JOURNAL OF AIRCRAFT
Vol. 40, No. 2, March–April 2003

Reduced-Order Modeling of an Elastic Panel in Transonic Flow

David J. Lucia[*]
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433-7765
Philip S. Beran[†]
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433-7531
and
Paul I. King[‡]
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433-7765

Reduced-order modeling is applied to a transonic aeroelastic panel that experiences oscillatory motions of a normal shock. Two-dimensional, transonic inviscid flow over an elastic panel produces transonic limit-cycle oscillations over a range of panel parameters. Proper orthogonal decomposition, in concert with domain decomposition, is shown to produce an accurate reduced-order model for the coupled aeroelastic system. Panel flutter in the transonic regime results in a large streamwise movement of a transonic normal shock across the panel surface. The accuracy and order reduction of the reduced-order model is quantified. In addition, the computational savings for this implementation is documented, and the robustness of the reduced-order model to changes in a panel parameter is explored.

Nomenclature

C	=	constraint type
D	=	plate stiffness, $E_s h^3/12(1-\nu^2)$
$\boldsymbol{E}, \boldsymbol{F}$	=	vector of X-axis and Y-axis fluxes
E_s	=	Young's modulus
E_T	=	total energy, $E_{Td}/\rho_\infty u_\infty^2$
F	=	modified flowfield expression for time-accurate Newton's iterations
F_s	=	function evaluation for Newton's iterations
G	=	vector valued function for overlapping domains
h	=	panel thickness
$\hat{\boldsymbol{i}}$	=	unit vector, x direction
$\hat{\boldsymbol{j}}$	=	unit vector, y direction
L	=	reference length
M	=	Mach number; number of degrees of freedom (DOF) for reduced-order model
N	=	number of DOFs for full system
P	=	pressure, $P_d/\rho_\infty u_\infty^2$
$\boldsymbol{P}$	=	array of pressure differentials on panel
Q	=	number of snapshots used in proper orthogonal decomposition/reduced-order model
R	=	flux calculation from the Euler equations
S	=	matrix of flowfield data, or snapshots
$\boldsymbol{S}$	=	vectors of s values on panel
s	=	panel deflection speed, $\dot{w}$
T	=	vector mapping overlapping variables
t	=	time, $t_d u_\infty/L$
$\boldsymbol{U}, \hat{\boldsymbol{U}}$	=	full- and reduced-order vector of conserved flow variables
u, v	=	fluid velocity components, u_d/u_∞ and v_d/v_∞
V	=	matrix of eigenvectors of $S^T S$
$\boldsymbol{W}$	=	vectors of w values on panel
w	=	normal panel deflection, w_d/L
$\boldsymbol{w}, \hat{\boldsymbol{w}}$	=	full- and reduced-order vector for a single fluid variable
x, y	=	spatial coordinates, x_d/L and y_d/L
$\boldsymbol{Y}_s$	=	structural state array, $[\boldsymbol{S}, \boldsymbol{W}]^T$
$\boldsymbol{y}, \hat{\boldsymbol{y}}$	=	flow variables $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$, augmented with Lagrange multipliers
α	=	expression containing structural parameters
γ	=	ratio of specific heats, 1.4
Δt	=	global time step determined to preserve stability for the smallest node spacing
Λ	=	matrix of singular values of $S^T S$
λ	=	nondimensional dynamic pressure, $\rho_\infty L/\rho_s h$
λ_i	=	Lagrange multiplier for ith constraint
μ	=	mass ratio, $\rho_\infty u_\infty^2 L^3/D$
ν	=	Poisson's ratio 0.3
ρ	=	density, ρ_d/ρ_∞
ρ_s	=	structural density
Ψ	=	reduced-order mapping matrix

Subscripts

d	=	dimensional quantity
i	=	constraint index
S	=	section number within domain
s	=	structural quantity
t	=	time derivative
x, y	=	spatial derivative
λ	=	overlapping domains
0	=	base flow
∞	=	freestream quantity, dimensional

Superscripts

n	=	index to increment discrete time steps
over	=	overlapping portion of domain
ν	=	index to increment Newton iterations

Introduction

REDUCED-ORDER modeling for transonic aeroelastic problems is challenging because of unsteady movement of normal shock waves. Reduced-order methods that linearize the governing

Received 15 January 2002; revision received 9 August 2002; accepted for publication 18 August 2002. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0021-8669/03 $10.00 in correspondence with the CCC.

[*] Doctoral Candidate, Department of Aeronautics and Astronautics, Room 201, 2950 P Street, Building 640; david.lucia@wpafb.af.mil. Member AIAA.

[†] Principal Research Aerospace Engineer, Air Vehicles Directorate, Building 146, 2210 Eighth Street; philip.beran@wpafb.af.mil. Associate Fellow AIAA.

[‡] Associate Professor, ENY, Room 201, 2950 P Street, Building 640; paul.king@afit.edu. Senior Member AIAA.

equations will produce errant results if the shock moves by a significant amount. Recent work showed that reduced-order models (ROMs) based on proper orthogonal decomposition (POD) can accurately handle the moving shock case,[1–3] but a domain decomposition (DD) technique is necessary to overcome difficulties in translating discontinuities in the flowfield. The POD/ROM/DD isolates the region of the flowfield containing the shock and models the region, either at full order, or with a special POD/ROM constructed with sufficient data to allow shock motion.

In this paper, the DD approach with POD/ROM is applied to an aeroelastic panel in crossflow. A nonlinear coupling of the two-dimensional Euler equations and the von Kármán plate equation simulates the dynamics of flow over a flexible panel. The inviscid flow produces transonic limit-cycle oscillations (LCO) for certain panel parameters and freestream conditions. Panel flutter in the transonic regime further produces a transonic shock that traverses the panel surface. Previous work[4] on this aeroelastic panel configuration neglected reduced-order modeling of the transonic case because of the difficulties in handling moving shocks. The objective is to apply POD/ROM/DD to the transonic aeroelastic panel and assess the performance of the reduced-order system in accuracy, order reduction, and computational expense. The robustness of the POD/ROM/DD to changes in panel dynamic pressure λ is explored to evaluate the model's utility for design analysis.

Numerical Formulation

In this section, the governing equations for the structural and fluid dynamics are described, DD is developed, and an overview of POD is provided. Emphasis is given to the reduced-order implementation of the fluid dynamics model, which includes a development of constrained optimization required to couple adjacent domains across internal boundaries.

Structural Dynamics Equations

Two-dimensional flow over a semi-infinite, pinned panel of length L was considered. Panel dynamics were computed with von Kármán's large-deflection plate equation (see Ref. 5), which was cast in nondimensional form using aerodynamic scales L and u_∞ $(-\frac{1}{2} < x < \frac{1}{2})$:

$$\frac{\mu}{\lambda}\frac{\partial^4 w}{\partial x^4} - N_x \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial t^2} = \mu\left(\frac{1}{\gamma M_\infty^2} - P\right) \tag{1a}$$

$$N_x = \alpha \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{\partial w}{\partial x}\right)^2 \mathrm{d}x \tag{1b}$$

Panel deflections $w(x)$ and velocities $s(x) = \dot{w}(x)$ were computed as a function of x and t, given the values of the parameters μ, λ, and h/L, such that

$$\alpha = (6\mu/\lambda)(h/L)^{-2}(1 - \nu^2) \tag{2}$$

Two boundary conditions were enforced at the panel endpoints $(x = \pm\frac{1}{2})$: $w = 0$ and $\partial^2 w/\partial x^2 = 0$.

The structural dynamics equations (1a–2) were discretized and placed in first-order form to facilitate numerical integration. Spatial discretization was accomplished with second-order accurate, central differences, and the midpoint rule was used to compute the integral in the definition of N_x. A uniform distribution of grid points was assumed. A modified central-difference formula explicitly enforced the end conditions. Excluding the endpoint states, the discrete structural variables were collocated into arrays $\boldsymbol{W}$ and $\boldsymbol{S}$ and then into the structural state $\boldsymbol{Y}_s$. The values of $(1/\gamma M_\infty^2 - P)$ at panel grid points were also collected into array $\boldsymbol{P}$. With these definitions, the structural equations took the first-order form

$$\dot{\boldsymbol{Y}}_s = \begin{bmatrix} 0 & L_N \\ I & 0 \end{bmatrix} \boldsymbol{Y}_s + \begin{bmatrix} \mu \boldsymbol{P} \\ 0 \end{bmatrix} \tag{3}$$

where I is the identity matrix and L_N is a nonlinear matrix operator given by

$$L_N \equiv -\frac{\mu}{\lambda}\frac{\delta^4}{\partial x^4} + N_x \frac{\partial^2}{\partial x^2} \tag{4}$$

Here, $\partial/\partial x$ represents central-difference, spatial discretization.

Equations (3) and (4) were integrated in time using an Euler implicit method[4] that was first-order accurate in time. The Euler implicit method used the value for pressure provided by a coupled fluid model:

$$F_s(\boldsymbol{Y}) = \left(I - \Delta t \begin{bmatrix} 0 & L_N \\ I & 0 \end{bmatrix}^{n+1}\right)\boldsymbol{Y} - \boldsymbol{Y}_s^n - \Delta t \begin{bmatrix} \mu \boldsymbol{P} \\ 0 \end{bmatrix} = 0 \tag{5}$$

where $\boldsymbol{Y} = \boldsymbol{Y}_s^{n+1}$. The nonlinearity in Eq. (5) was accounted for through subiteration. $\underline{\boldsymbol{Y}}$ was iteratively computed using the chord method, with L_N evaluated about an undeflected panel state, as represented by L_N^0:

$$\left(I - \Delta t \begin{bmatrix} 0 & L_N^0 \\ I & 0 \end{bmatrix}\right)\boldsymbol{Y}^{\nu+1} - \boldsymbol{Y}^{\nu} = -F_s(\boldsymbol{Y}^{\nu}) \tag{6}$$

The matrix L_N^0 was computed and lower–upper (LU) decomposed once at the start of the time-integration procedure to increase the efficiency with which Eq. (6) was solved. The chord method converged quickly for the transonic cases considered due to the small deflections of the panel (<2% of the panel length). Four subiterates applied at each time step were generally sufficient to force F_s to be near machine zero.

A fine structural grid of 101 interior points (103 points, including the endpoints of the panel, where $w = s = 0$ was enforced) was used to discretize the panel. The large number of structural grid points reduced the effects of translating fluid values to structural nodes.

Fluid Equations

The dynamics of fluid flows are governed by the Navier–Stokes equations for viscous flow, or the Euler equations for inviscid flow. The two-dimensional Euler equations are given here in strong conservation form[6]:

$$U_t + E_x + F_y = 0 \tag{7}$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_T \end{bmatrix} \tag{8a}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ (E_T + P)u \end{bmatrix}, \qquad F = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ (E_T + P)v \end{bmatrix} \tag{8b}$$

Here, ρu is the x-direction momentum, ρv is y-direction momentum, and E_T is total energy per unit volume.

The solution of the flowfield was approximated using a finite volume approach. It required the integral form of the Euler equations shown here[6]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V U \,\mathrm{d}\mathcal{V} + \int_{\partial V} (E\hat{i} + F\hat{j}) \cdot \mathrm{d}\bar{\mathcal{S}} = 0 \tag{9}$$

Solutions of the full system were obtained by numerically solving the integral Euler equations for small volumes within the flowfield. Spatial discretization results in a computational mesh with many small volumes, or cells. Stability of the numerical scheme necessitated small cell sizes. Each fluid variable at every cell represents a separate degree of freedom (DOF).

Within each cell, the integral form of the Euler equations reduce to the following, assuming no grid deformation,

$$\frac{\mathrm{d}}{\mathrm{d}t}U_{i,j} + \sum_{\text{sides}} (\boldsymbol{E}_{i,j}\hat{i} + \boldsymbol{F}_{i,j}\hat{j}) \cdot \frac{\mathrm{d}\bar{\mathcal{S}}_{i,j}}{\mathrm{d}\mathcal{A}_{i,j}} = 0 \tag{10}$$

The flux terms $\boldsymbol{E}_{i,j}$ and $\boldsymbol{F}_{i,j}$ were computed using first-order Roe averaging (see Ref. 6). Row-by-row ordering is used to collocate flow variables at each spatial cell location to a corresponding location in a column vector, producing an overall vector of flow variables $\boldsymbol{U}(t)$.

Time integration across the computational mesh is used to obtain flow solutions. This is accomplished with a first-order-accurate, forward Euler approximation. Small time steps combined with the large number of DOF make design analysis and control impractical with this type of numerical method.

Because the Euler equations are linear in the time derivative, and quasi linear in the spatial derivative,[6,7] the spatial derivatives and the time derivatives in Eq. (7) can be separated to form an evolutionary system. To accomplish this, the spatial terms are grouped to form a nonlinear operator R acting on the set of fluid variables. The fluid dynamics from Eq. (10) can then be expressed as

$$\frac{\mathrm{d}\boldsymbol{U}(t)}{\mathrm{d}t} = R[\boldsymbol{U}(t)] \tag{11}$$

Equation (11) is referred to as the full-system dynamics.

The panel dynamics were enforced using a transpiration boundary condition for $(-\frac{1}{2} < x < \frac{1}{2}, y = 0)$ (Refs. 8 and 9). The transpiration boundary condition replaces the solid wall with blowing or suction designed to mimic the effect of a solid boundary. For a static case, the no-flow-through condition at a solid wall implies that the flow momentum is tangent to the solid surface for all of the cells adjacent to the surface. Flow momentum is injected (blowing) or removed (suction) such that the flow momentum at the solid surface is turned (via vector addition) as though the solid wall were present. For dynamic cases, additional blowing is required because the wall velocity imparts additional momentum to the flow, in excess of the momentum needed to turn the flow tangent to the solid surface.[8,9] Transpiration boundary conditions are valid for small wall deflections, and they are a convenient way to model dynamic solid boundaries without moving grids. Moving grids require special treatment in the POD/ROM formulation.[10]

The finite volume fluid solver and the transpiration boundary condition were validated using both computational results and experimental data. Validation of subsonic performance was accomplished by comparison with experimental data.[11] Transonic performance was validated by comparison with results from COBALT.60, a widely used production code. Validation of the aeroelastic panel response is addressed later in the paper.

Grid Generation

The elastic panel model was inserted into the solid boundary at $-\frac{1}{2} < x < \frac{1}{2}$ and $y = 0$. A square domain of dimension $-23.668 < x < 23.668$ and $0 < y < 24.6$ was used for this research. (The panel length was one nondimensional length unit.) The large domain was intended to contain all of the flow dynamics, and the arbitrary choice of approximately 25 chord lengths in all directions mimics the domain used by other authors for the same problem.[4,12] Extending the domain to include all flow dynamics facilitates the use of characteristic boundary conditions for the outer portion of the domain. The results may not be general because the domain size impacts the panel response. The structured grid used 141 nodes along the solid surface and 116 nodes extending to the freestream. The spacing of the grid points increased geometrically from the solid wall in the normal direction. In the streamwise direction, the grid spacing was held constant at $\Delta_{\text{wall}} = 0.0125$ over the entire panel and was increased geometrically upstream and downstream of the panel position.

Fluid–Structure Coupling

The structural system from Eq. (3) was loosely coupled to the fluid system from Eq. (11) through the transpiration boundary condition. The structural solver was explicitly stepped forward in time by Δt from time level t_n to time level t_{n+1} using the pressure value on the panel from the fluid system. An extrapolated panel pressure value was used to avoid time-lagging errors that can arise in this coupling scheme.[12] $\boldsymbol{P}^n$ (the pressure value at time level n) was used to estimate $\boldsymbol{P}^{n+1}$ by

$$\boldsymbol{P}^{n+1}_{ex} = \boldsymbol{P}^n + \Delta\boldsymbol{P}^n \tag{12a}$$

$$\Delta\boldsymbol{P}^n = \boldsymbol{P}^n - \boldsymbol{P}^{n-1} \tag{12b}$$

$$\boldsymbol{P}^{n+1}_{ex} = 2\boldsymbol{P}^n - \boldsymbol{P}^{n-1} \tag{12c}$$

$\boldsymbol{P}^{n+1}_{ex}$ was used to update the panel deflection and velocity $\boldsymbol{Y}^{n+1}_s$, which were incorporated into the transpiration boundary condition. The fluid system was then time stepped to update the flowfield to $\boldsymbol{U}^{n+1}$. This sequence was repeatedly performed to integrate the coupled system and produce time-accurate results.

The fluid system dominated the computation time required to produce solutions for the coupled system. Time-step size was limited by the stability of the flow solver, and the number of DOF for the flow solver (64,400) was far larger than the structural model (202). As such, the focus of this research was to couple an ROM of the flowfield to the full-order structural model and produce an accurate aeroelastic panel response with less computational expense.

Domain Decomposition

The solution domain was divided into sections to facilitate the use of POD with the moving transonic shock. Isolation of the transonic shock was the primary goal of the DD. Because the transonic shock was always attached to the panel, the region of the flowfield directly above the panel was identified as the shock region. In addition, DD improved the computational performance when obtaining unsteady solutions from the reduced-order solver. Different time steps were employed for these different regions of the flowfield to minimize the number of solver iterations.[13,14]

The solution domain was divided into three regions to form the POD/ROM/DD. The geometric shape of the region did not influence the solver performance, and so regions formed by the intersection of rectangles were used for simplicity. The results presented were generated using the DD depicted in Fig. 1. The large outer region is denoted section I and is called the far field. Section II is the middle region whose outer boundary is defined by three line segments. These line segments are the sides of a rectangle with corners at the (x, y) pairs $(-3, 0)$, $(-3, 4)$, $(2.5, 4)$, and $(2.5, 0)$. Section II is called the near field, and it will be solved more frequently than the far field to update the internal boundary shared with section III. Section III contains the flow over the panel. Its outer boundary is a rectangle with corners at the $(x,\ y)$ pairs $(-0.7, 0)$, $(-0.7, 0.65)$, $(0.7, 0.65)$, and $(0.7, 0)$ and is the aforementioned shock region. A small shock forms on the upstream portion of the panel, and traverses over a large portion of the panel's length. This is the moving shock of interest for this analysis, and it is confined entirely to section III. No overlap is included in this decomposition. Domain overlap will be addressed later. Section I contains 9300 cell centers, section II

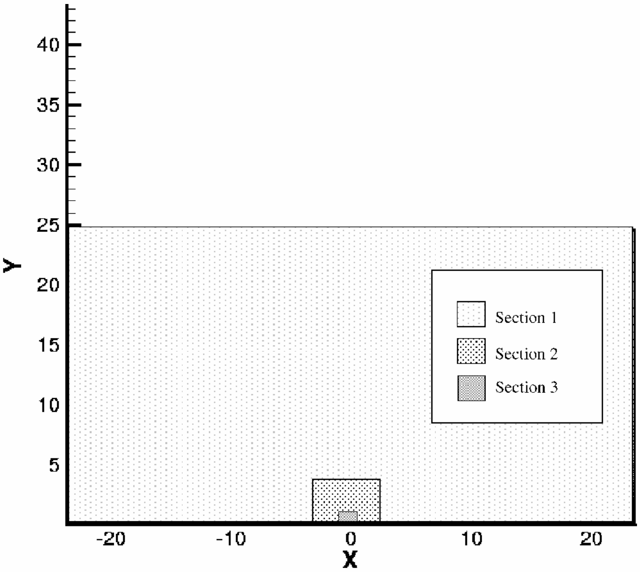

Fig. 1 DD.

contains 4886 cell centers, and section III contains the remaining 1914 cell centers.

Because the domain was divided into three sections, analysis for the POD/ROM/DD involved solving three smaller fluid problems that were linked by internal boundaries. Internal boundaries were handled with ghost cells that were filled with the corresponding values from the adjacent domain. Domain overlap was necessary under certain conditions that will be outlined later in the paper. The DD approach produced flexibility in choosing the set of snapshots and number of modes used to create the POD/ROM for each section.

Proper Orthogonal Decomposition

POD is a technique to identify a small number of DOF that adequately reproduce the behavior of the full system. The greater the reduction in DOF, the faster the computer produces a solution for each time step. Time-step size for stability can also be improved using POD because the application of POD modes exclude some of the high-frequency content that brings instability. A summary of POD as it applies to a spatially discretized flowfield follows. A detailed description of POD is available in the literature.[15,16]

For simplicity, consider only one fluid variable $w(t)$. Because the value of $w(t)$ is obtained using numerical approximation, $w(t)$ is spatially descritized using N nodes and collocated into a large array denoted $\boldsymbol{w}(t)$. For this fluid variable, the full-system dynamics in Eq. (11) are depicted hereafter,

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} = R(\boldsymbol{w}) \tag{13}$$

POD produces a linear transformation Ψ between the full-system solution $\boldsymbol{w}$, and the reduced-order solution for this fluid variable $\hat{\boldsymbol{w}}$, given by

$$\boldsymbol{w}(t) = \boldsymbol{W}_0 + \Psi\hat{\boldsymbol{w}}(t) \tag{14}$$

Note that Ψ is not time varying where $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ are functions of time t. The POD reduced-order variable $\hat{\boldsymbol{w}}(t)$ represents deviations of $\boldsymbol{w}(t)$ from a base solution $\boldsymbol{W}_0$. The subtraction of $\boldsymbol{W}_0$ will result in zero-valued boundaries for the POD modes wherever constant boundary conditions occur on the domain.

Ψ is constructed by collecting observations of the solution $\boldsymbol{w}(t) - \boldsymbol{W}_0$ at different time intervals throughout the time integration of the full-system dynamics. These observations are called snapshots and are generally collected to provide a good variety of flowfield dynamics while minimizing linear dependence. The snapshot generation procedure is sometimes referred to as the POD training period.[2]

A total of Q snapshots are collected from the full-system dynamics. These are vectors of length N. The set of snapshots describe a linear space that is used to approximate both the domain and the range of the nonlinear operator R. The linear space is defined by the span of the snapshots.[17] POD identifies a new basis for this linear space that is optimally convergent.[15] To identify the POD basis, the snapshots are compiled into a $N \times Q$ matrix S, known as the snapshot matrix. The mapping function Ψ is then developed using

$$S^T S V = V\Lambda \tag{15}$$

$$\Psi = SV \tag{16}$$

Here, Λ is the diagonal matrix of eigenvalues. To eliminate redundancy in the snapshots, the columns of V corresponding to very small eigenvalues in Λ are truncated. The matrix of eigenvalues Λ is also resized to eliminate the rows and columns corresponding to the removed eigenvalues. If $Q - M$ columns of V are truncated, the resulting reduced-order mapping Ψ will be an $N \times M$ matrix. This reduced-order mapping is a modal representation of the flowfield. The modes are the columns of Ψ that are discretized spatial functions fixed for all time. The vector $\hat{\boldsymbol{w}}(t)$ is a time dependent set of coefficients representing the coordinates of $\boldsymbol{w}(t)$ projected into the truncated linear space described by the POD basis.

The final step is to recast the governing equations to solve for the coefficients. Several methods are contained in the literature,[2] and selection of the appropriate implementation depends on the solution strategy. Both implicit and explicit time-accurate methodologies were used in this research.[1-3] Implicit time stepping was an efficient way to handle the nonshocked regions of the domain decomposed flowfield (sections I and II). Here the dynamics were benign enough that the reduced-order Jacobian required no updates. Explicit time stepping was used for the shock region because the moving discontinuity in section III was too strong a nonlinearity for the implicit scheme to capture efficiently.[3]

Explicit Implementation

Explicit integration for the full-order solver was accomplished using the first-order-accurate, forward Euler time integration:

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + \Delta t R(\boldsymbol{w}^n) \tag{17}$$

For this analysis, a non-Galerkin approach was used for simplicity. The non-Galerkin approach, also known as the subspace projection method (see Ref. 18), uses the full-system dynamics and a forward difference approximation to yield the following reduced-order flow solver:

$$\hat{\boldsymbol{w}}^{n+1} = \hat{\boldsymbol{w}}^n + \Delta t \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\hat{\boldsymbol{w}}^n) \tag{18}$$

The pseudoinverse of V is shown assuming modal truncation is employed. The inverse of Λ and pseudoinverse of V exist assuming modal truncation is employed to eliminate the zero-valued eigenvalues of $S^T S$ and their corresponding eigenvectors.[16] Notice that $\Lambda^{-1}(V^T V)^{-1} V^T S^T$ from Eq. (18) is equivalent to $(\Psi^T \Psi)^{-1}\Psi^T$.

The solution to the Euler equations in two dimensions requires simultaneous solution for four fluid variable vectors $\boldsymbol{w}(t)$. The reduced-order mapping components for each fluid variable (S and V) were collocated as blocks into larger matrices of the same name. This produced the following variant of Eq. (18), which applies to all four fluid variables simultaneously:

$$\hat{\boldsymbol{U}}^{n+1} = \hat{\boldsymbol{U}}^n + \Delta t \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\hat{\boldsymbol{U}}^n) \tag{19}$$

The subspace projection method relied on the full-system function evaluation R at each time integration step. As such, the order of each integration step was not actually reduced. Subspace projection allowed for study of POD accuracy and order reduction without the use of Galerkin projection. Also, this reduction technique can greatly increase the time-step size allowed for stability; therefore, the total number of time steps required for the explicit time-accurate solver to reach steady state can be significantly reduced.[2]

Implicit Implementation

A chord method implementation was used to obtain time-accurate solutions implicitly. The chord method is essentially Newton's method with numerically approximated Jacobians. Although the equations governing the reduced-order operator $\hat{R}(\hat{\boldsymbol{U}})$ are never explicitly obtained, the value of the reduced-order operator at any time can be obtained from Eq. (19) using the full-system function evaluation R:

$$\hat{R}(\hat{\boldsymbol{U}}) = \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\hat{\boldsymbol{U}}) \tag{20}$$

Consider the implicit time integration function F:

$$F(\boldsymbol{U}^{n+1}) = \boldsymbol{U}^{n+1} - \boldsymbol{U}^n - \Delta t R(\boldsymbol{U}^{n+1}) \tag{21}$$

The value of $\boldsymbol{U}^{n+1}$ that results in $F(\boldsymbol{U}^{n+1}) = 0$ is the solution for the flowfield at time $t_n + \Delta t$ from $\boldsymbol{U}^n$. The solution is readily obtained from Newton iterations. The Newton iterations (for subiterate from ν to $\nu + 1$) at reduced order are shown hereafter:

$$\hat{F}(\widehat{\boldsymbol{U}^{\nu}}) = \Lambda^{-1}(V^T V)^{-1} V^T S^T F(\boldsymbol{U}^{\nu}) \tag{22a}$$

$$\Delta\hat{\boldsymbol{U}} = \left[\frac{\mathrm{d}\hat{F}(\hat{\boldsymbol{U}})}{\mathrm{d}\hat{\boldsymbol{U}}}\right]^{-1} \hat{F}(\widehat{\boldsymbol{U}^{\nu}}) \tag{22b}$$

$$\hat{\boldsymbol{U}}^{\nu+1} = \hat{\boldsymbol{U}}^{\nu} + \Delta\hat{\boldsymbol{U}} \tag{22c}$$

$$\boldsymbol{U}^{\nu+1} = \boldsymbol{U}_0 + \Psi\hat{\boldsymbol{U}}^{\nu+1} \tag{22d}$$

Notice that the full-system function call is required for each Newton iteration. The reduced-order Jacobian is not updated between

Newton iterations to reduce computational cost. The reduced-order Jacobian can be quickly identified using the full-system dynamics,

$$\frac{\mathrm{d}F(\boldsymbol{U})}{\mathrm{d}\hat{\boldsymbol{U}}} = \frac{\mathrm{d}}{\mathrm{d}\hat{\boldsymbol{U}}}[\boldsymbol{U} - \boldsymbol{U}_{\text{last}} - \Delta t R(\boldsymbol{U})] \tag{23a}$$

$$\boldsymbol{U} = \Psi\hat{\boldsymbol{U}} + \boldsymbol{U}_0 \tag{23b}$$

$$\frac{\mathrm{d}F(\boldsymbol{U})}{\mathrm{d}\hat{\boldsymbol{U}}} = \Psi - \Delta t\frac{\mathrm{d}R(\boldsymbol{U})}{\mathrm{d}\hat{\boldsymbol{U}}} \tag{23c}$$

$$\frac{\mathrm{d}\hat{F}(\hat{\boldsymbol{U}})}{\mathrm{d}\hat{\boldsymbol{U}}} = \frac{\mathrm{d}}{\mathrm{d}\hat{\boldsymbol{U}}}[\Lambda^{-1}(V^TV)^{-1}V^TS^TF(\boldsymbol{U})] \tag{24a}$$

$$\frac{\mathrm{d}\hat{F}(\hat{\boldsymbol{U}})}{\mathrm{d}\hat{\boldsymbol{U}}} = \Lambda^{-1}(V^TV)^{-1}V^TS^T\frac{\mathrm{d}F(\boldsymbol{U}))}{\mathrm{d}\hat{\boldsymbol{U}}} \tag{24b}$$

For computational purposes, it was efficient to obtain the reduced-order Jacobian numerically from Eq. (24b). The Jacobian was obtained using a central-difference approximation. The computation of $\mathrm{d}F(\boldsymbol{U})/\mathrm{d}\hat{\boldsymbol{U}}$ only required $2M$ function calls (M being the number of reduced-order variables). The reduced-order Jacobians for domain sections I and II were obtained using the initial flowfield. They did not require recalculation throughout the entire time integration.

Implicit Analysis with Overlapping Domains

Constraints must be introduced to enforce smoothness on the internal boundary between adjacent domains modeled with implicit POD/ROMs. The domain coupling provided by the use of ghost cells and the full-system dynamics is not sufficient to ensure continuity along internal boundaries for the implicit implementation. To formulate the technique, consider the case where the internal boundary between two adjacent regions overlap, such that the outer boundary of one region extends beyond the inner boundary of the adjacent region by a few cells. When computing solutions for each individual domain at reduced order, the flowfield from both domains should match in the overlapping portion of the flowfield. When this does not occur, a constraint can be included to force the flowfield sections to match in the overlap region.

Consider $\boldsymbol{U}_{S1}$ and $\boldsymbol{U}_{S2}$ as adjacent domains with a shared internal boundary Γ. The domains overlap on the internal boundary, and the sections combine to form the fluid vector $\underline{\boldsymbol{U}}$,

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_{S1} \\ \boldsymbol{U}_{S2} \end{bmatrix} \tag{25}$$

Two constraint options are considered. The first constraint C^1 can be written as

$$C^1(\boldsymbol{U}) = \sum_{k\in\Gamma} \boldsymbol{U}(k)_{S1} - \boldsymbol{U}(k)_{S2} = 0 \tag{26}$$

Here, k represents the index number of the fluid variables within the flow vector $\boldsymbol{U}_S$ that correspond to the overlapping region Γ. This constraint is similar to requiring the mean difference be zero on the boundary, and C^1 approximates the L^1 norm. Notice that the absolute value is not included. This weakens the constraint because negative errors can be canceled by positive errors of equal magnitude; however, neglecting the absolute value improves computational performance. The solver formulation will use the first and second derivatives of the constraints [refer to Eq. (34)]. When the C^1 constraints are used without the absolute values, the first and second derivatives are a constant function and zero, respectively. The C^1 constraint formulation allows the Jacobian to be precomputed before solver integration, significantly reducing computation time. If an absolute value is introduced, the first and second derivatives are both functions of the flow variables, and precomputing the Jacobian is not possible. Instead of using the absolute value on the C^1 constraint, the squared difference is constrained to zero, which is written as

$$C^2(\boldsymbol{U}) = \sum_{k\in\Gamma}[\boldsymbol{U}(k)_{S1} - \boldsymbol{U}(k)_{S2}]^2 = 0 \tag{27}$$

The C^2 constraint formulation is similar to requiring the variance to be zero on the boundary, approximating the L^2 norm. Similar to including the absolute value on the C^1 constraint, the use of the C^2 expression provides a much stronger constraint, and the computational load is increased because the first and second derivatives are functions of flow variables. The derivative expressions are much simpler with C^2 constraints than the case of absolute value on the C^1 constraints, which is why this formulation is preferred.

For the time-accurate case, a functional $\ell(\boldsymbol{U})$ is defined such that

$$\frac{\mathrm{d}\ell(\boldsymbol{U})}{\mathrm{d}\boldsymbol{U}} = F(\boldsymbol{U}) \tag{28}$$

where F comes from Eq. (21). Solving for the critical values of $\ell(\boldsymbol{U})$ is equivalent to solving $F(\boldsymbol{U}) = 0$, that is, finding the flow solution of interest.

C^1 *Constraints*

With reference to the function F from Eq. (21) and the functional $\ell(\boldsymbol{U})$ from Eq. (28), a series of C^1 constraints were introduced to enforce equality within a subset of the overlap region. Lagrange-constrained optimization minimizes $\ell(\boldsymbol{U})$ subject to the constraints through the use of Lagrange multipliers λ_i, introduced as additional DOF. One Lagrange multiplier was used for each constraint. Lagrange-constrained optimization[19] modifies $\ell(\boldsymbol{U})$ by adding the linear constraints to form the functional $Q(\boldsymbol{y})$:

$$Q(\boldsymbol{y}) \equiv \ell(\boldsymbol{U}_S) + \sum_{i=1}^{I}\lambda_i\left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right)^T T_i \tag{29}$$

The solution vector is augmented to include the λ_i,

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{U}_S \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix} \tag{30}$$

For each constraint, the summation operator in Eq. (26) was implemented by taking the dot product of a vector $\boldsymbol{T}_i$ with the vector difference $(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}})$. For some number of constraints I, a total of I vectors $\underline{\boldsymbol{U}}_i^{\text{over}}$ and $\boldsymbol{T}_i$ were required with the same dimensions as $\boldsymbol{U}_S$. For the ith constraint, $\boldsymbol{U}_i^{\text{over}}$ contained flow values from a subset of the overlapping portion of the adjacent domain section. The goal of the ith constraint was to force $\boldsymbol{U}_S$ to match the flow values in $\boldsymbol{U}_i^{\text{over}}$. These flow values were collocated to the vector locations in $\boldsymbol{U}_i^{\text{over}}$ that corresponded to the identical locations within the overlapping region of $\underline{\boldsymbol{U}}_S$. $\boldsymbol{T}_i$ contained the number 1 in each fluid variable location corresponding to the selected fluid variables from the adjacent domain, collocated into $\boldsymbol{U}_i^{\text{over}}$. Zeros were placed everywhere else in T. The dot product of T with $(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}})$ tended to zero when the flow variables in $\boldsymbol{U}_S$ matched $\boldsymbol{U}_i^{\text{over}}$. Otherwise, the dot product produced a small scalar residual. Fluid values for $\boldsymbol{U}_i^{\text{over}}$ could be chosen as required to produce a reasonable solution. Each constraint could apply to any combination of the four fluid variables, over any portion of the overlapping region.

The critical values of Q are the values of $\boldsymbol{U}_S$ and λ_i such that

$$G(\boldsymbol{y}) = \frac{\mathrm{d}Q(\boldsymbol{y})}{\mathrm{d}\boldsymbol{y}} = \begin{bmatrix} F(\boldsymbol{U}_S) + \sum_1^I \lambda_i T_i \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_1^{\text{over}}\right)^T T_1 \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_2^{\text{over}}\right)^T T_2 \\ \vdots \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_I^{\text{over}}\right)^T T_I \end{bmatrix} = [0] \tag{31}$$

As long as the constraints are linearly independent, and the fluid problem has a unique solution, the critical value will be the unique minimizing function for Q (Ref. 19). The reduced-order mapping includes the Lagrange multipliers,

$$\boldsymbol{y} = \begin{bmatrix} \Psi_S & [0] & [0] & [0] & [0] \\ [0] & 1 & 0 & 0 & 0 \\ [0] & 0 & 1 & 0 & 0 \\ [0] & 0 & 0 & \ddots & 0 \\ [0] & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{\boldsymbol{U}}_S \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix} \tag{32}$$

which is rewritten as

$$\boldsymbol{y} = \Psi_\lambda \hat{\boldsymbol{y}} \tag{33}$$

Newton iterations are used to solve the reduced-order system for $\hat{\boldsymbol{y}}$. The flowfield is obtained by expanding $\hat{\boldsymbol{U}}_S$ with reduced-order mappings for each section, after which the λ_i are discarded. The full-order Jacobian $\mathrm{d}G(\boldsymbol{y})/\mathrm{d}\boldsymbol{y}$, shown hereafter,

$$\begin{bmatrix} \dfrac{\mathrm{d}F(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S} + \displaystyle\sum_1^I \lambda_i \dfrac{\mathrm{d}^2 C_i(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S^2} & \dfrac{\mathrm{d}C_1(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S} & \cdots & \dfrac{\mathrm{d}C_I(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S} \\ \dfrac{\mathrm{d}C_1(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\mathrm{d}C_I(\boldsymbol{U}_S)}{\mathrm{d}\boldsymbol{U}_S} & 0 & \cdots & 0 \end{bmatrix} \tag{34}$$

is used to obtain the reduced-order Jacobian, as follows:

$$\frac{\mathrm{d}\hat{G}(\hat{\boldsymbol{y}})}{\mathrm{d}\hat{\boldsymbol{y}}} = \left(\Psi_\lambda^T \Psi_\lambda\right)^{-1} \Psi_\lambda^T \frac{\mathrm{d}G(\boldsymbol{y})}{\mathrm{d}\hat{\boldsymbol{y}}} \tag{35}$$

For computational reasons, this formulation [similar to Eq. (24b)] is used to identify the reduced-order Jacobian from the full system. For C^1 constraints, $\mathrm{d}G(\boldsymbol{y})/\mathrm{d}\hat{\boldsymbol{y}} = [\mathrm{d}G(\boldsymbol{y})/\mathrm{d}\boldsymbol{y}]\Psi_\lambda$ is as follows:

$$\frac{\mathrm{d}G(\boldsymbol{y})}{\mathrm{d}\hat{\boldsymbol{y}}} = \begin{bmatrix} \dfrac{\mathrm{d}F(\boldsymbol{U}_S)}{\mathrm{d}\hat{\boldsymbol{U}}_S} & T_1 & T_2 & \cdots & T_I \\ T_1^T \Psi_S & 0 & 0 & \cdots & 0 \\ T_2^T \Psi_S & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_I^T \Psi_S & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{36}$$

where $\mathrm{d}F(\boldsymbol{U}_S)/\mathrm{d}\hat{\boldsymbol{U}}_S$ is obtained from $\mathrm{d}R(\boldsymbol{U}_S)/\mathrm{d}\hat{\boldsymbol{U}}_S$ by using Eq. (23c). Notice that this expression is independent of the solution vector $\boldsymbol{y}$. It can be precomputed and does not require updates during the time integration.

C^2 *Constraints*

A modification to the development is required to use C^2 constraints. This modification is presented hereafter. Again, a series of constraints are considered. The ith constraint becomes

$$C_i^2 = \left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right) \cdot \left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right) = 0 \tag{37}$$

and the Lagrangian is minimized as follows:

$$Q(\boldsymbol{y}) = \ell(\boldsymbol{U}_S) + \sum_{i=1}^I \lambda_i \left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right)^T \left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right) \tag{38}$$

$$G(\underline{y}) = \frac{\mathrm{d}Q(\boldsymbol{y})}{\mathrm{d}\boldsymbol{y}} = [0] \tag{39}$$

$$G(\boldsymbol{y}) = \begin{bmatrix} F(\boldsymbol{U}_S) + 2\displaystyle\sum_1^I \lambda_i \left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right) \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_1^{\text{over}}\right)^T \left(\boldsymbol{U}_S - \boldsymbol{U}_1^{\text{over}}\right) \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_2^{\text{over}}\right)^T \left(\boldsymbol{U}_S - \boldsymbol{U}_2^{\text{over}}\right) \\ \vdots \\ \left(\boldsymbol{U}_S - \boldsymbol{U}_I^{\text{over}}\right)^T \left(\boldsymbol{U}_S - \boldsymbol{U}_I^{\text{over}}\right) \end{bmatrix} \tag{40}$$

The Jacobian for the Newton iterations comes from Eq. (35), where $\mathrm{d}G(\boldsymbol{y})/\mathrm{d}\hat{\boldsymbol{y}}$ is found to be

$$\begin{bmatrix} \dfrac{\mathrm{d}F(\boldsymbol{U}_S)}{\mathrm{d}\hat{\boldsymbol{U}}_S} + 2\displaystyle\sum_{i=1}^I \lambda_i \Psi_i^{\text{over}} & \cdots & 2\left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 2\left(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}}\right)^T \Psi_S & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{41}$$

The Ψ_i^{over} term is simply the matrix Ψ_S overwritten with zero for any mode value not pertaining to the subset of the overlap region and the fluid variables constrained by the ith constraint. Unfortunately, this implementation requires that the Jacobian be reformed for each Newton iteration because $\mathrm{d}G(\boldsymbol{y})/\mathrm{d}\hat{\boldsymbol{y}}$ is now a function of the unknowns $\boldsymbol{U}_S$ and λ_i. Even though $\mathrm{d}F(\boldsymbol{U}_S)/\mathrm{d}\hat{\boldsymbol{U}}_S$ will not need to be recomputed, the additional multiplies within the $2(\boldsymbol{U}_S - \boldsymbol{U}_i^{\text{over}})^T \Psi_S$ term and the

$$2\sum_{i=1}^I \lambda_i \Psi_i^{\text{over}}$$

term will make this implementation less efficient.

Results

The full-order, transonic LCO behavior is described first, followed by a discussion of the POD/ROM/DD solver implementations. Observations are made on the effectiveness of overlap, boundary-constraint types, and domain-specific time steps. The accuracy, order reduction, and computational savings for a variety of implementations are discussed. Finally, the robustness is explored by observing the change in solver accuracy as panel dynamic pressure is modified.

Full-System Transonic LCO

Variations in panel dynamic pressure λ constitute the parameter space of interest for the transonic panel analysis. Increased values of λ correspond to decreased panel stiffness. Panel response at Mach 0.95 across this parameter space is illustrated in Fig. 2. For comparison, Fig. 2 contains data obtained from the literature for the same problem.[4,20] The maximum panel deflection amplitude at the one-half-chord point is shown for a range of λ. For λ less than 1750, panel stiffness was sufficient to prevent time oscillatory behavior. Any initial panel deflection and velocity eventually damped to a static deflection state, with the panel deflected either upward or downward depending on the initial condition. In Fig. 2, the two static deflection branches are evident for $\lambda < 1750$. Values of λ above 1750 resulted in oscillatory panel behavior with the proper initial condition, otherwise the panel deflection settled at the downward deflected static solution. Figure 2 illustrates how the upper static branch evolves into LCO, whereas the lower branch remains static when $1750 > \lambda > 2500$. The midchord amplitude of the upward panel deflection during LCO is about 15% lower than other cases reported in the literature. The dissipation in the first-order Roe solver provides more damping of the high-frequency panel deflections than the higher-order solvers used in the archived publications.

This additional damping is the primary cause of the muted panel deflections. The muting of the upward panel deflection is not as prominent in the static case. Otherwise, the dynamic and static behavior of the aeroelastic model for both static and LCO solutions from the full-system simulation are shown to be in good agreement with results found in the literature for the same problem.[4,20] The muted panel deflections were not a concern because the transonic moving shock was the nonlinear feature for analysis. The POD/ROM will be shown to match the full system accurately, and a full-system model with higher fidelity could be inserted without loss of generality.

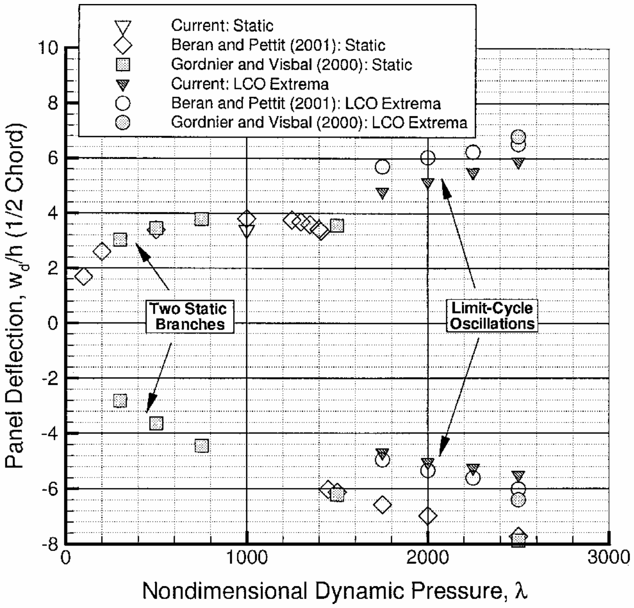

Fig. 2 Panel response with changing dynamic pressure.

With $\lambda = 2500$, the full-system simulation produced the transonic LCO shown in Figs. 3 and 4. A portion of the LCO cycle exhibits a moving shock.[21] The panel deflections corresponding to the portion of LCO containing a moving shock are shown in Fig. 3. The plot of C_P shows the shock motion progressing from Figs. 3a–3d. The dissipation in the first-order Roe's solver slightly smears the moving shock. A stronger shock also develops at the end of the panel during this portion of the LCO. Because this was a stationary shock, it did not pose a significant problem for the POD/ROM/DD.

Figure 4 shows the panel time history at the three-quarter-chord point. Transonic LCO was established very quickly from a small panel deflection in freestream conditions. The initial panel deflection was a small, downward sinusoidal panel deflection of 1×10^{-3}. The LCO was well established by 15 s, which was about one-half of the period of a single panel oscillation. This extremely rapid LCO onset makes time integration an efficient implementation for analysis of this case. Figure 4b is a phase plot of panel velocity and position at three-quarter chord. Strong nonlinearities are evident in the upswing of the panel (at point I), and again after the crest at points II and III on the downswing. These appear as loops or peaks in the phase map.

Solver Implementation

Two cases of the POD/ROM/DD were considered. The first case modeled the shock region at full order, the second case modeled it with POD/ROM. The far field (section I) and the near field (section II) used the same ROM for both cases. Both the far and near fields used an implicit, time-accurate solver, and both were modeled with POD/ROMs trained by 100 snapshots taken at evenly spaced intervals over one complete cycle of the panel LCO. Five modes per fluid variable were adequate for both sections, resulting in 20 DOF in the POD/ROM for each. The far- and near-field domains were overlapped by three cells everywhere on the common

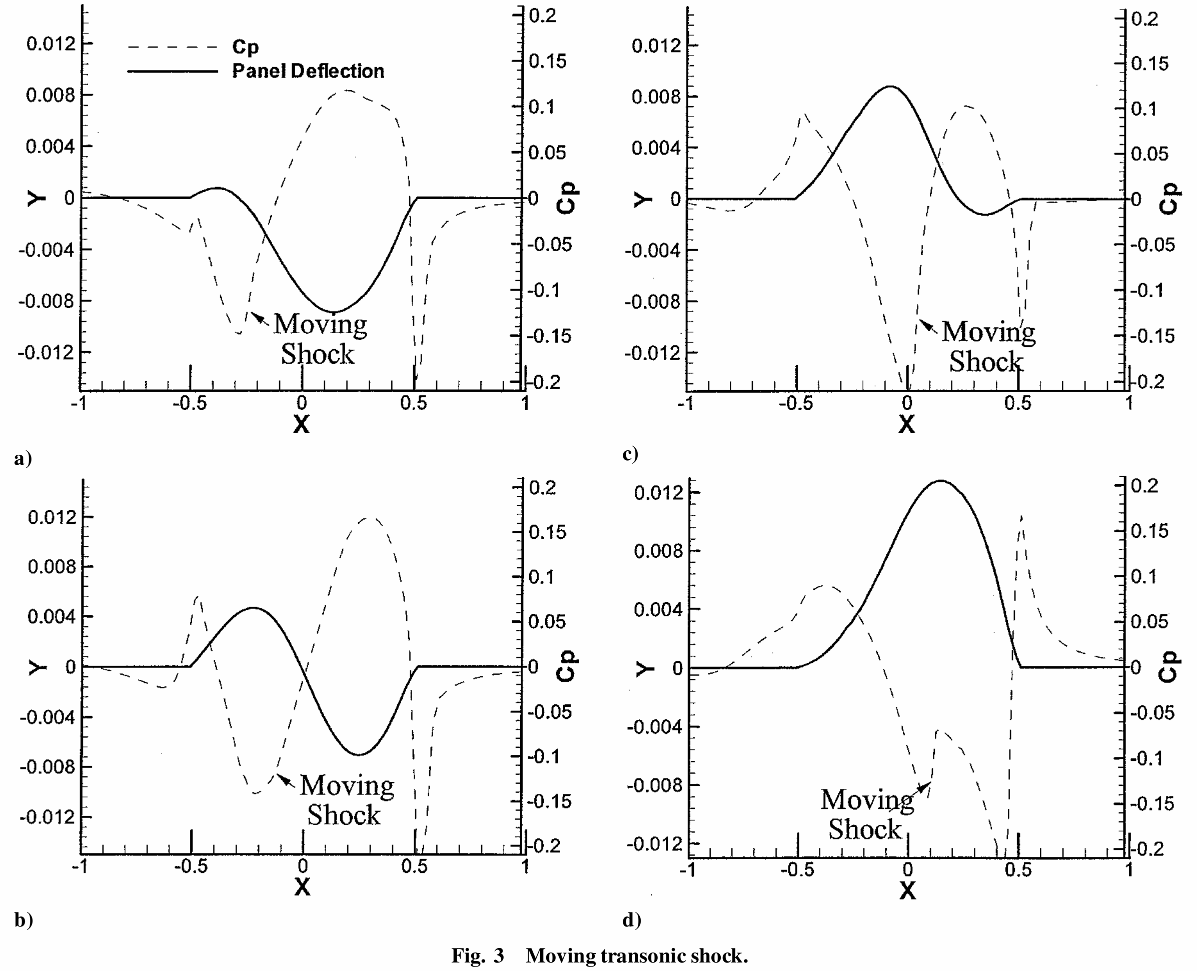

Fig. 3 Moving transonic shock.

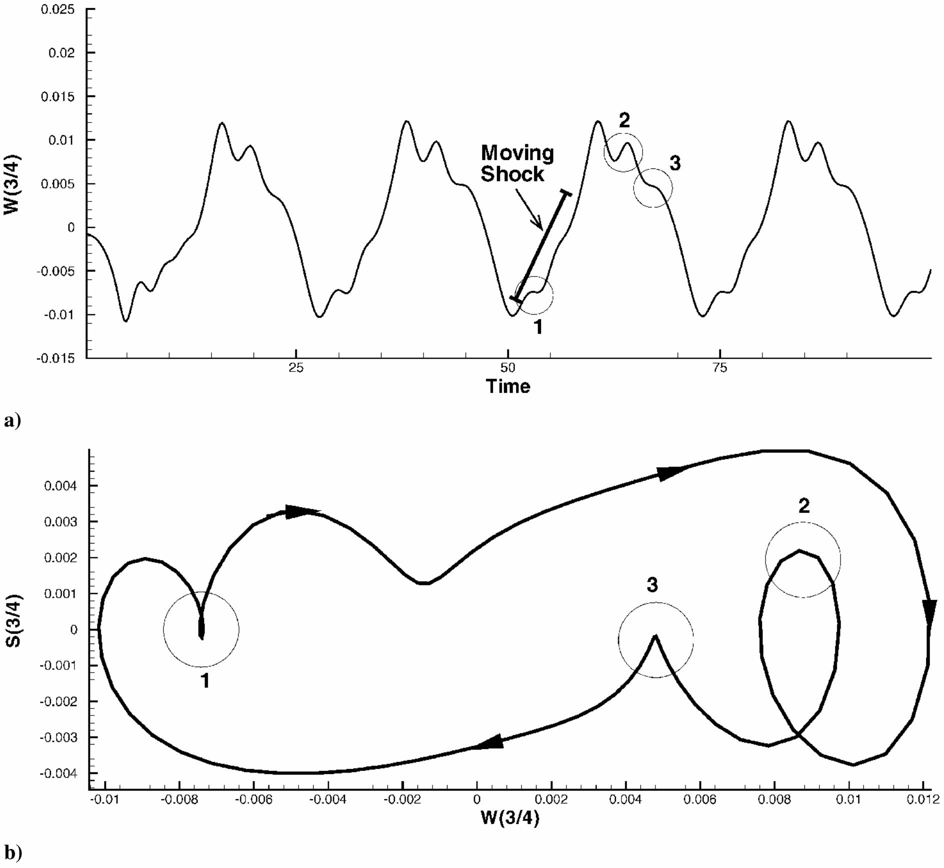

Fig. 4 LCO onset.

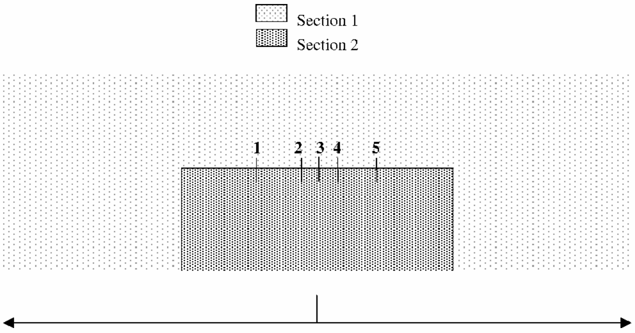

Fig. 5 C^1 constraints.

boundary. Constraints were required when the time steps were different, otherwise the near and far field were effectively modeled with a single domain using five DOF per fluid variable.

Both C^2 and C^1 constraints were implemented on the shared boundary between the far and near fields. A single C^2 constraint on density over the entire overlapping region was sufficient to produce accurate results. To use C^1 constraints, a careful selection of subsets within the overlap region was required. A series of five subsets, with one C^1 constraint for each subset, were used to couple the near- and far-field domains. Density was the constrained fluid variable. Each subset contained a vertical column of three cells, positioned on the top portion of the internal boundary as shown in Fig. 5. Each vertical strip is referred to as a "staple." The x locations for each staple were -1, -0.25, 0, 0.25, and 1. Each staple comprised a single cell in width and three cells in height to traverse the overlap region in the vertical direction.

Solver implementation produced several insights regarding the effective use of C^1 constraints. First, the size of the subsets played a critical role in the effectiveness of the C^1 constraints. Also, the position of each constrained subset was determined by the flow dynamics. In general, regions of high dynamics on the boundary required some form of constraint, whereas regions of lower dynamics required none. The careful placement of a few localized C^1 constraints could stabilize the entire boundary.

When the shock region was modeled with POD/ROM, 20 modes per fluid variable were obtained from the set of 100 snapshots described earlier. An explicit time-accurate solver was used for the shock region. No overlap was required between the near field and the shock region. An attempt to introduce one cell of overlap everywhere on the common boundary produced an instability in the full system. The location of an internal boundary in a region of strong nonlinear flow behavior can cause solver instabilities.[22] A wide variety of C^1 constraint combinations were tried. Whereas some combinations stabilized the overlapping boundary between the near field and shock region, none produced a particularly accurate flowfield. The number of constraints was limited by the number of modes used per fluid variable. When more constraints were used, the reduced-order Jacobian was not invertible.

Accuracy, Order Reduction, and Compute Time

The domain sizes and the POD/ROMs for the near and far field were not varied. As a result, efficiency was only affected by the number of domain updates for the near and far field, the time-step size for the shock region, and the type of constraint. Each solver is referred to by these key values for convenience. For example, consider the POD/ROM/DD with a Courant–Friedrichs–Lewy (CFL) of 0.9 in the full-order shock region, the near-field domain updated every 40 shock region time steps, and the far-field domain updated every 80 time steps. The CFL of 0.9 yielded a time step size of 4.6455^{-3} (nondimensional time units), and the near and far field were updated every 0.186 (which is 40×4.6455^{-3}) and 0.37 (which is 80×4.6455^{-3}), respectively. This was denoted as the full-case 0.9/0.186/0.37 model. When the full-order shock region is replaced by a POD/ROM with CFL of 2.5, the CFL of

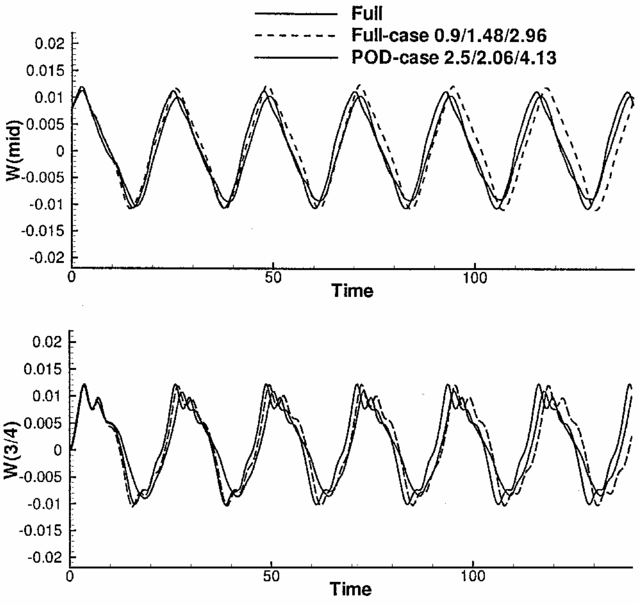

Fig. 6 Modeled panel response.

2.5 yielded a time step size of 1.29^{-2}, and the near and far field were updated every 0.52 (which is 40×1.29^{-2}) and 1.04 (which is 80×1.29^{-2}), respectively. This POD/ROM/DD is denoted as the POD-case 2.5/0.52/1.04 model. Both the full-case 0.9/0.186/0.37 model and the POD-case 2.5/0.52/1.04 model use the same POD basis for the near- and far-field domains.

The panel LCO from full-case 0.9/1.48/2.96 and POD-case 2.5/2.06/4.13 models are compared at the one-half-chord position and the three-quarter-chord position in Fig. 6. The use of either C^1 or C^2 constraints provided similar accuracy. The moving shock was replicated accurately within the shock region for both models. Regional time stepping over a range of near-field updates from 1/50 to 1/12 of one LCO period yielded similar accuracy. Models discussed in this section generally used larger time steps to improve computational performance. The CFL improvement to 2.5 was the largest allowed for stability of the POD/ROM in the shock region. Much larger CFL increases with the subspace projection method have been reported[2,4] in the literature; however, these cases did not involve a moving shock.

Although both models produced panel responses with similar accuracy, the full-case implementation produced results with less computational cost. The full-system solver required 358.08 s (wall-clock time) to compute 25 nondimensional time units on a 500-MHz Compaq workstation. This modeled one cycle of the panel LCO. In contrast, the POD case 2.5/2.06/4.13 required 133.15 s, whereas the full case 0.9/1.48/2.96 required 128.96 s. The full-order solver was more efficient in the shock region than the explicit time integration of POD/ROM. The CFL increase to 2.5 was not large enough to compensate for this inefficiency.

The use of C^1 constraints approached the efficiency of C^2 constraints when the number of far-field updates was reduced. C^2 constraints required additional multiplies to update the Jacobian for every Newton iterate. For example, the full-case 0.9/0.185/0.37 model required 67 far-field updates. The C^2 constraints were 21.4% slower for this case. When the number of far-field updates was reduced to eight for the full-case 0.9/1.48/2.96 model, the C^2 constraint case was only 5.79% less efficient. The additional multiplies with C^2 constraints were offset by a reduction in DOF (five C^1 constraints vs only one C^2 constraint), and the C^2 constraints required only two Newton iterations in comparison to four with C^1 constraints.

The full system had 64400 DOF, the full-case models had 7701 DOF (including five C^1 constraints), and the POD-case models had 125 DOF. The 88% DOF reduction for the full-case models resulted in a 66% reduction in computational cost with this solver implementation. The repeated use of the full-order function call in the Newton iterations prevented the cost reduction from reaching 88%. The additional reduction in DOF for the POD-case models did not result in additional cost savings for this solver implementation. Although the explicit integration scheme demonstrated the existence and adequacy of the POD modes for capturing a moving shock, a Galerkin-type approach should be investigated for possible computational savings in proportion to the reduction in DOF. Extrapolated to three dimensions, an Euler solver of similar fidelity in all three dimensions would yield approximately 5,300,000 DOF. A similar domain decomposition would produce full-case models with approximately 335,000 DOF. This would be a 93% reduction in DOF for the full-case models, slightly better than the 88% reduction in DOF for the two-dimensional implementation. Similarly, a slightly better than 66% reduction in computational cost could be expected from the full-case implementation in three dimensions.

Robustness

The panel phase plot in Fig. 7 demonstrates the ability of POD/ROM/DD to capture nonlinear behavior at $\lambda = 2500$, which was the value used for POD training. When POD is used in the shock region, the flowfield near the panel is constrained by the projection into the reduced-order space. The flowfield does not damp high-frequency structural responses in the same way as the full-order flowfield, resulting in the differences between the full-system and the POD-case responses on the phase plot. Reducing the number of DOF used for the structural model could improve the panel response by eliminating high-frequency structural modes. At Mach 0.95, LCO states exist for any value of λ greater than 1750 (Ref. 4). Only static panel deflections have been identified for $\lambda < 1750$. POD/ROM/DD trained with snapshots from LCO at Mach 0.95 and $\lambda = 2500$ were able to model the LCO states at Mach 0.95; however, neither POD/ROM/DD could produce a static solution for $\lambda < 1750$. The snapshots only contained data from an LCO flowfield, and the resulting modes could not be linearly combined to produce a static solution.

Next, the extendability of the POD/ROM/DD was explored. The POD basis from training at $\lambda = 2500$ (the same model used for results shown in Fig. 7) was used to model the panel response for $\lambda = 2000$. With reference to Fig. 2, this parameter variation encompasses two-thirds of the LCO parameter space. No effort was made to optimize POD training for accuracy or robustness over the parameter space. This example demonstrates the robustness of POD/ROM in the presence of moving shocks with suboptimal training.

The panel response and phase plot for $\lambda = 2000$ is shown in Fig. 8. The full-system LCO response at $\lambda = 2500$ is shown with a thick dashed line. At $\lambda = 2000$, the panel response is reduced, and the phase plot fits within the phase plot for $\lambda = 2500$.

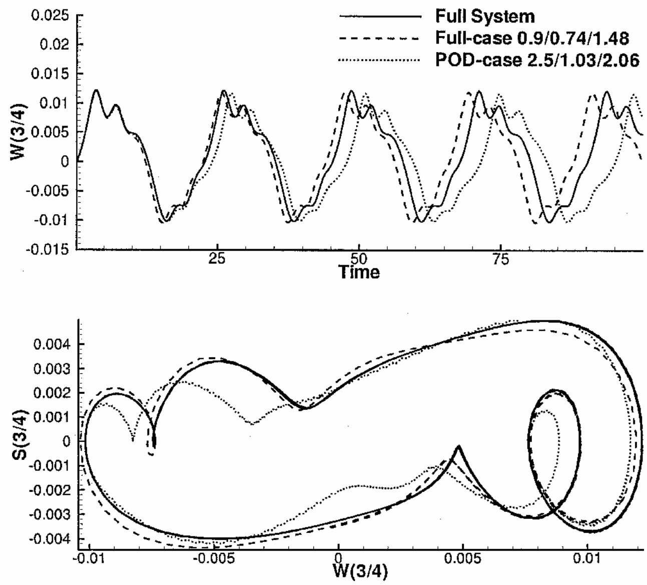

Fig. 7 LCO for $\lambda = 2500$.

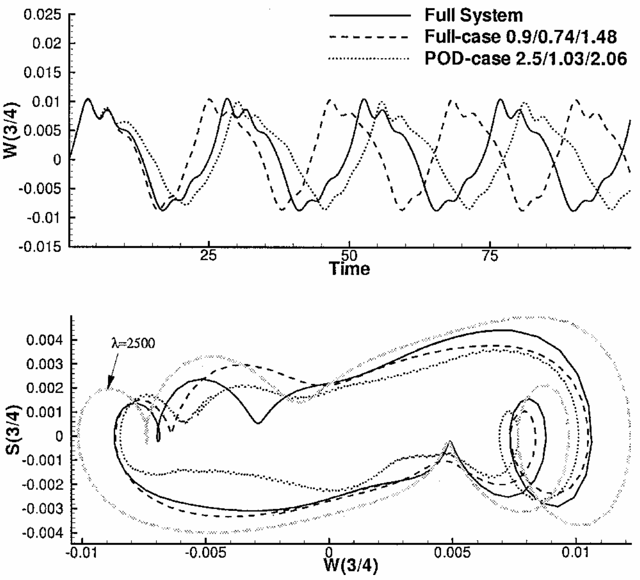

Fig. 8 LCO for λ = 2000 (trained at λ = 2500).

Both POD/ROM/DD track this trend, while preserving the nonlinear panel behavior at the new value of λ. The extendability of POD/ROM/DD across this large change in parameters using this simple, suboptimal approach demonstrates the potential for future analysis of flows with moving shocks using this implementation.

Conclusions

A reduced-order modeling approach was successfully applied to a transonic aeroelastic panel in a flowfield with large shock motion. Domain decomposition enabled the use of POD/ROM for this case. The non-Galerkin solver used a constrained optimization approach to link internal boundaries. Time-accurate results were obtained using different time steps for each domain, and the reduced-order fluid model was coupled to the panel dynamics to provide an accurate reduced-order aeroelastic system.

An 88% reduction in DOF resulted in a 66% computational saving for implementations using the full-system model in the region of the flowfield containing the shock. When POD/ROM was used in the shock region, there was an almost three orders of magnitude reduction in DOF (from 64,400 to 125). Both cases yielded accurate panel LCO and replicated correct motion of the transonic shock. In addition, both cases accurately modeled transonic LCO states corresponding to parameter values not used for POD training.

This robustness across a parameter space demonstrated the potential of POD/ROM for analysis of transonic LCO. The ultimate goal of such analysis is to characterize the LCO branches within a parameter space, especially in attempting to identify regions of supercritical LCO, where LCO states can exist at parameter levels lower than those observed in the gradual onset of LCO. At supercritical parameter values, both static and LCO solutions are possible, depending on the initial condition, with larger disturbances generally exciting the LCO state. Whereas linear analysis can be used to predict LCO onset values, it can not determine the existence of supercritical LCO states. Analysis with POD/ROM may eventually be able to identify these LCO states and allow adjustment of parameters to preclude LCO within the performance envelope of the aeroelastic system.

References

[1] Lucia, D. J., King, P. I., Beran, P. S., and Oxley, M. E., "Domain Decomposition for Reduced Order Modeling of a Flow with Moving Shocks," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. **XX–XX**.

[2] Beran, P., and Silva, W., "Reduced Order Modeling: New Approaches for Computational Physics," AIAA Paper 2001-0853, Jan. 2001.

[3] Lucia, D. J., King, P. I., and Beran, P. S., "Reduced Order Modeling of a Two Dimensional Flow with Moving Shocks," *Computers and Fluids* (to be published).

[4] Beran, P., and Pettit, C., "Prediction of Nonlinear Panel Response Using Proper Orthogonal Decomposition," AIAA Paper 2001-1292, April 2001.

[5] Selvam, R. P., and Morton, S. A., "Computation of Nonlinear Viscous Panel Flutter," AIAA Paper 98-1844, April 1998.

[6] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemishpere, Washington, DC, 1997, pp. 321–367, 388–398.

[7] Stakgold, I., *Green's Functions and Boundary Value Problems*, 2nd ed., Wiley, Washington, DC, 1998, pp. 496, 498.

[8] Pettit, C. L., and Beran, P. S., "Reduced-Order Modeling for Flutter Prediction," *41st AIAA/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA Paper 2000-1446-CP, Atlanta, GA, April 2000; also *International Journal for Numerical Methods in Engineering* (to be published).

[9] Sankar, L. N., Ruo, S. Y., and Malone, J. B., "Application of Surface Transpiration in Computational Aerodynamics," AIAA Paper 86-0511, 1986.

[10] Antonnen, J. S. R., "Techniques for Reduced Order Modeling of Aeroelastic Structures," Ph.D. Dissertation, School of Engineering and Management, Air Force Inst. of Technology, Wright–Patterson AFB, OH, Oct. 2001.

[11] Kaplan, C., "The Flow of a Compressible Fluid Past a Curved Surface," NACA Rept. 768, 1944.

[12] Hurka, J., and Ballmann, J., "Elastic Panels in Transonic Flow," AIAA Paper 2001-2722, June 2001.

[13] Lions, J., and Pironneau, O., "Non Overlapping Domain Decomposition for Evolution Operators," *Comptes Rendus de l'Academie des Sciences. Serie I. Mathematique*, Vol. 330, No. 10, 2000, pp. 943–951.

[14] Yu, H., "Solving Parabolic Problems with Different Time Steps in Different Regions in Space Based Domain Decomposition Methods," *Applied Numerical Mathematics: Transactions of IMACS*, Vol. 30, No. 4, 1999, p. 475.

[15] Holmes, P., Lumley, J., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Univ. Press, Cambridge, England, U.K., 1996, pp. 86–108.

[16] Lucia, D. J., "Reduced Order Modeling for High Speed Flows with Moving Shocks," Ph.D. Dissertation, School of Engineering and Management, Air Force Inst. of Technology, Wright–Patterson AFB, OH, Nov. 2001.

[17] Naylor, A. W., and Sell, G. R., *Linear Operator Theory in Engineering and Science*, Springer-Verlag, New York, 1982, pp. 161–165.

[18] Beran, P. S., Huttsell, L. J., Buxton, B. J., Noll, C., and Osswald, G., "Computational Aeroelastic Techniques for Viscous Flow," CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, June 1999.

[19] Strang, G., *Introduction to Applied Mathematics*, Wellesley Cambridge Press, Cambridge, MA, 1986, pp. 96–101.

[20] Gordnier, R. E., and Visbal, M. R., "Development of a Three-Dimensional Viscous Aeroelastic Solver for Nonlinear Panel Flutter," AIAA Paper 2000-2337, June 2000.

[21] Bendiksen, O. O., and Davis, G. A., "Nonlinear Traveling Wave Flutter of Panels in Transonic Flow," AIAA Paper 1995-1486, 1995.

[22] Lions, J., and Pironneau, O., "Overlapping Domain Decomposition for Evolution Operators," *Comptes Rendus de l'Academie des Sciences. Serie I. Mathematique*, Vol. 330, No. 10, 2000, pp. 937–943.