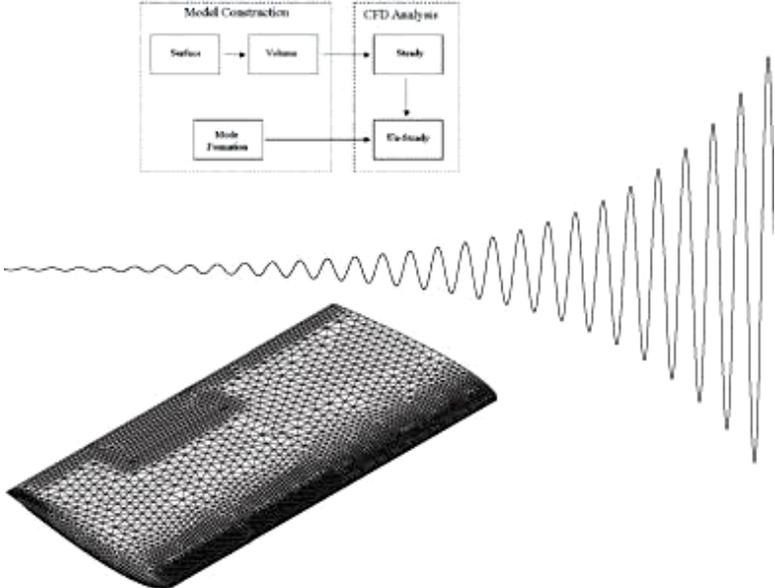


Euler3d Validation with the BACT Aeroelastic Test Case



Submitted by: Charles O'Neill

Euler3d Validation with the BACT Aeroelastic Test Case

Submitted by: Charles O'Neill

Submitted to:

May 3, 2001

The CASELab validated the Euler3d computational fluid dynamics computer program. The Euler3d code correctly predicted experimental static and dynamic Benchmark Active Controls Technology (BACT) pressure data at subsonic and transonic Mach numbers. The Euler3d code did not accurately predicted the BACT flutter boundaries. I recommend further research to fix the Euler3d code.

Abstract

The OSU CASELab developed a new computational fluid dynamics program, Euler3d. Before presenting any Euler3d results, the CASELab must validate the Euler3d program. This report validates and evaluates the Euler3d program against a known aeroelastic test case. The solution validated the Euler3d program against the Benchmark Active Controls Technology (BACT) aeroelastic test case. This test case provides experimental frequency, pressure and modal data for a wing. I established four validation criteria based on the Euler3d's grid resolution, steady pressure distribution, system identification, and flutter boundary. The Euler3d program partially passes the validation process. Euler3d passes the grid resolution and steady pressure validation. The program failed the unsteady, system identification and flutter boundary validation. Further research recommended in this report will fix the unsteady problems. While this validation process failed, the Euler3d program is still a promising CFD development.

Table of Contents

Abstract	iii
List of Illustrations	v
Glossary	vi
Introduction	1
A Description of Flutter and Aeroelasticity	1
The Purpose of Code Validation	2
Problem Description	2
An Aeroelasticity and Flutter History	2
Theodorsen and Garrick's Linear Flutter Theory (1930-1940)	2
Jet Propulsion (1940-1960)	3
Computer Flutter Modeling (1960-present)	3
Current Flutter Prediction	4
The New CFD code, Euler3d	4
The Benchmark Active Controls Technology (BACT) Wing	5
The BACT Geometry	5
Published BACT Data	6
Methodology	6
Modeling the BACT with Euler3d	6
System Identification	7
The Validation Criteria	7
Criterion 1: Finite Element Grid Resolution	8
Criterion 2: Steady Pressure Distribution	8
Criterion 3: BACT System Identification	8
Criterion 4: Flutter Boundary	8
Analysis of Solution	8
Finite Element Grid Resolution	8
Steady Pressure Distribution	9
Mach 0.51	9
Mach 0.82	10
BACT System Identification	11
Flutter Boundary	11
Implementation Budget	12
Personnel Requirements	12
Computational Requirements	12
Conclusions	13
Recommendations	13
References	14

List of Illustrations

Figure 1. Aircraft with Destroyed Horizontal Stabilizer (Bisplinghoff & Ashley, 1962).	2
Figure 2. Maximum Aircraft Velocities in the Jet Age (Bisplinghoff & Ashley, 1962)...	3
Figure 3. BACT Wing Geometry (Stephens, 1998).	5
Figure 4. Euler3d Modeling Steps	6
Figure 5. System Identification Integration	7
Figure 6. BACT Surface Grid	9
Figure 7. Steady Pressure Distribution at $\alpha=0$ deg. and Mach 0.51	10
Figure 8. Steady Pressure Distribution at $\alpha=0$ deg. and Mach 0.82	10
Figure 9. BACT Flutter Boundary	12
Table 1. Published BACT Structural Parameters (Rivera, et al, 1992).	6

Glossary

ARMA	Auto Regression Moving Average
BMP	Benchmark Models Program
BACT	Benchmark Active Controls Technology
CASELab	Computational AeroServoElasticity Laboratory
CFD	Computational Fluid Dynamics
Euler3d	Eulerian Finite Element Non-Inertial CFD
NASA	National Aeronautics and Space Administration

Introduction

Euler3d is an Eulerian based finite element computational fluid dynamics program. The CASELab at Oklahoma State University uses Euler3d for aeroelastic prediction and research. As part of the total validation process, I tested the Euler3d program against a known aeroelastic test case.

This project validated the Euler3d computational fluid dynamics program for both steady and unsteady flows with experimental Benchmark Active Controls Technology (BACT) data. I established criteria guidelines and applied those guidelines to the Euler3d BACT model.

Flutter validation will help the CASELab to defend its research and will open new research grant opportunities. As part of the CASELab research, a new 3D Eulerian finite element computational fluid dynamics (CFD) code was written. While the CASELab staff already finished the initial verification and debugging process, they must validate the Euler3d code with experimental data.

This report introduces aeroelasticity and validates a computer flutter prediction program. I discuss the Euler3d output for steady and unsteady validation cases.

A Description of Flutter and Aeroelasticity

Flutter consists of the flow-induced vibration of a bendable structure. For example, a flapping street sign during high wind and a waving flag are both flutter. While fluttering street signs and flags are innocuous, the twisting and flapping of an aircraft will cause a disaster.

This twisting and flapping occurs because of the interactions between mass and stiffness of both the structure and the surrounding fluid. When structural motions couple to form out of phase oscillations, the total energy of the system increases and flutter occurs. Structural component damping influences the flutter amplitude and phase. Specifically, if any structural vibration mode has zero damping, the mode is capable of flutter. Quantifying these vibrational modes obviously requires tremendous computational power.

Computers number crunching permits analysis of arbitrary geometries or flight conditions. Computational fluid dynamics (CFD) results from integrating computer processing power with mathematical flutter descriptions. Using CFD for determining steady forces and moments is common. Unsteady analysis has recently been added to predict pressures and forces for moving and rotating bodies.

The Purpose of Code Validation

Before professionally presenting the Euler3d code, the CASELab must first perform validation test cases. Schlesinger defines validation as “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” (1979). Without validation, all Euler3d output is of questionable quality and will not be accepted.

Problem Description

The problem definition consists of four parts. First, I introduce a history of aeroelasticity and flutter. Next, I discuss the current flutter prediction methods. Then, I introduce the new Euler3d computer program and its relationship to the CASELab. Finally, I present the Benchmark Active Controls Technology aeroelasticity test case.

An Aeroelasticity and Flutter History

Flutter research developed from vibration and dynamics research. Until the development of aircraft, engineering research placed little emphasis on flutter. Land and water vehicles moved too slowly to encounter destructive flutter. Aircraft require light and large structures, which are conducive to flutter. The aircraft in Figure 1 shows the destructive effects of flutter.

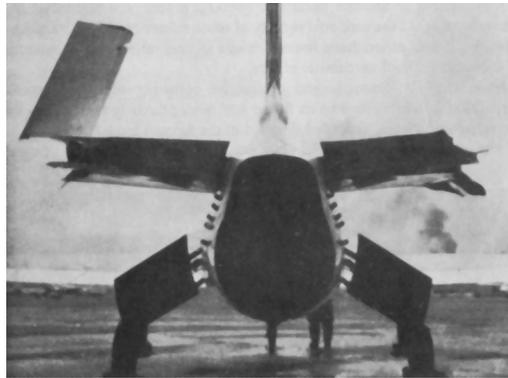


Figure 1. Aircraft with Destroyed Horizontal Stabilizer (Bisplinghoff & Ashley, 1962).

The fighter aircraft shown in Figure 1 experienced horizontal stabilizer flutter. The outboard tips of the stabilizer either tore off the aircraft as seen on the starboard side or experienced drastic bending as seen on the port side. Unfortunately, the aircraft designers neglected flutter effects and the pilot almost died.

Theodorsen and Garrick's Linear Flutter Theory (1930-1940)

Theodorsen and Garrick derived the first mathematical solution for flutter in the 1930s and 40s. Earlier studies gave clear evidence of flutter but lacked rigorous mathematical descriptions. The Theodorsen and Garrick solution considered a 2D flutter case with two degrees of freedom, plunge and pitch or a 3D case with the addition of an aileron. Their

solution was the state of the art until the 1950's. From Bisplinghoff and Ashley, "the [NACA] issued approximately as many Technical Notes during the nine-year period from 1950 to its date of absorption into NASA as were released during the previous thirty-five years of its existence" (1962). Recently, Zeiler (2000) found that while the Theodorsen and Garrick's theory captures the relevant physics, Theodorsen and Garrick incorrectly computed the resulting flutter boundary plot. These errors propagated throughout the historical literature so that many flutter and aeroelasticity references contain incorrect flutter boundaries (Zeiler, 2000).

Jet Propulsion (1940-1960)

During the 1950's, transonic and supersonic aircraft accelerated flutter research drastically. Aircraft technology finally reached a point where flutter routinely occurred. Recently developed jet and rocket engines contributed to increased flight speeds. Figure 2 shows the drastic increase in maximum flight speed.

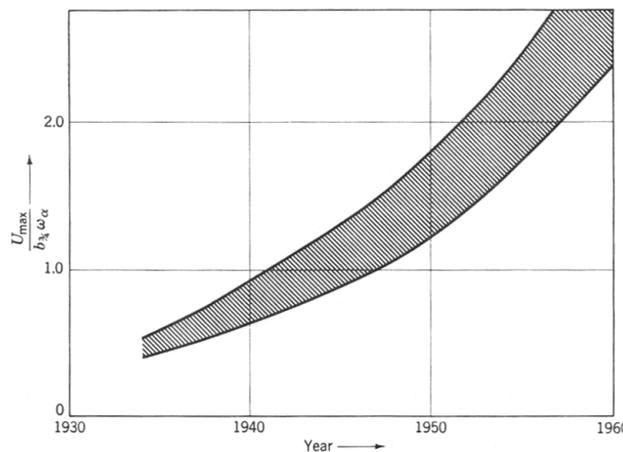


Figure 2. Maximum Aircraft Velocities in the Jet Age (Bisplinghoff & Ashley, 1962).

The velocity, y-axis, is nondimensionalized by the flutter natural frequency. As seen in the figure, maximum aircraft velocities increased dramatically after widespread development of the jet engine in 1940. The increase corresponds is exponential. Thus, flutter will increasingly dominate aircraft performance. Linear approximations, such as the Theodorsen and Garrick theory, can not fully capture the relevant physics.

Computer Flutter Modeling (1960-present)

Computational methods advanced flutter knowledge beyond linear approximations. Accounting for the entire flow field requires the memory and speed of computers. In the 1960's, computer power became powerful enough to model fluid flow. In 1998, Bennet and Edwards stated that "in the past decade, workstation-type machines have attained the performance level of the supercomputers of the previous decade and the cost of the computation has decreased by between two and three orders of magnitude." Larger and more complicated geometries met the increase in computer speed. Since accurate aeroelastic solutions require a large modeling domain, large computational hurdles remain. Because CFD codes determine single arbitrary flow solutions, flutter prediction

is not trivial. Since flutter boundaries are related to both the flow geometry and the flow dynamic pressure, total CFD flutter solutions do not exist. Flutter research demands a never-ending cycle of faster computers and more complicated models.

Current Flutter Prediction

Aerospace engineers traditionally relegated flutter testing to the wind tunnel or estimates from basic theory. Time and cost limit these methods in aircraft design. CFD offers accurate results for arbitrary geometries. Wind tunnel tests require both the construction of a model and an adequate test facility. Additionally, the lag time between the paper design and the wind tunnel results can be considerable. Furthermore, any configuration change requires a change of the test model.

CFD offers a more direct approach to finding flutter. Integration of surface pressures along the body gives the resulting aerodynamic forces and moments. In general, the most complicated geometry can be solved with the proper selection of a CFD method. Finding an efficient, solvable and accurate method of determining pressures is a problem. Computing speed, storage space and geometry complexities limit computational modeling for arbitrary aircraft configurations. CFD solutions to arbitrary aerodynamic problems are now available due to increases in computer power.

CFD literature contains ingenious methods to create better results from less powerful computers. Remeshing the elements increases the solution accuracy of a finite element method. Remeshing changes element spacing near high gradients. This results in an iterative method of solving an initial grid and updating the elements with the solution. Not surprisingly, Shapiro found that remeshing results in better solutions than an overall fine grid. While remeshing initially sounds feasible, it easily consumes more than 20% of the total computational time (Stephens, 1998). This is especially inefficient if the structure repeatability is oriented in the same direction. Another remeshing problem consists of changing the element distribution in a manner that reflects the flow patterns without distorting the following solution.

CFD analysis has limitations. Low Mach number analysis is particularly difficult. Pressure gradients must remain smooth in low Mach number computations. Poorly defined gradients drastically affect surface pressures. To properly define the gradients the grid requires extra elements in high gradient areas.

The New CFD code, Euler3d

Tim Cowan, a CASELab research assistant, developed a new non-inertial CFD code. The code is named Euler3d because it solves the non-viscous fluid Euler equations. The new Euler3d code reduced the number of settings as compared to the old CFD code. The new non-inertial code moves and rotates any 3D structure in 6 degrees of motion.

Two similar fluid flow governing equations exist, Navier-Stokes and Euler. Euler equations neglect viscosity while Navier-Stokes doesn't. Technically, Navier-Stokes solutions are more precise. However, the computations required to generate a Navier-

Stokes general solution vastly outnumber that of an Euler solution. Because Euler3d solves the Euler equations and not a general Navier-Stokes solution, the flow solution neglects boundary layer effects. The presence of relatively small boundary layers justifies using a non-viscous flow solver for most applications.

The Benchmark Active Controls Technology (BACT) Wing

NASA Langley formed a Benchmark Models Program (BMP) to measure and record flutter solutions for use with computational fluid dynamics codes (Rivera, etc 1992). The BMP studies included the BACT wing. NASA published the BACT results solely to verify aeroelastic and flutter codes.

The BACT Geometry

The BACT geometry consists of a rectangular non-tapered non-twisted wing. A NACA 0012 forms both the root and tip airfoil. Figure 3 shows the basic BACT geometry.

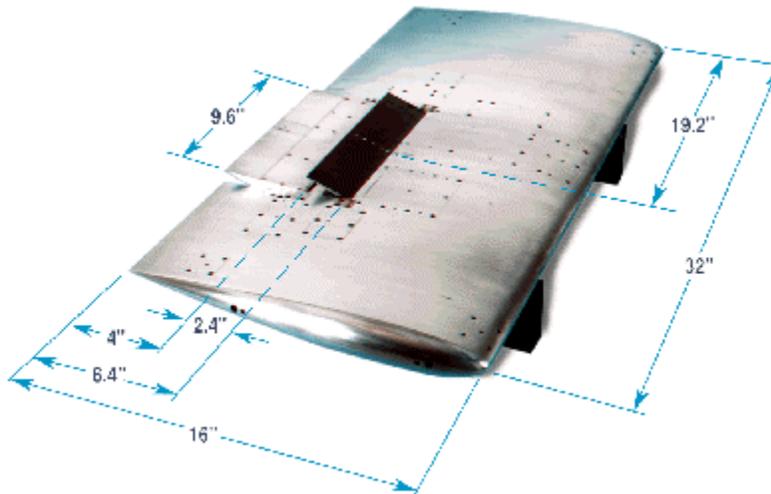


Figure 3. BACT Wing Geometry (Stephens, 1998).

To assist CFD modeling, NASA Langley created a simple BACT geometry as seen in Figure 3. The wing contains an aileron and a spoiler at the 60 percent half-span. The wingtip is rounded. NASA Langley put a series of 80 pressure transducers along the wing top and bottom at 60 percent and 95 percent half-span.

A pitch and plunge apparatus, PAPA, supported the BACT wing during wind tunnel testing (Rivera, etc. 1992). The PAPA device allowed for an adjustable dynamic mass and stiffness. Adjusting the weight distribution of the PAPA device changed the structural properties of the wing. While theoretically this allowed for testing different structural configurations, NASA Langley only tested one configuration. Table 1 gives the reported BACT structural configuration.

Mode	Frequency [Hz]	Structural Damping [g]	Stiffness	Mass
Plunge	3.36	0.00024	2659 lbs/ft	1.0 slug
Pitch	5.20	0.00024	2897 ft-lbs/rad	1.0 slug-ft ²

Table 1. Published BACT Structural Parameters (Rivera, et al, 1992).

The frequencies given for the pitch and plunge modes consist of the single degree of freedom frequency. Although subtle, NASA Langley placed the elastic axis exactly on the center of gravity. Thus, the plunge and pitch mode frequencies are independent. As explained by Stephens (1998), placing the elastic axis coincident with the center of gravity causes large experimental uncertainty.

Published BACT Data

The BMP group published steady and unsteady BACT measurements in NASA-TM-104211 (Rivera, etc 1992). The report presents steady pressure results for several different Mach numbers. These pressures were determined from a series of 80 pressure transducers mounted at both 60 percent and 95 percent span (Rivera, etc 1992). For the steady cases, they tested at Mach numbers of 0.3 to 0.8. For two steady tests, the wing operated at 0 and 1 degree angle of attack.

Methodology

Determining the validity of the Euler3d code consisted of three stages. First, the CASELab staff created a BACT model. Next, I created a system identification for the BACT model. Finally, I applied the validation criteria to the Euler3d BACT model.

Modeling the BACT with Euler3d

The Euler3d CFD program requires specific model geometry definitions. Modeling the BACT with Euler3d consisted of two stages: model construction and CFD analysis. Figure 4 shows the required steps and order.

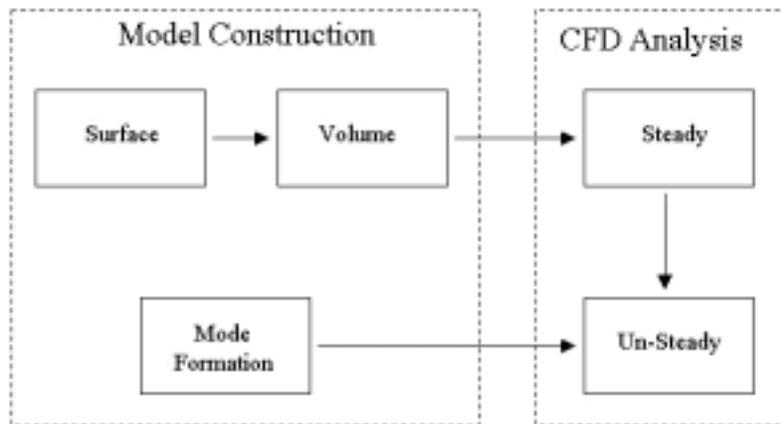


Figure 4. Euler3d Modeling Steps

The first stage consists of model construction. Model construction requires three steps. First, a grid generation program defines the surface elements. Next, another grid generation program defines the elements within the modeling domain. Finally, analysis of motion defines the mode shapes for each degree of freedom.

The second stage concerns CFD analysis. CFD analysis consists of a steady and an unsteady computation. The steady analysis requires the grid definitions from the model construction. Unsteady analysis requires both a previous steady computation and the modes formation from model construction.

System Identification

System identification concerns predicting output states from known input states. System identification predicts unsteady response from an ARMA model. Figure 5 shows how the system identification model integrates into the aerodynamic and structural models.

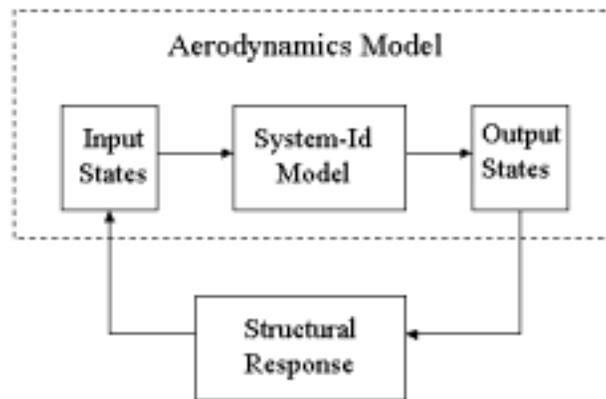


Figure 5. System Identification Integration

From Figure 5, system identification links only the input and output states of the aerodynamics. There is no direct interaction between the system identification model and the structural response. Thus, system identification decouples the structural and aerodynamic solutions. Structural changes require no changes in the aerodynamics model. This system identification property provides a tremendous advantage in aeroelastic tailoring and aircraft design.

Another advantage of system identification concerns speed. Cowan (2001) estimates a 94% time reduction. While the system identification process requires a special unsteady training input, “[once] a model is constructed for a given structure and Mach number, it can be executed repeatedly at different dynamic pressures to search for the dynamic divergence pressure” (Cowan, 2001).

The Validation Criteria

Criterion for evaluating the resulting solution quantifies the Euler3d output’s accuracy and precision. The code must meet the following four criteria.

Criterion 1: Finite Element Grid Resolution

The Euler3d code must solve geometries without an excessive number of volume elements compared to the old CFD code. The solution speed of a solution is highly dependent on the number of elements. Doubling the number of elements increases the solution time by more than twice.

Criterion 2: Steady Pressure Distribution

The Euler3d solution should accurately predict the steady pressure distribution. Flutter analysis requires an accurate steady pressure model. If the Euler3d code prediction is incorrect for the steady pressures, the resulting unsteady pressures and forces have no hope of being correct. The BACT research measured steady pressure distributions at 60% and 95% half-span.

Criterion 3: BACT System Identification

System identification consists of modeling the BACT's unsteady aerodynamics. The unsteady pressure results should match that of the unsteady BACT wing data. If the system identification is good, an arbitrary wing movement results in similar forces for both the system identification and the actual CFD output.

Criterion 4: Flutter Boundary

A flutter boundary consists of a dynamic pressure flutter limit over a Mach number range. The NASA Langley reported the BACT flutter boundary between Mach 0.3 and 0.82. The Euler3d solution needs to accurately predict these boundaries. It is expected that a total match of the experimental data is unlikely. However, the solution should contain the general trends and magnitudes of the experimental data.

Analysis of Solution

The solution analysis consists of five ordered steps. First, I test the finite element grid resolution of the Euler3d program. Next, I validate the steady pressure distribution output from Euler3d. Then, I test the system identification process. Next, I analyzed the BACT's flutter boundary. Finally, I analyzed the Euler3d integration and implementation into a typical aeroelastic problem.

Finite Element Grid Resolution

The Euler3d code meets the grid resolution criterion. The BACT wing case required 22000 surface elements and 600000 volume elements. The Euler3d code required no surface or volume grid tweaking. Figure 6 shows the wing elements along the top surface.

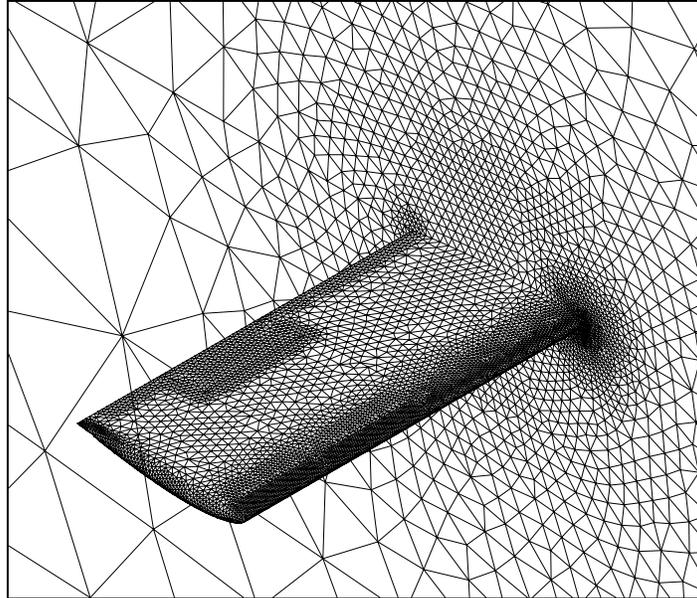


Figure 6. BACT Surface Grid

As expected and seen in Figure 6, the leading and trailing edges require considerably more elements than the mid-chord. Line sources run from the wing root to the tip at both the leading and trailing edges. For later control mode analysis, the grid around the aileron required considerable refining. Addition of an aileron further tested the Euler3d code beyond that required for this unsteady analysis.

Steady Pressure Distribution

The CASLab evaluated the steady Euler3d BACT model at six Mach numbers. I present only the pressure distributions for the low and high Mach number tests: 0.51 and 0.82.

Mach 0.51

The Euler3d code correctly determined the Mach 0.51 steady pressures. The pressure cut plane lies at 60% span. Figure 7 shows the steady Mach 0.51 solution pressure distribution.

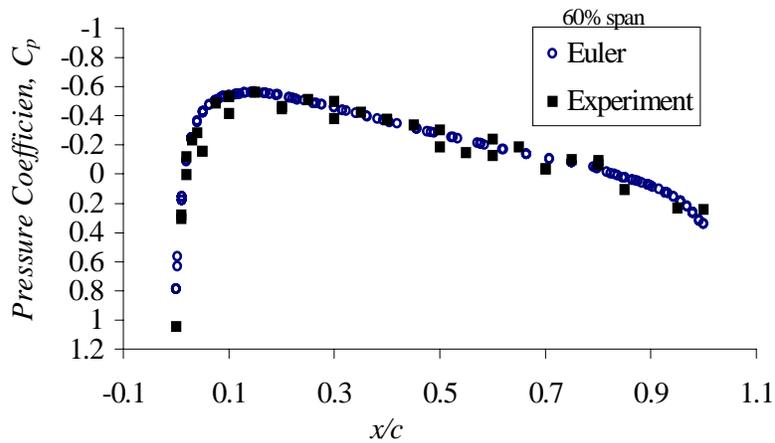


Figure 7. Steady Pressure Distribution at $\alpha=0$ deg. and Mach 0.51

From Figure 7, the calculated and experimental pressure distributions closely match. No transonic flow exists at Mach 0.51 as apparent from the smooth pressure distribution. At x/c of 0.1 to 0.3, the Euler3d solution over predicts the pressure. This pressure over-prediction may be due to a turbulence strip added to the experimental model (Rivera, 1992). Additionally, the Euler3d code only modeled non-viscous effects while the experimental BACT wing experienced viscous airflow. Generally, viscosity does not highly influence pressure distributions. Overall, the Euler3d code correctly predicted the BACT pressure distribution.

Mach 0.82

The Euler3d code correctly predicted the Mach 0.82 BACT test case. The code also correctly captured the standing normal shock. Figure 8 shows the calculated and experimental pressure distributions.

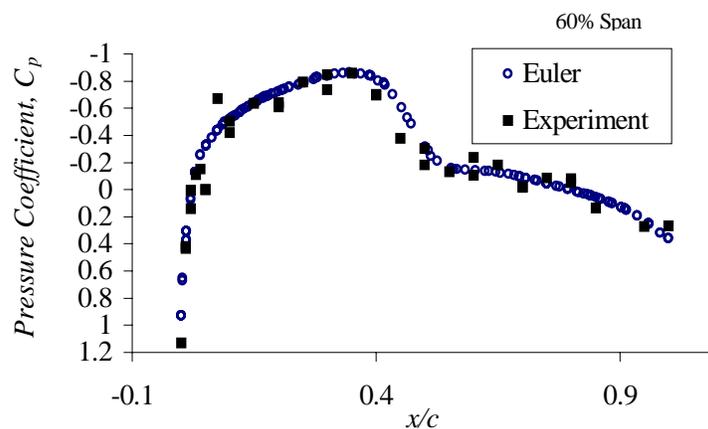


Figure 8. Steady Pressure Distribution at $\alpha=0$ deg. and Mach 0.82

The Euler3d pressure distribution matches the experimental distribution given in the figure. Again at x/c of 0.1 to 0.3, a small discrepancy occurs. The larger discrepancies in

the Mach 0.82 experimental pressure values are probably caused by local supersonic flow near the turbulence strip.

From Figure 8, a shock occurs on the wing. The Euler3d code accurately predicted the shock location and strength. Because this solution exceeds the airfoil's critical Mach number of 0.77, I expected a shock.

Overall, the Euler3d code correctly predicted both subsonic and transonic flow conditions on the BACT wing. The Euler3d code meets the steady pressure distribution criteria.

BACT System Identification

The CASELab performed system identification on the BACT for six Mach numbers. We encountered problems with the process. Only three successful models resulted from over 20 system identification attempts.

The CASELab staff encountered difficulties with the BACT system identification. We found two problems. First, the Euler3d code contained programming bugs. These bugs rendered useless 2 months of unsteady calculations. Second, the multistep unsteady solution calculated noisy solutions because the training signal frequency response was too low. This low training signal frequency response diluted the Euler3d output. A higher frequency variable amplitude multistep signal failed similarly. The CASELab staff is currently investigating better training signals.

I performed system identification for six Mach numbers: 0.51, 0.67, 0.71, 0.77, 0.80 and 0.82. Adequate models resulted only from three Mach numbers: 0.51, 0.71 and 0.82. Currently, Euler3d does not meet the system identification criterion.

Flutter Boundary

I determined a flutter boundary from the partial system identification information. However, since an accurate flutter boundary depends on accurate system identification, I

didn't expect a perfect boundary. The BACT flutter boundary is given in Figure 9.

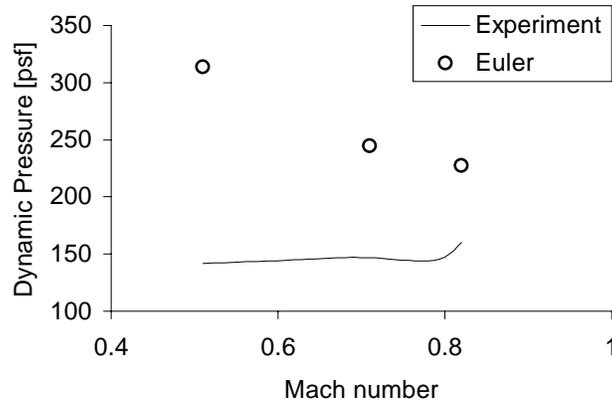


Figure 9. BACT Flutter Boundary

As seen in Figure 9, the resulting Euler3d derived boundary does not resemble the BACT experimental results. At Mach 0.51, my Euler3d boundary missed the actual boundary by a factor of two. Aside from the general decrease, the Euler3d boundary does not contain a transonic dip or a sonic increase. The flutter boundary validation failed.

Implementation Budget

Computational flutter implementation consists of personnel and computer time. Because the overall model complexity and size influence the final times, the implementation budget assumes a flutter case similar to the BACT. Overall, determining a flutter boundary requires at least 1 month and 600 computer hours. The overall budget is consistent with previous CFD programs.

Personnel Requirements

CASELab personnel must complete three stages during any flutter boundary test. First, they must research the new test case and input the geometry and structural parameters into Euler3d. The research and input stage requires 1 week. Second, they must setup and administer the Euler3d program for each phase of the computations. Computations need at least 2 weeks; however, the computations do not require the continuous human presence. Finally, the data must be analyzed and reported. Data analysis and reports typically requires 2 weeks.

Computational Requirements

The computational budget requires computer time for five ordered processes. Computation times vary depending on the model's complexity and size. The BACT lies on the lower end. First, the computer must grid the domain. The BACT grid generation required 1 hour. Second, the computer must perform a steady CFD solution. The steady analysis required 20 hours per solution. Due to grid and Euler3d parameter refining, the

BACT validation required at least three steady solutions before finding a good solution at each Mach number. Next, we compute a multistep unsteady solution. Each BACT multistep solution took 14 hours. One multistep is needed per Mach number. Next, a computer program derives an ARMA model. The BACT ARMA model only takes a few minutes to run and analyze. Finally, the computer must compute a transient unsteady solution. Because Euler3d requires over a day per cycle, we performed only one unsteady transient solution. The unsteady solution ran for 250 hours and completed seven full cycles.

Conclusions

The CASELab partially validated the Euler3d CFD code. The Euler3d code correctly predicts static pressure distributions for subsonic and supersonic Mach numbers. The code predicted steady transonic shocks correctly. Unfortunately, the system identification portion of Euler3d failed to predict flutter boundaries. The multistep requires a higher frequency response training data than Euler3d currently outputs.

I am disappointed with the system identification failure. Previous CASELab research computed flutter boundaries within 5 percent; however, this research failed to accurately predict the boundary within 100% at Mach 0.51.

The Euler3d program is not ready for aeroelastic use. While the steady pressure results were excellent, Euler3d did not accurately predict a flutter boundary.

Recommendations

I was unable to fully validate the Euler3d code with the BACT test case. Euler3d correctly determined the steady solutions; however, the unsteady solutions were incorrect. Because the system identification process failed, I recommend that the CASELab staff concentrate on the following projects:

References

- Bisplinghoff, R., Ashely, H., & Halfman, R., (1955). Aeroelasticity, Reading, MA: Addison Wesley Longman.
- Cowan, T., Arena, A., & Gupta, K., (2001). Accelerating computational fluid dynamics based aeroelastic predictions using system identification. Journal of Aircraft, Vol 38, No. 1, pp 811-87.
- Rivera, J., Dansberry, B., Durham, M., Bennett, R. & Silva, W., (1992). Pressure measurements on a rectangular wing with a NACA 0012 airfoil during conventional flutter. NASA-TM-104211.
- Schlesinger, S., (1979). Terminology for Model Credibility. Simulation, Vol. 32, No. 3, pp. 103-104.
- Stephens, C., (1998). CFD-Based Aeroservoelastic predictions on a Benchmark Configuration using the transpiration method, Oklahoma State University, Stillwater, OK.
- Zeiler, T., (2000). Results of Theodorsen and Garrick Revisited. Journal of Aircraft, Vol 37, No. 5, pp. 918-919.