Control of Aeroelastic Structures Based on a Computational Reduced Order Modeling Method

Charles Hindman*

Advisors:

Mark Balas#

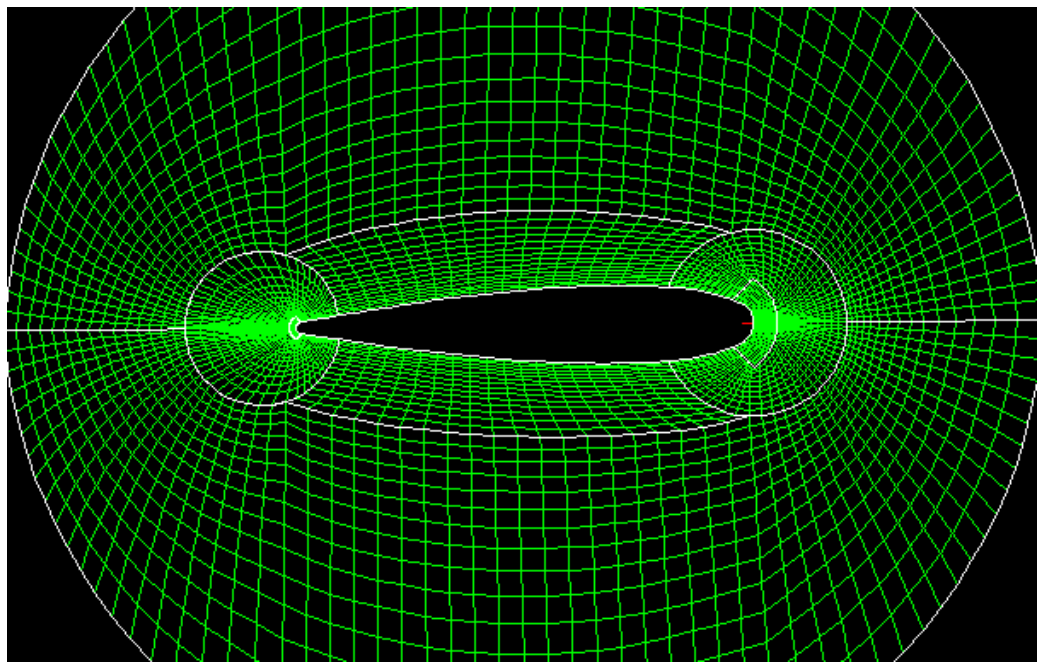
Michel Lesoinne#

DOE Computational Science Graduate Fellowship Conference

July 16, 2003

 # Center for Aerospace Structures, University of Colorado at Boulder

*** Air Force Research Laboratory**



Aeroelastic Instability: Flutter

- Flutter occurs when the fluid surrounding a structure feeds back dynamic energy into the structure instead of absorbing it. Typically a structure will be stable up to a limiting velocity (the flutter velocity) for given conditions.

- The crash of a F117 Stealth Fighter in 1997 was linked to flutter (and many others…)

Flutter Suppression

- Passive Flutter Suppression Techniques
  - Local or global stiffening
    - Adds weight, cost, requires redesign
  - Mass balancing
    - Moves components around – requires redesign, may not be feasible
  - Avoidance
    - Requires operation below the flutter velocity – reduces performance
- Active Flutter Suppression
  - An onboard automatic control system actuates a control surface to suppress flutter
    - First tested in 1973 on a B-52: achieved flight above the flutter speed. Problems with model accuracy / robustness.

Motivation

- We wish to stabilize aeroelastic structures (for example, to suppress flutter) by using automatic actuation
- Past attempts at flutter suppression have been based on “small” empirical and theoretical models that approximate the aerodynamics
  - These techniques have had some success, but suffer from the underlying approximations used in the aerodynamic models
- Several highly accurate computational methods have been created in recent years to simulate the time response of aeroelastic systems.
  - These time integration methods do not easily provide ways to develop rational control laws (they are too large and complex).
- A model is needed that matches all the relevant dynamics of a full scale aeroelastic simulation and yet is still small enough for standard control design techniques to apply.

Computational Aeroelasticity

- The nonlinear computational aeroelastic simulation used in this research is based on a 3-field approach:

$$\left(A(u)w\right)_{,t} + \overline{F}\left(w, x, \dot{x}\right) = 0$$

$$M\ddot{u} + f_{\text{int}}\left(u, \dot{u}\right) = f_{ext}\left(u, w\right)$$

$$\tilde{M}\ddot{x} + \tilde{D}\dot{x} + \tilde{K}x = K_c u$$

- Separate fluid and structure codes are used, coupled by a set of matched interface nodes
- A staggered half time step integration is used to advance the solution in time

Modern Control Design

- Most modern linear control theory is based on a state space system representation:
  - <u>System</u> <u>Controller</u>

$$\dot{x} = Ax + Bu \qquad u = Gx \qquad \Rightarrow \dot{x} = (A + BG)x$$

$$y = Cx + Du$$

- Where:
  - x: system state
  - u: control input
  - y: system output
- By design of the gain matrix G we can alter the response of our system – it acts “as if” it is more damped, for example

Controller Design

- For large systems, the design of the gain matrix G becomes extremely difficult (plus the controller becomes slower)
- Also, access to the state matrix may be impossible (no measurements), in which case a separate model must be developed to estimate the state from whatever measurements are available (y).

System

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Observer

$$\dot{\hat{x}} = A\hat{x} + Bu + L(\hat{y} - y)$$

$$\hat{y} = C\hat{x}$$

Controller

$$u = G\hat{x}$$

Reduced Order Modeling

- Reduced Order Modeling is the generic name for a class of methods that attempt to approximate a high order (linear or nonlinear) dynamical system by a very low order, typically linear, approximation.
- A plethora of techniques have been developed for reduced order modeling:
  - Eigenvalue (Modal) Truncation
  - Balanced Model Reduction (Approximate)
  - Karhunen-Loeve or P.O.D. (snapshots)
  - System ID methods
  - Hybrid techniques
  - -and many, many more

Modal Truncation

- Modal truncation uses a diagonalizing projection T applied to the (A,B,C) system:

$$x_r = T_R x \quad T_L{}^T A T_R = \Lambda_r \left(diag\right)$$

$$\dot{x}_r = a_r x_r + b_r u \qquad a_r = T_L^T A T_R \quad b_r = T_L^T B$$

$$y_r = c_r x_r + Du \qquad c_r = C T_R$$

- The error for the reduced transfer function is given by:

$$\left\| T - T_r \right\|_\infty = \sum_{i=r+1}^{n} \frac{\bar{\sigma}\left(c_i b_i\right)}{\left|\Re\left(\lambda_i\right)\right|}$$

- This error bound is problematic for fluid systems!

Balanced Reductions

- A Balancing transformation of a system is a special coordinate transformation that makes the system grammians equal and diagonal.

$$W_c = \int_0^\infty e^{At} BB^T e^{A^T t} dt \qquad W_o = \int_0^\infty e^{A^T t} C^T C e^{At} dt$$

$$AW_c + W_c A^T + BB^T = 0 \qquad A^T W_o + W_o A + C^T C = 0$$

- This approach takes into account the system inputs and outputs, which is what we are interested in.
- An error bound on the reduced order model is given by:

$$\left\| G(s) - G_r(s) \right\|_\infty \le 2 \sum_{i=r+1}^{n} \sigma_i$$

- For typical systems, σ_i drops off very rapidly.

Model Reduction Approach

- Linearization: a large scale nonlinear CFD code is used to create a linearized model about some steady-state operating point
- Structure: The structure is reduced to a small number of modes by eigenmodal truncation
- Fluid: The linearized fluid is reduced to a small number of states by an approximate balancing method, using the coupling with the structure as the input and output matrices
- This results in a reduced order model that retains the structural modes we wish to control and the relevant aeroelastic dynamics

Linearization

- The nonlinear flux equation is linearized about an operating point ($u_o = v_o = 0$, $w_o = w(steady\ state)$)

$$\left(A(u)w\right)_{,t} + F(w,u,y) = 0$$

  - This method only requires access to the numerical flux function, $F(w,y,u)$, and cell volumes $A(u)$.

- The Linearized equations are:

$$\begin{bmatrix} \dot{w} \\ \dot{y} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} -A^{-1}H & -A^{-1}(E+C)K^* & -A^{-1}GK^* \\ M^{-1}P & 0 & -M^{-1}K_s \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} w \\ y \\ u \end{bmatrix}$$

w – fluid state vector

u – structural displacement vector

y – structural velocity vector

  - Where:

$$H_o = \frac{\partial F}{\partial w}(w_o, u_o, y_o) \qquad E_o = \frac{\partial A}{\partial u} w_o \qquad K_s = K_o - \frac{\partial f_{ext}}{\partial u}$$

$$G_o = \frac{\partial F}{\partial u}(w_o, u_o, y_o) \qquad C_o = \frac{\partial F}{\partial y}(w_o, u_o, y_o) \qquad P_o = \frac{\partial f_{ext}}{\partial w}$$

(Farhat, Lesoinne)

Approximate Balancing

- There are many algorithms for computing the balancing transformation matrix directly.
- None of these methods are suitable for large, sparse problems
  - They form and use the (generally dense) grammians
  - Or they use the (generally dense) Cholesky factors of the grammians
- Instead, various approximate methods have been developed
  - UASI
  - The Laub – Gudmundsson algorithm
  - Alternating Direction Implicit
  - LRCF-ADI

ADI

- The Alternating Direction Implicit (LRCF-ADI) method is an approximate method for solving the full-rank continuous time Lyapunov equation:

$$AX + XA^T = -BB^T$$

- The main idea in ADI is to use an iterative technique that converges rapidly (Peaceman & Rachford, Wachpress):

$$\begin{aligned} X_0 &= 0 \\ (A + p_i I_n) X_{i-1/2} &= -BB^T - X_{i-1}\left(A^T - p_i I_n\right) \\ (A + \overline{p}_i I_n) X_i^T &= -BB^T - X_{i-1/2}^T\left(A^T - \overline{p}_i I_n\right) \end{aligned}$$

- Where p is a set of shift parameters selected with some heuristic for rapid convergence.

LRCF-ADI

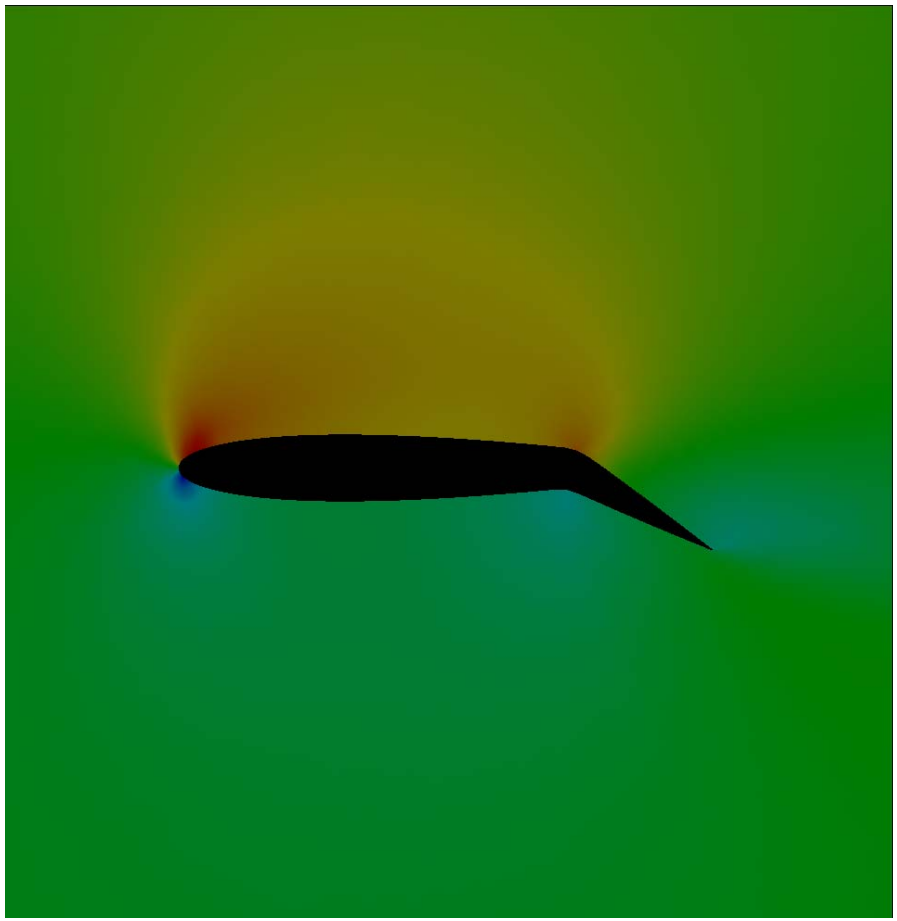
- The Low Rank Cholesky Factorization –ADI method replaces the Xj iterates in the ADI algorithm with a low rank Cholesky factorization:

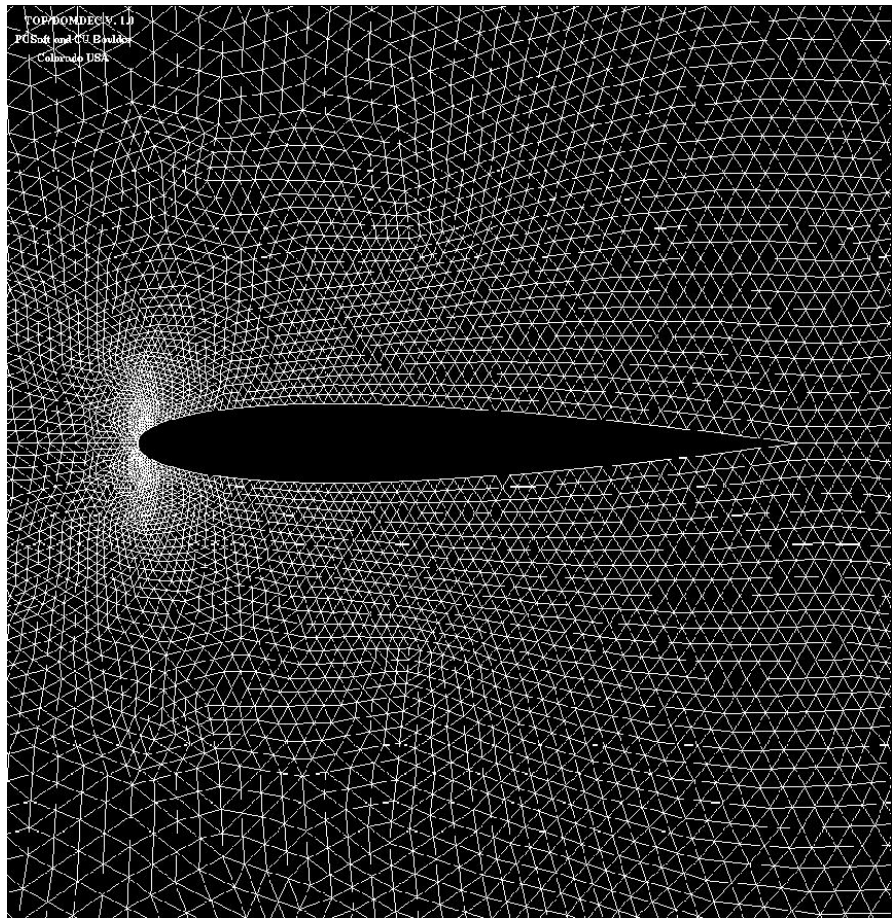
$$X_j = Z_j Z_j^T$$

  - Where Zj has j*m columns, where m is the number of columns in B or C’
- There are many advantages to this algorithm:
  - Never forms the full-scale grammians (preserves sparsity).
  - Only requires solutions of (complex, shifted and transposed) linear systems, which can be developed from existing simulation codes.
  - Is easily parallelized
  - Converges quickly, as long as the shift parameters p_i are chosen in some (sub) optimal manner (they approximate the eigenspectrum of A).

2-D: NACA0012 Airfoil

- This system consists of a rigid airfoil with an integral flap, restrained by rotational and vertical displacement springs, in 2-D Euler flow with ~ 3 - 50,000 DOF's (for different grid discretizations).

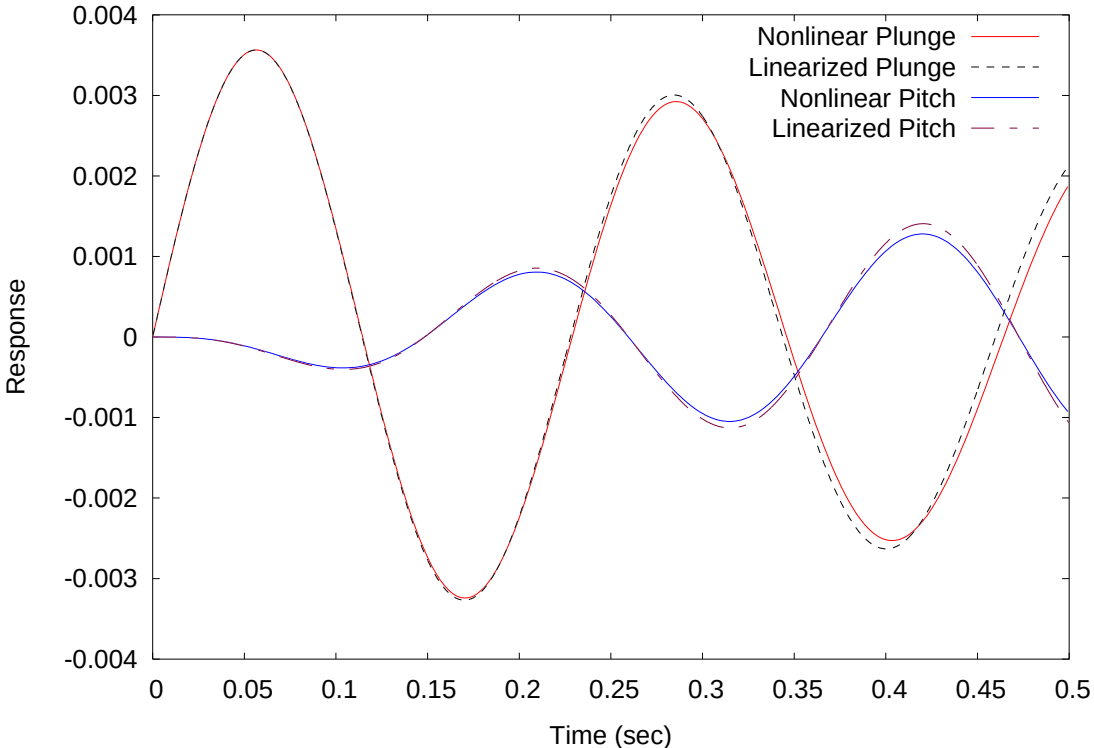


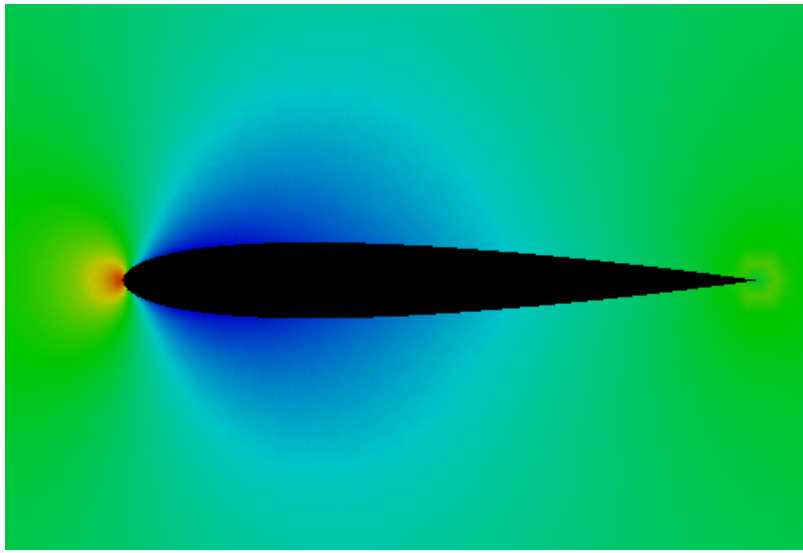


NACA 0012 Linearization Results

- The Linearized model produced results almost identical to the nonlinear model
  - 1% RMS error over 1 period

Plunge and pitch time response

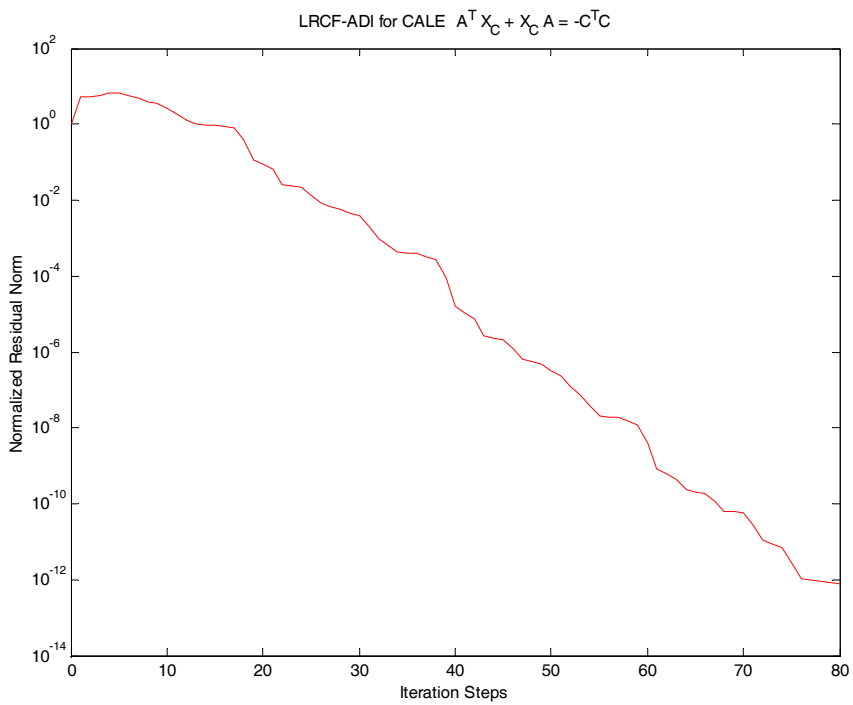


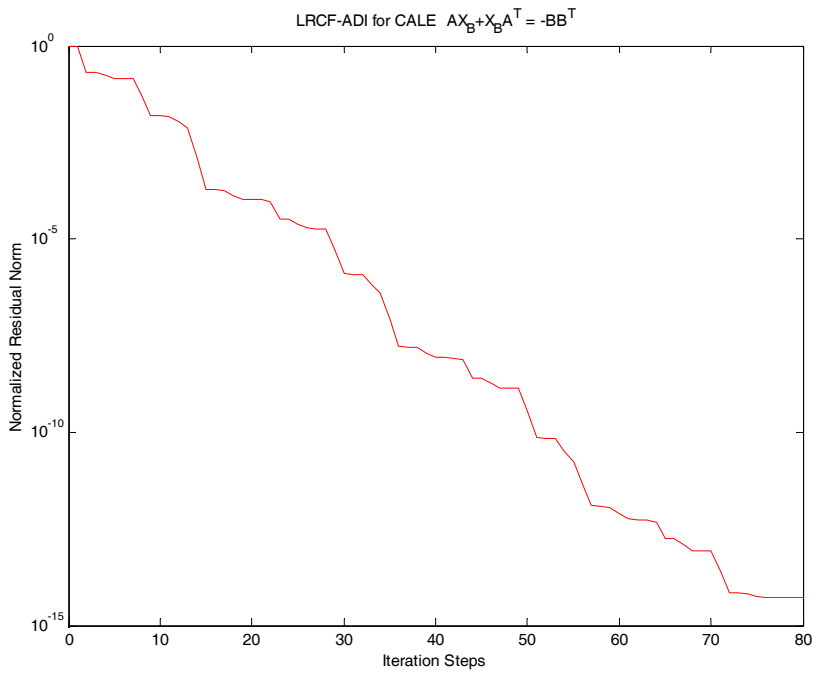


Pressure around airfoil, linearized model

2-D Airfoil-ROM

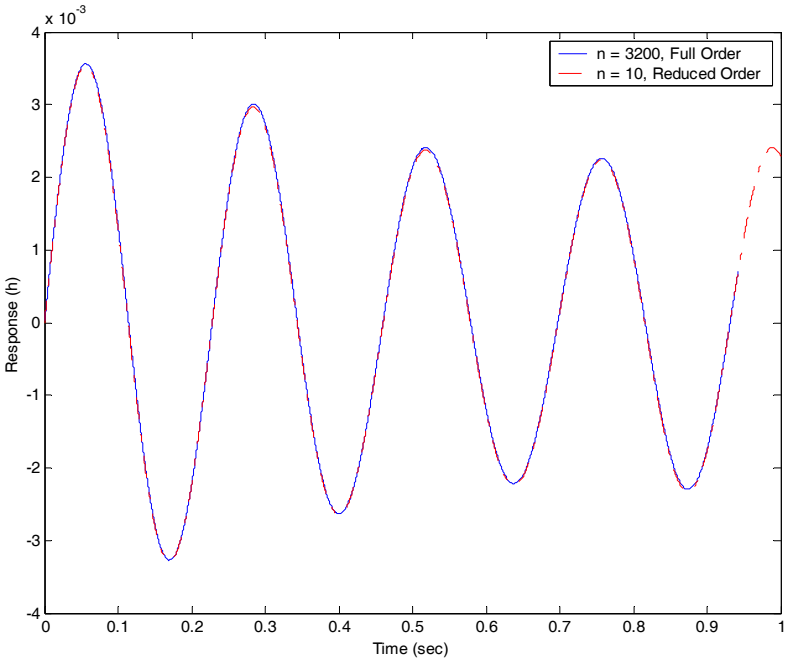
- The LRCF-ADI method was applied to the linearized 2D NACA0012 airfoil with a trailing edge flap. The performance of the algorithm on this problem was excellent, with a convergence history shown below:

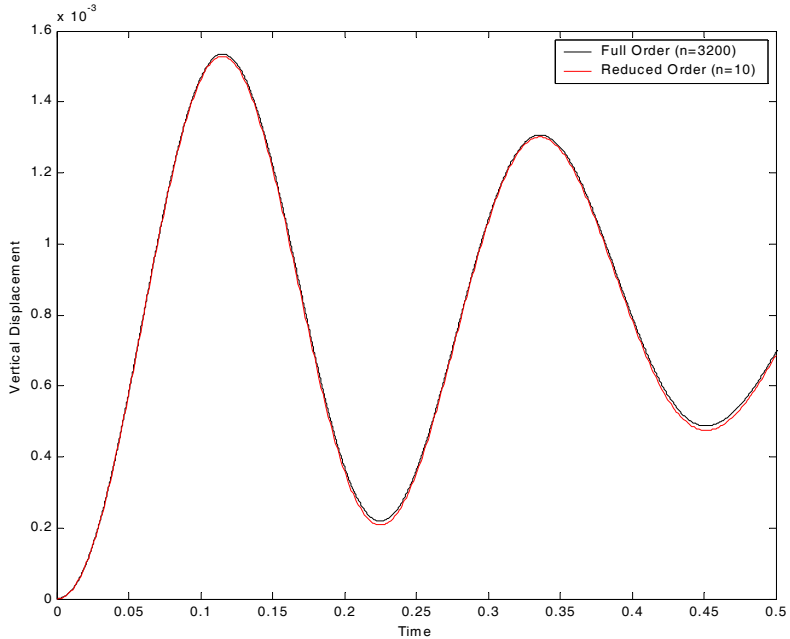




2-D Airfoil-ROM

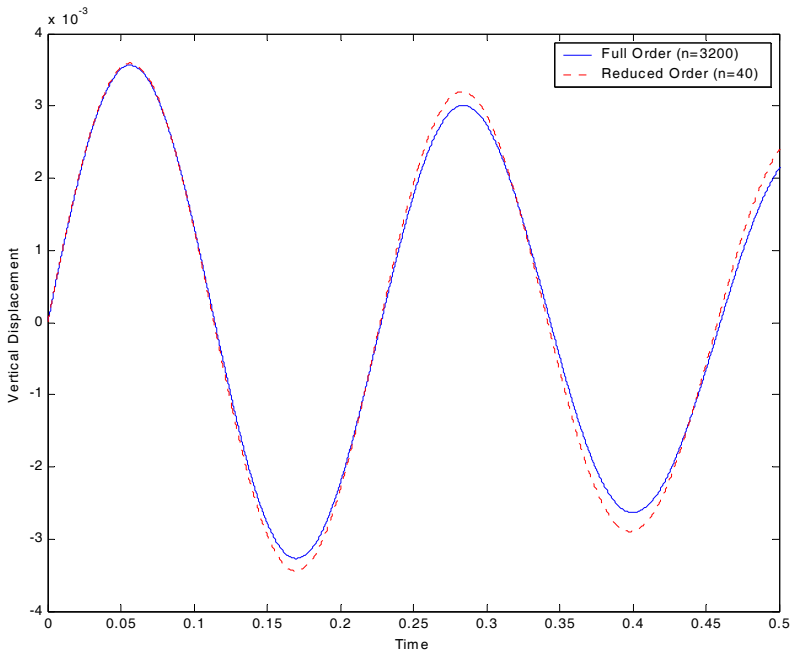
- A ROM of size 10 was created from the output of the LRCF-ADI algorithm. A comparison of this model with the full scale system is shown for an initial velocity and an initial flap deflection; RMS errors for 0.5 sec were ~1%.

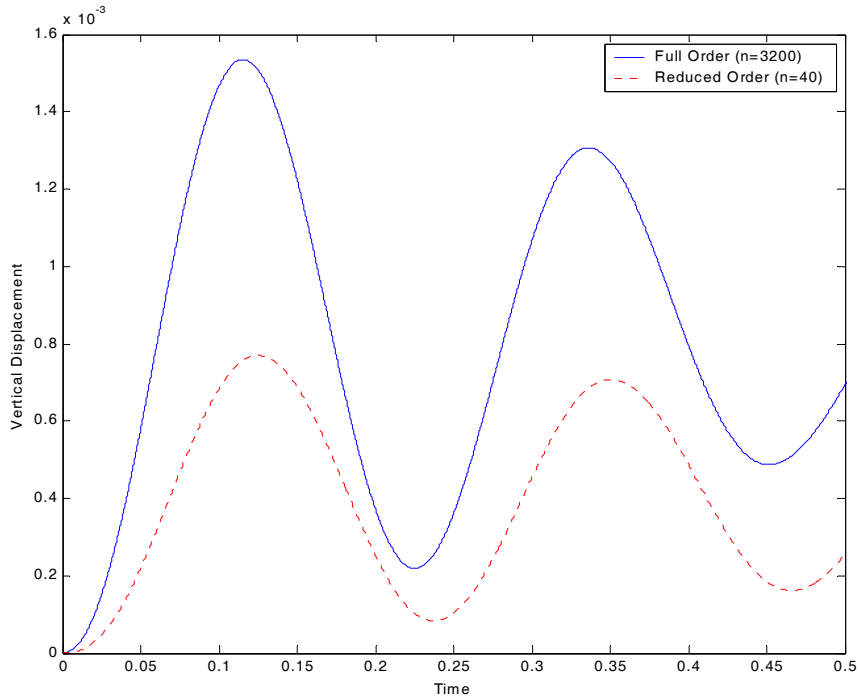




Modal vs Balanced ROM

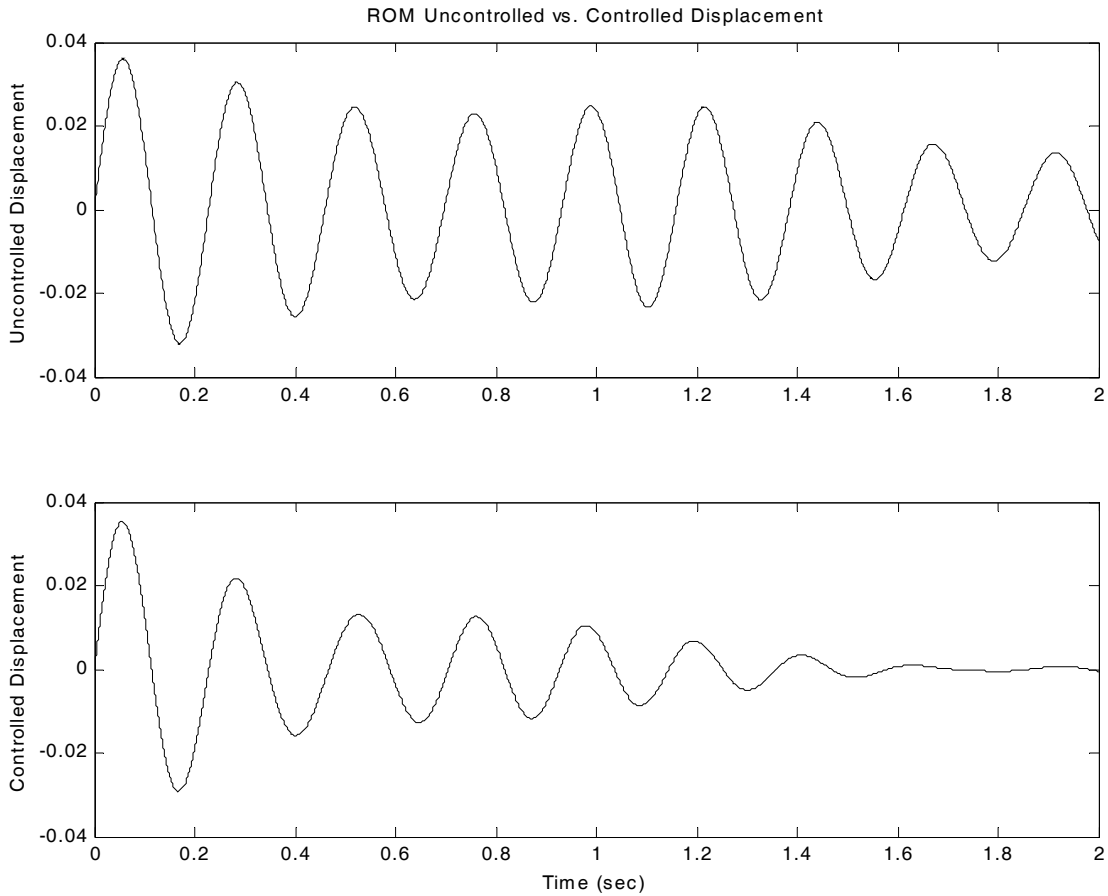
- Compare the response to a flap deflection on the previous slide to the response an n=40 state modal based ROM gives – a RMS error of ~50% for the first .5 seconds of a flap deflection (7% for an initial velocity).





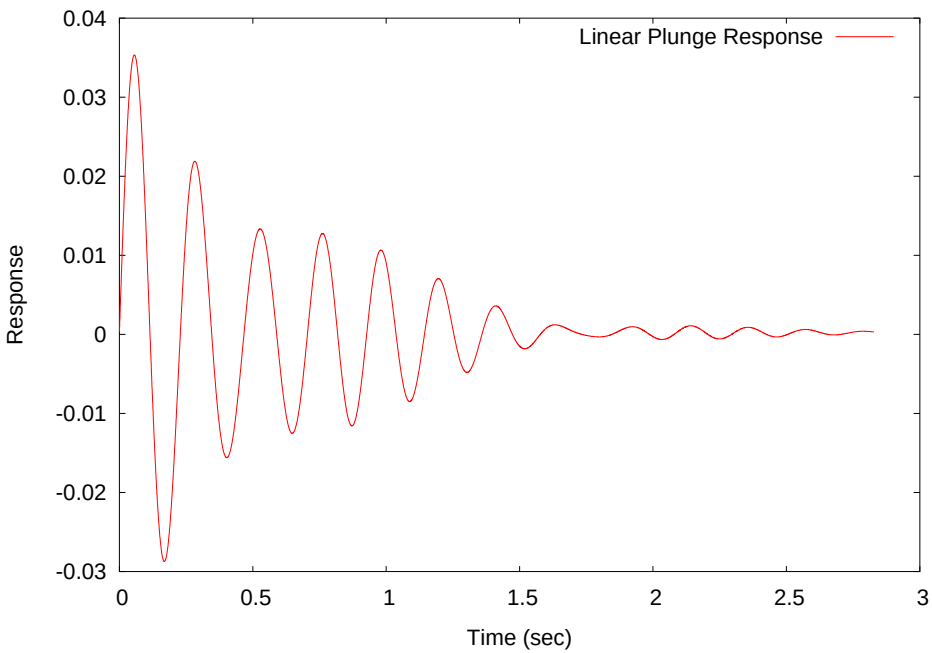
2-D Airfoil-Control

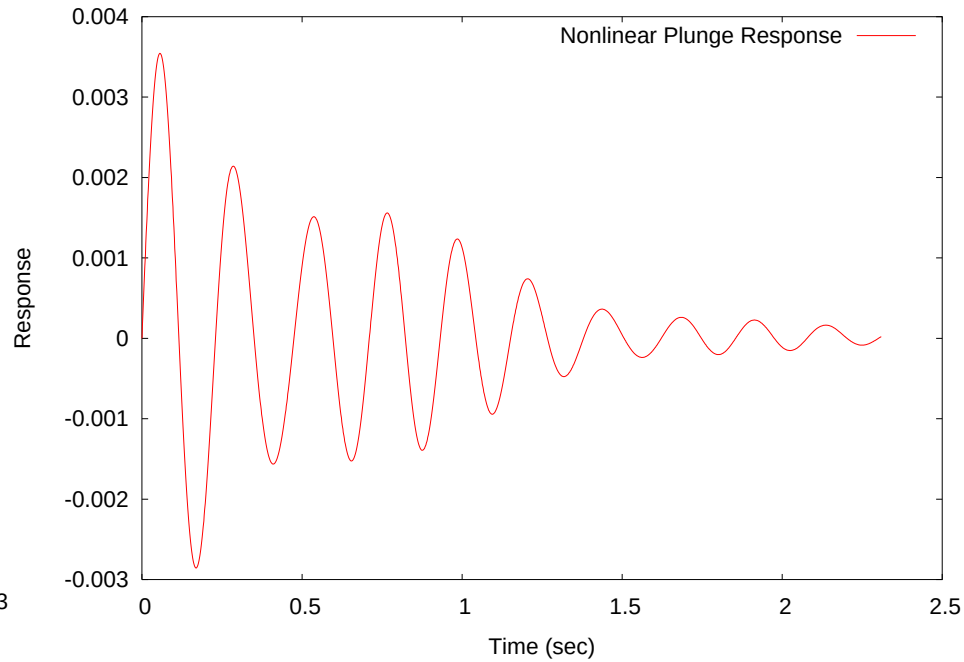
- An output feedback state estimator control law was developed for the airfoil using the 10 state balanced-based ROM. The following plots show the uncontrolled model (top) and controlled system (bottom).



2-D Airfoil/Flap-Control

- The same control law was applied to the full scale linear and nonlinear models, with the linearized on the left and the nonlinear on the right.





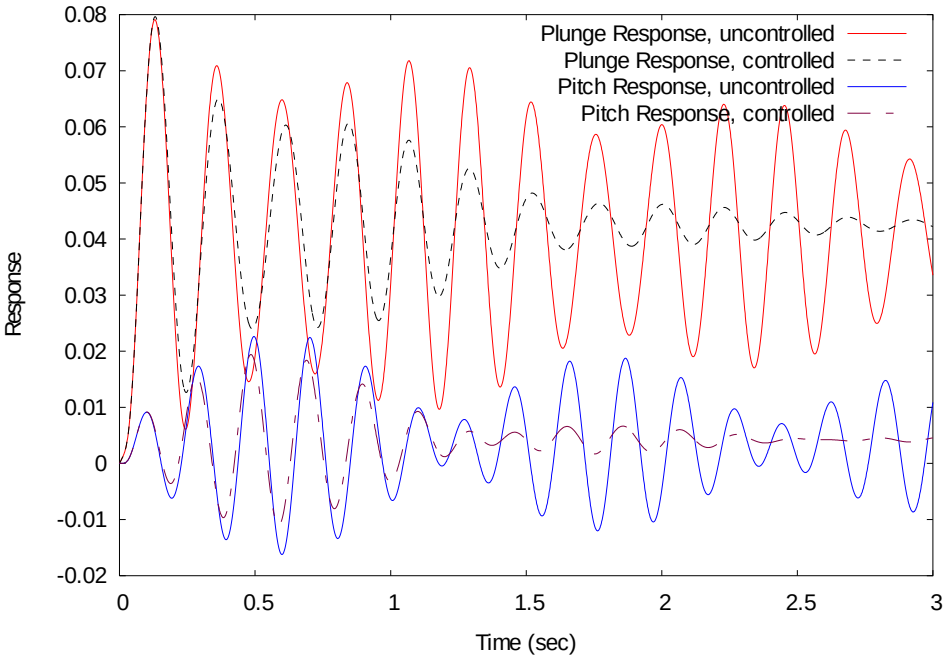
Animations: No Control Controlled

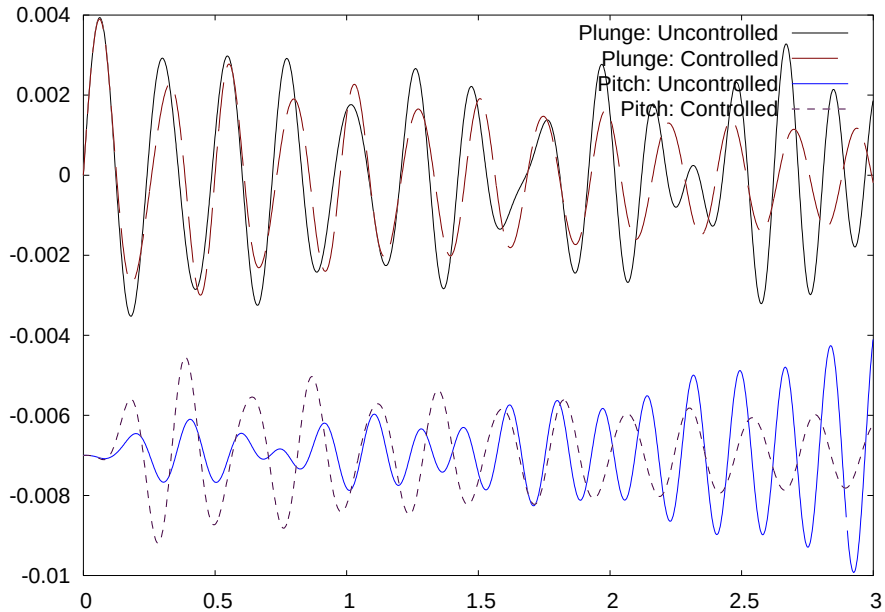
2-D Airfoil/Flap Robustness

- To test the range of applicability of the controlled system, the nonlinear simulation with the controller was used to test the following conditions:
  - Grid dependence: The controller was applied to 800, 3000, and 12,000 node models (all unstructured grids).
  - Flight speed: tested at Mach 0.5, 0.75, 0.9, and 0.95.
  - Angle of Attack: tested at 0, 5, 10, and 15 degrees (stall limit).
- This was all done using the same controller, built around a linearized model at Mach 0.5 and 0 angle of attack.
  - Creating multiple models at different linearization points and using gain scheduling or some other adaptive technique would produce better results

2-D Airfoil/Flap Robustness

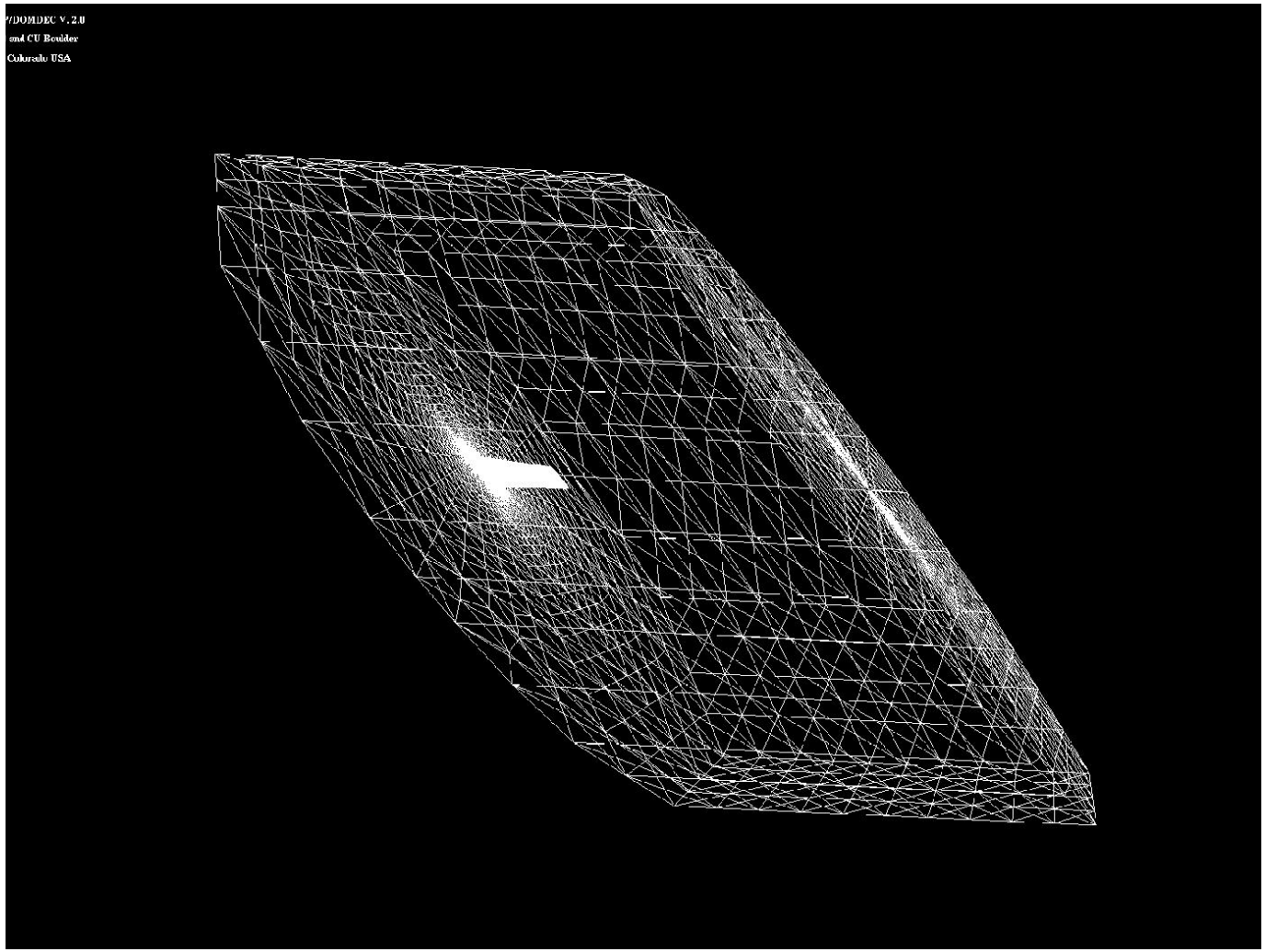
- Example results:
  - Left: 15 degree angle of attack (near-stall)
  - Right: Operation at Mach 0.9 (Flutter conditions)





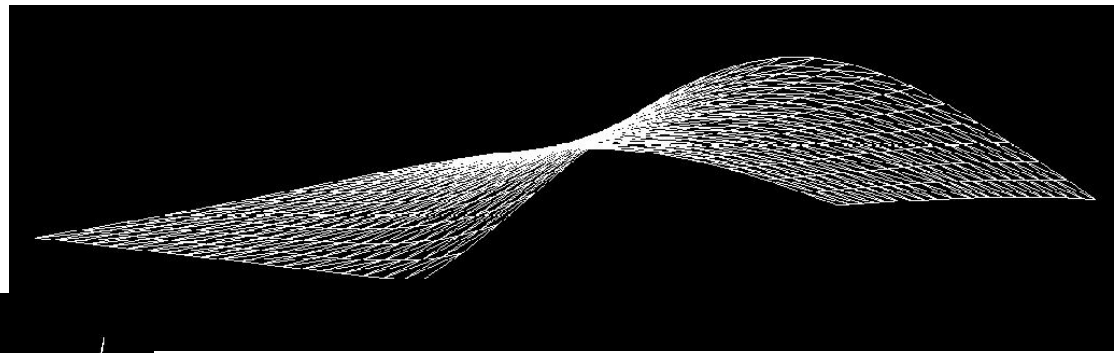
3-D: AGARD Wing

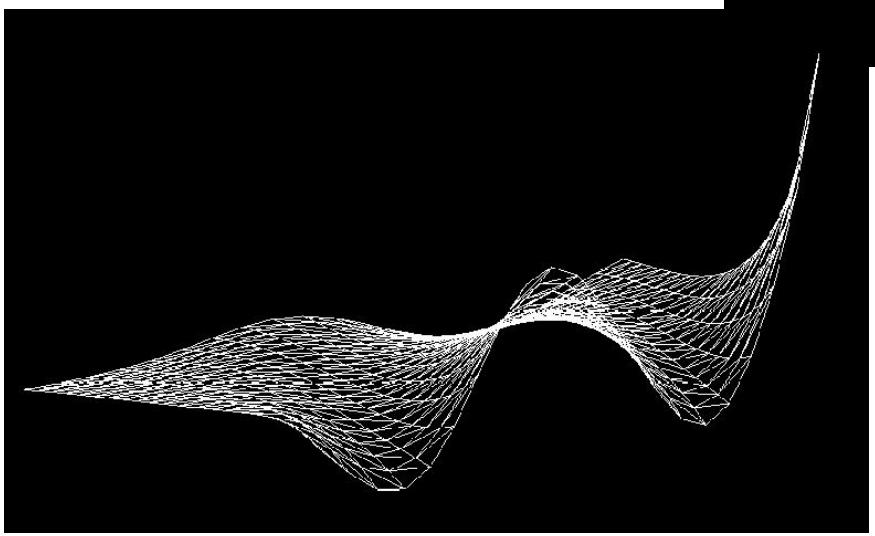
- This systems consists of a flexible Agard wing in 3-D Euler flow with ~ 110,000 DOF's.



AGARD Wing Continued

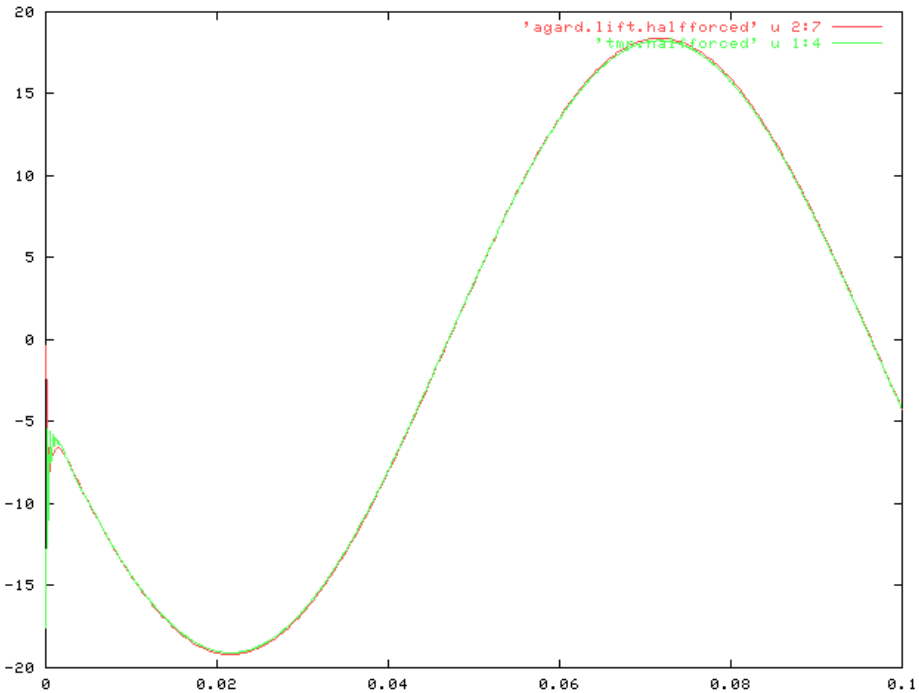
- The model is based on a parallel non-linear aeroelastic simulation. The the first 8 structural modes were used as a basis for developing a parallel linearized model.

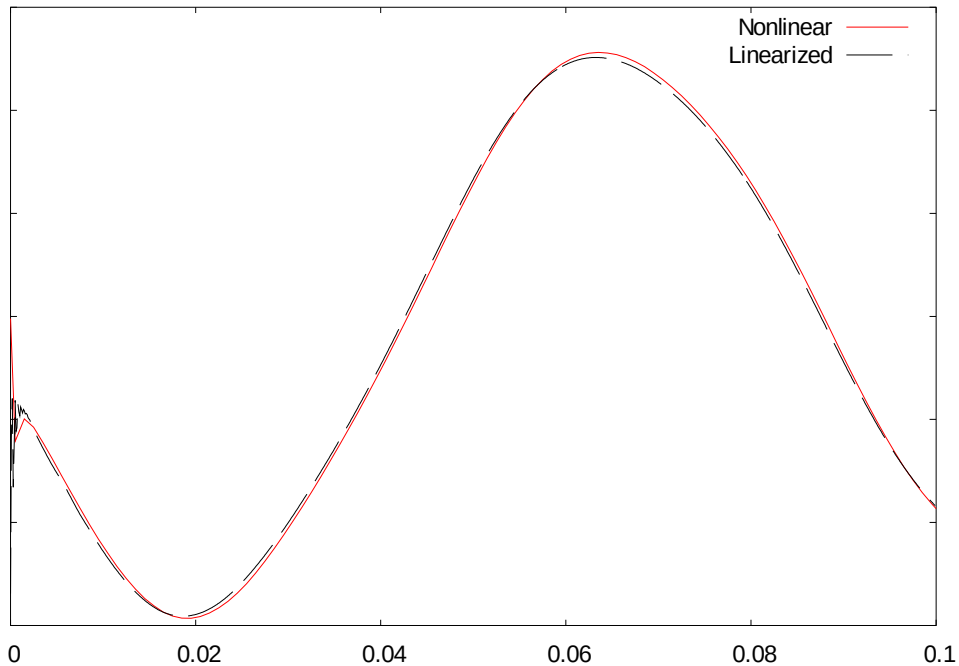




AGARD Linearization Results

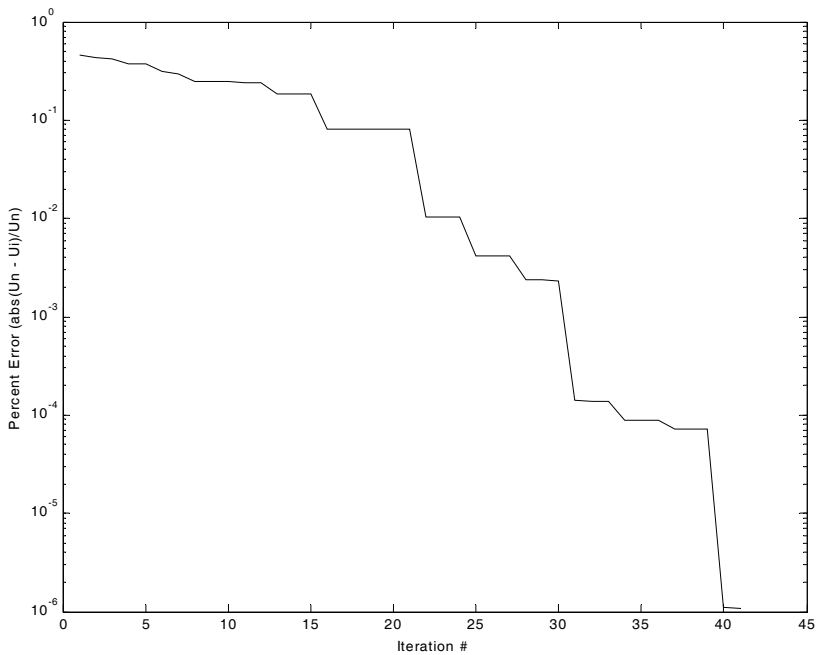
- The lift response to a forced oscillation (left) and an initial velocity (right) is shown for both the nonlinear and linearized AGARD models.

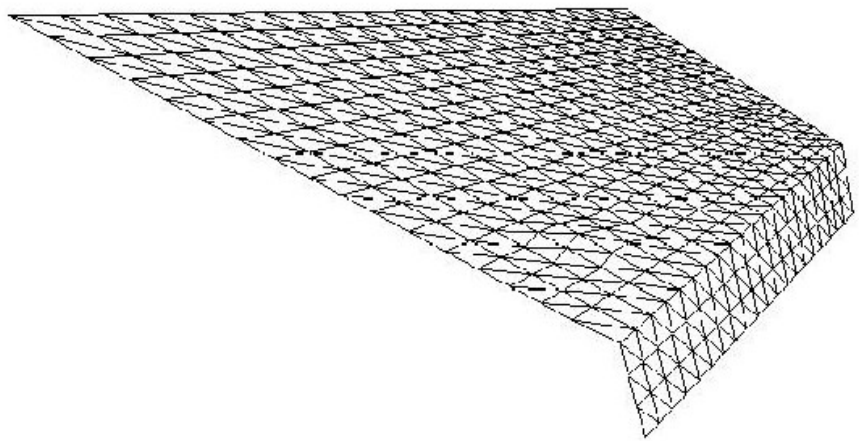




3-D Results

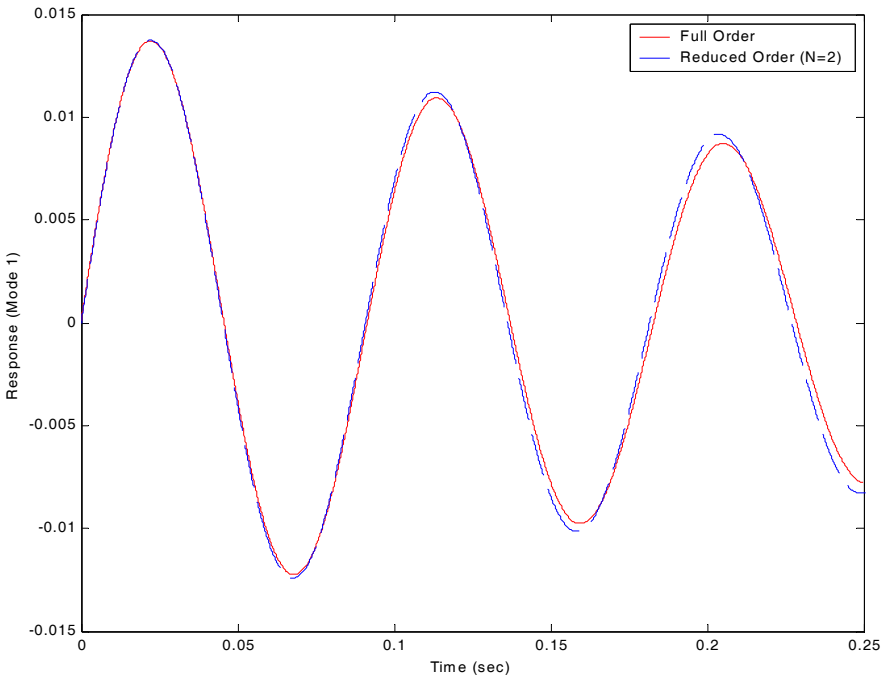
- The LRCF-ADI algorithm was implemented on the AGARD wing model with a trailing edge flap (seen below). This required developing a parallel version on the algorithm, and modifying the simulation code to allow complex and transposed operations.
- The relative convergence of the largest eigenvalue of WcWo was rapid.

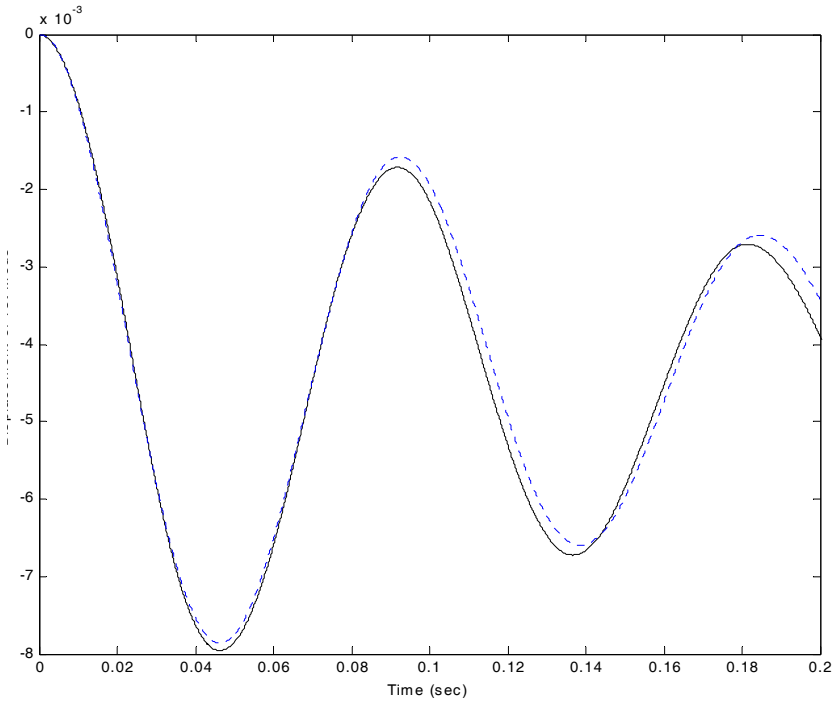




3-D Results

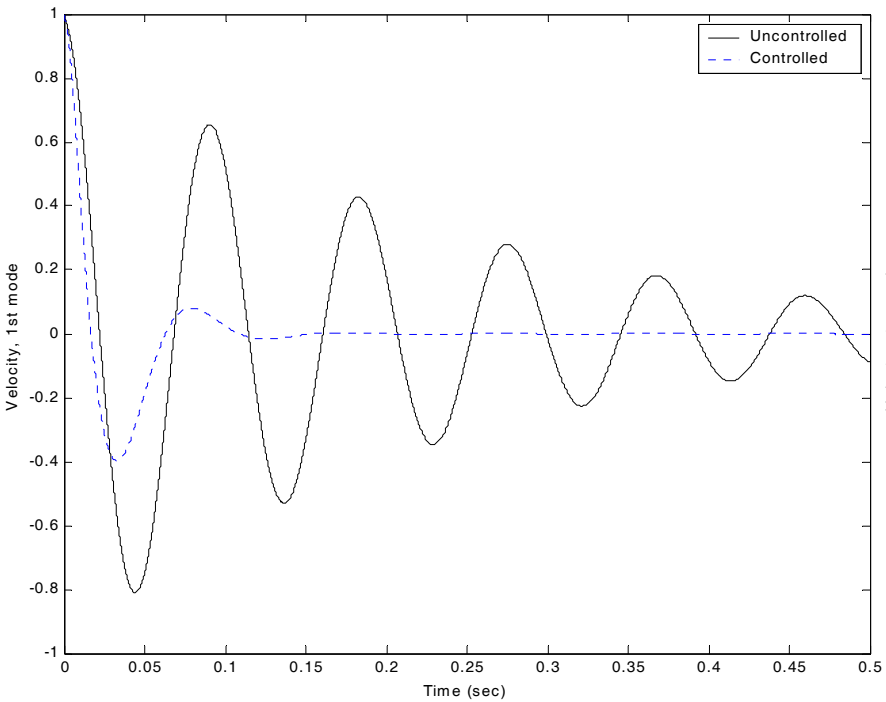
- The results from the LRCF-ADI algorithm were used to construct a 2 state ROM, which is compared to the full order model below for an initial modal velocity input. The RMS error for the first 0.1 sec was 4.6%.
- A 13 state ROM of the wing with a flap gives a 1.6% RMS error for the response to a flap deflection (shown on the left)
- Operation count: 218 linear solves, 70 mat-vec multiplications

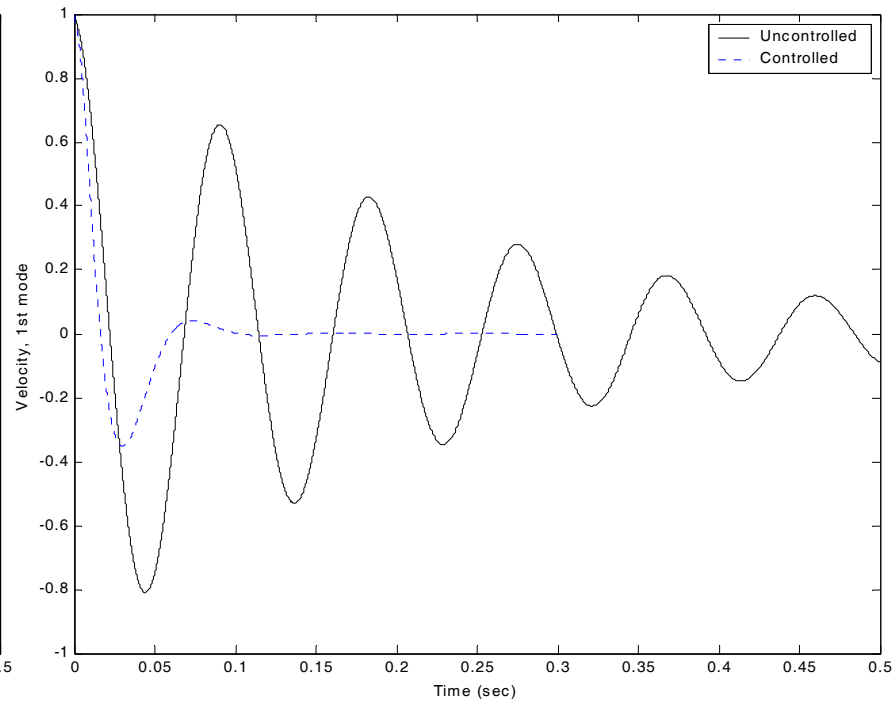




3-D Control

- An output feedback state estimator controller was constructed based on the reduced order model to stabilize the system. ROM (left) and FOM (right) uncontrolled and controlled responses are shown below.





Future Work

- Possible extensions to this work include:
  - Adaptive controllers based on one or several ROM's
  - Use of the ROM's for flutter prediction
  - Integration of a control law to the 3D nonlinear parallel code
  - Improvements to the parallel LRCF-ADI algorithm (error tracking)