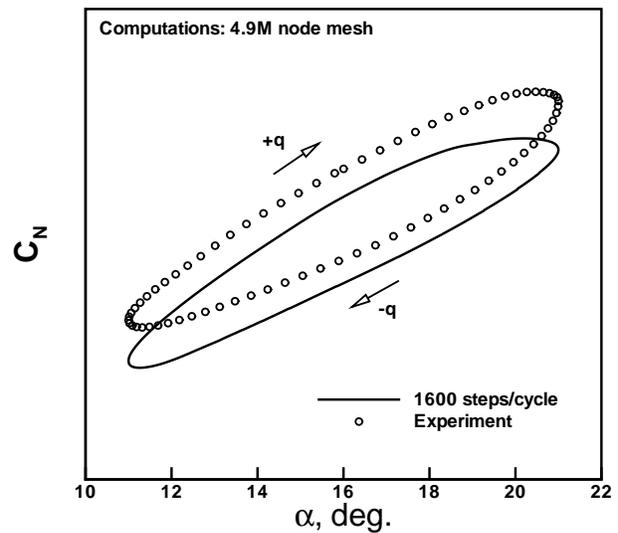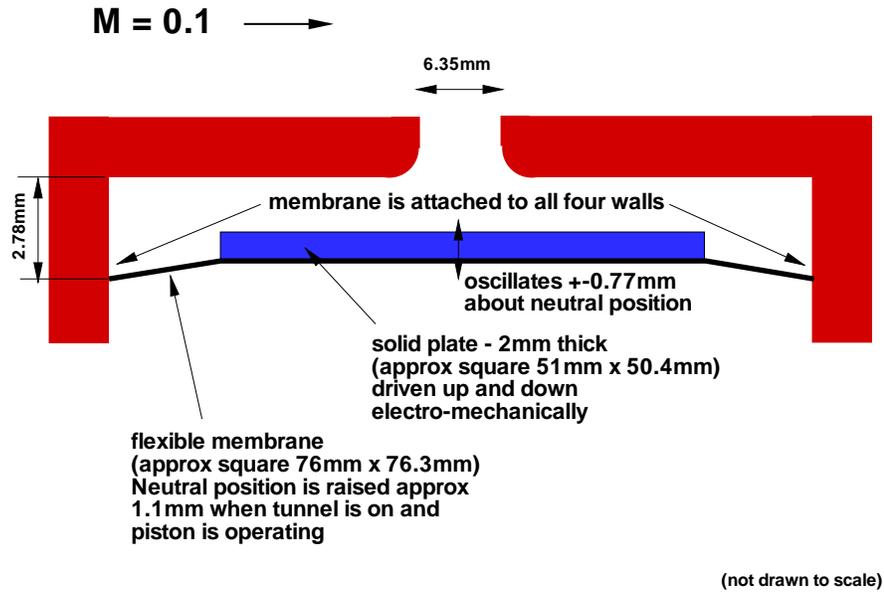
Fig. 17   Sketch of fluidic actuator, showing cavity and piston (from Ref. 19).

x(mm)

z(mm)
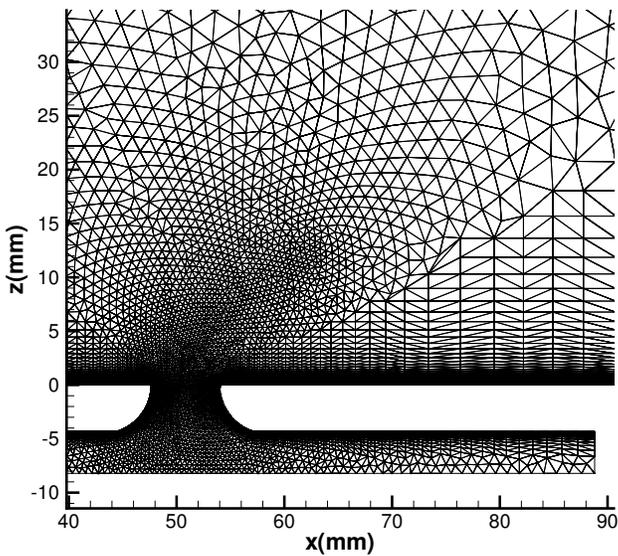
Fig. 18   Symmetry plane grid for the actuator.

x=57.15 (1D downstream)
x=63.5 (2D downstream)
x=76.2 (4D downstream)
x=101.6 (8D downstream)
x=50.8
x=44.45
x=0
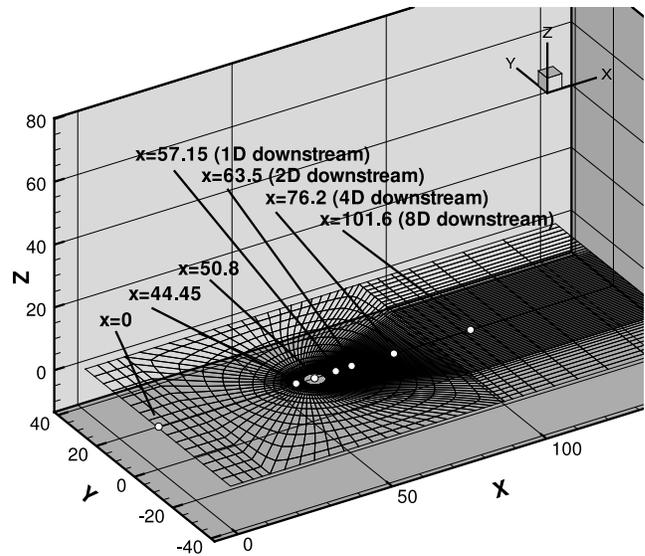
Fig. 19   Sketch showing locations of data comparison with experiment (from Ref. 19).

American Institute of Aeronautics and Astronautics
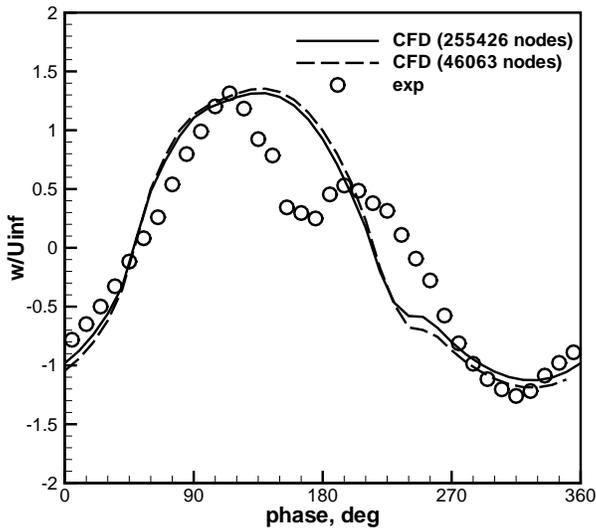
Fig. 20   Normal velocity component as a function of actuator phase angle, x=50.8 mm.
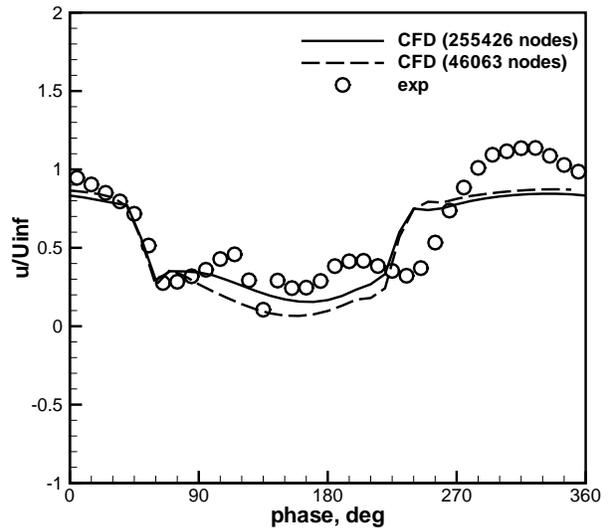

Fig. 21   Streamwise velocity component as a function of actuator phase angle, x=50.8 mm.
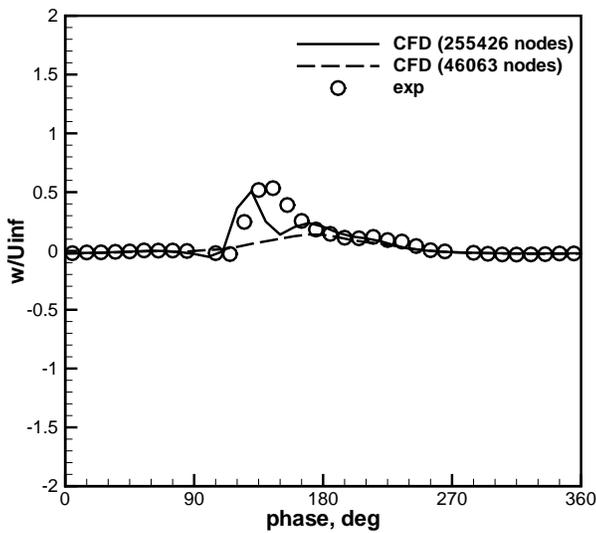

Fig. 22   Normal velocity component as a function of actuator phase angle, x=63.5 mm.
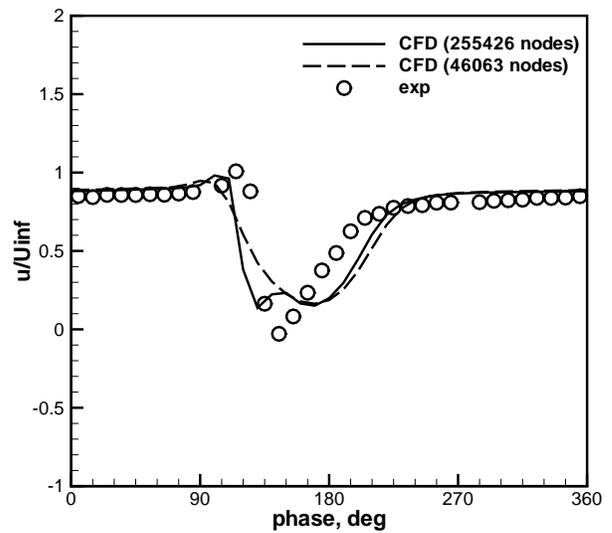

**Fig. 23   Streamwise velocity component as a function of actuator phase angle, x=63.5 mm.**

16
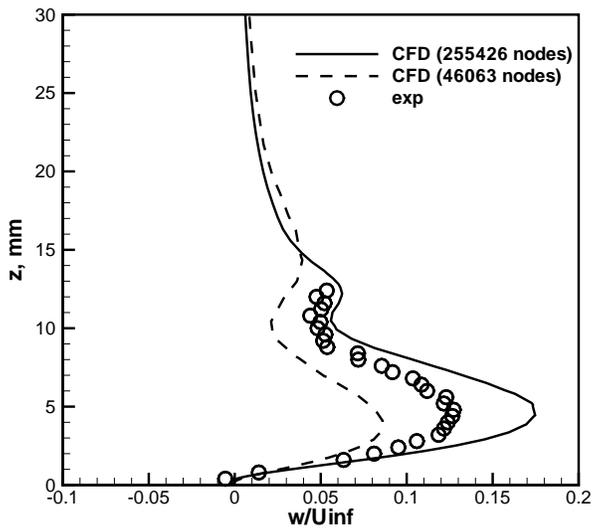
**Fig. 24   Time averaged normal velocity component, x=63.5 mm.**
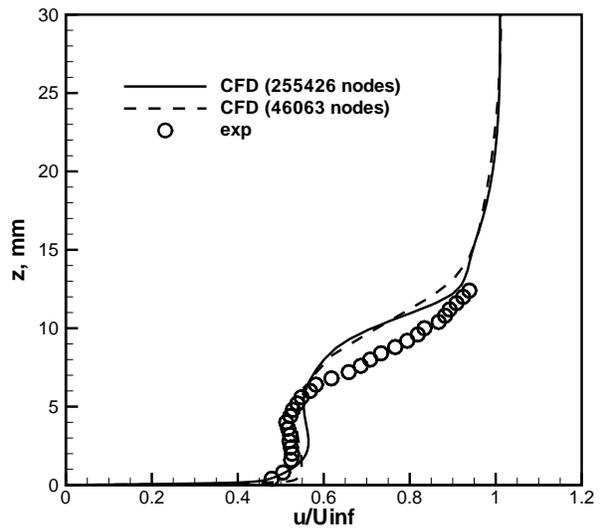
Fig. 25   Time averaged streamwise velocity component, x=63.5 mm.