Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 188 (2003) 252–280
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Projection methods for reduced order models of compressible flows ☆

David J. Lucia *, Philip S. Beran

Air Force Research Laboratory, AFRL/VASD, 2210 Eight Street, Bldg. 146, WPAFB, OH 45433, USA

Received 17 October 2002; received in revised form 26 February 2003; accepted 28 February 2003

Abstract

Two different projection methods, Galerkin projection and direct projection, are developed for reduced-order modeling applications. The projection methods are used to identify low-dimensional systems of ordinary differential equations to represent the dynamics of a compressible, two-dimensional, inviscid flow-field under oscillatory forcing. Proper orthogonal decomposition is used to identify a small number of fluid modes to serve as the basis functions for the projections. Performance is evaluated relative to a high-order numerical model in terms of accuracy, order reduction, and computational efficiency. The treatment of boundary conditions, and stability of the reduced-order model are addressed in detail. The methods developed in this paper are suitable for general application to the Euler equations. With the addition of dissipation parameters, both the Galerkin projection and direct projection methods are tractable, stable, and properly treat the boundary conditions.
Published by Elsevier Science B.V.

Keywords: Proper orthogonal decomposition; Reduced order modeling; Galerkin projection; Compressible flow; Euler equations

1. Introduction

Galerkin projection is a widely used technique for obtaining a system of ordinary differential equations (ODEs) to describe modal amplitudes in dynamical systems. Since the introduction of spectral methods for the solution of fluid problems in the 1980s, aerodynamicists have been using Galerkin projection to obtain ODEs from the partial differential equations (PDEs) that govern fluid dynamics. However, Galerkin projection has its limitations. Boundary conditions are not explicitly accounted for using Galerkin projection, instead the basis functions used as modes for the projection must be formed to meet the boundary conditions [7,14,18]. Because spectral methods are more sensitive to errors in boundary conditions than other methods, solver implementations using Galerkin projection have been limited to flow fields with

☆ The views expressed in this article are those of the authors and do not reflect the official policy or position of the US Air Force, the Department of Defense, or the US Government.

* Corresponding author. Tel.: +1-937-255-9707; fax: +1-937-656-6321.
E-mail address: david.lucia@wpafb.af.mil (D.J. Lucia).

0021-9991/03/$ - see front matter. Published by Elsevier Science B.V.
doi:10.1016/S0021-9991(03)00166-9

either constant or oscillatory boundaries, and with simple, smooth geometries [7]. For these reasons, virtually all of the Galerkin projection applications have addressed incompressible flow. Spectral method applications for compressible flows do exist (e.g. see [15]), but they use a collocation process in lieu of Galerkin projection to allow Dirichlet boundary conditions to be satisfied explicitly [7].

Perhaps the greatest challenge preventing use of Galerkin projection on compressible flow fields is the explosion in the number of terms encountered when addressing the Euler or Navier–Stokes equations [18]. Cubic nonlinearities in the conserved flux expressions produce a series of terms containing M^3 components during the Galerkin projection process, where M is the number of basis functions in the modal expansion. Orthogonality provides no reduction in the number of terms, leaving the analyst with a daunting task of tracking, differentiating and integrating all the terms arising from the expansion. In addition, the projection cannot generally be reused since the number and type of basis functions can vary greatly between applications. However, for a projection of minimal order, Galerkin projection is a promising candidate if the challenges just outlined can be dealt with.

As an alternative approach to Galerkin projection, we introduce the term "direct projection" to describe the use of a Taylor series expansion for developing reduced-order fluid models. Since the Euler equations are linear in the time derivative, the nonlinear flux terms can be replaced using a Taylor series expansion about a base flow field. This approach is seldom used in computational fluid dynamics (CFD) because the generalized Taylor expansion involves Jacobian terms that are extremely large and expensive to compute, and other methods are generally much more efficient.

Our ultimate desire is to develop a reduced-order modeling approach suitable for design optimization of flexible air-vehicles operating in nonlinear flight regimes. Proper orthogonal decomposition (POD) of computed data became popular in the mid 1990s to reduce the number of degrees of freedom (DOFs) of flow solvers. The literature refers to such POD based fluid models as POD reduced order models (ROMs), where the term "order" is used to imply the number of DOF. POD/ROM has enjoyed a great deal of success for incompressible flow fields (e.g., see [3,19,21,31,33,34]). More recently, POD has been extended to compressible flow applications, but Galerkin projection has not been used in light of the difficulties discussed earlier (e.g., see [4,6,8–13,16,22,24,26,29,32]). The extension of POD/ROM to compressible flow has produced as much as three orders of magnitude reduction in DOFs, but the best solvers have only realized one order of magnitude in computational savings [5,24,25]. If the order reduction represents the potential in computational savings from using POD/ROM, current implementations are not efficiently exploiting the low-order behavior.

The advent of POD applied to compressible flow applications motivates the use of both Galerkin and direct projection as a means to improve computational efficiency. POD basis functions reduce the difficulties when applying Galerkin projection to the Euler or Navier–Stokes equations. First of all, POD basis functions will satisfy the boundary conditions as they are represented in the data used to form the POD basis. Secondly, a very small number of basis functions are retained in the modal expansion. This reduces the number of terms when evaluating the nonlinear fluxes. POD is essentially a spectral method, and the POD procedure identifies basis functions for the modal expansion in lieu of the Fourier series, Legendre or Chebychev polynomials typical of a conventional spectral method. Integrating the small number of ODEs produced by Galerkin projection will be very fast, providing computational savings on the order of the DOF reductions reported for compressible flow applications. Likewise, the direct-projection approach becomes computationally practical when combined with reduced-order methods. The reduced-order mapping identified using POD produces Jacobians that are very small. In addition, the generation of this reduced-order Jacobian using a high-order model is much less expensive than the high-order counterpart [5,23].

In this paper, we develop both Galerkin and direct projection for the linearized Euler equations in two-dimensions using POD basis functions. Both of the two techniques are viable, and both are compared to identify the limitations and/or advantages of either approach. Since many flows are dominated by linear

behavior, our initial application to linearized Euler is an important first step to a fully nonlinear capability, which is ultimately desired. We record the techniques that make both methods general for an arbitrary number of global fluid modes corresponding to a general fluid problem bounded by some combination of fixed, solid walls, and Dirichlet or Neumann boundary conditions. The computational cost and implementation complexity to perform the projections are greatly reduced by these techniques.

Galerkin and direct projection are demonstrated on an unsteady two-dimensional (2-D) flow over a bump. These methods will ultimately be used to generate aeroelastic models requiring moving boundaries; however, we consider stationary boundaries in this paper as a first step. Unsteadiness in this model problem is created by oscillating the ratio of specific heats over a small portion of the domain above the bump location. Representation of the boundary conditions, the addition of dissipation into the reduced-order models, and computational efficiency are addressed. The results provide fundamental insights for future use of Galerkin or direct projection in reduced-order modeling.

2. Background

This section provides background as a foundation for the detailed reduced-order modeling development that comprises the majority of this paper. Included below is an overview of fluid dynamics, POD, and the legacy, non-Galerkin approach to generating POD/ROMs for compressible flows. Some background on both Galerkin projection and direct projection is also provided.

2.1. Fluid dynamics

The dynamics of inviscid fluid flows are governed by the Euler which are given below in strong conservation form for two space dimensions [37]:

$$\frac{\partial U}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=0, \tag{1a}$$

$$U(\mathbf{x},t)=\begin{bmatrix}\rho\\ m_x\\ m_y\\ E_{\mathrm{T}}\end{bmatrix},\quad E=\begin{bmatrix}m_x\\ -\frac{m_x^2(\gamma-3)+(\gamma-1)(m_y^2-2E_{\mathrm{T}}\rho)}{2\rho}\\ \frac{m_xm_y}{\rho}\\ \frac{m_x(-m_x^2(\gamma-1)-m_y^2(\gamma-1)+2E_{\mathrm{T}}(1+\gamma)\rho)}{2\rho^2}\end{bmatrix}, \tag{1b}$$

$$F=\begin{bmatrix}m_x\\ \frac{m_xm_y}{\rho}\\ -\frac{m_y^2(\gamma-3)+m_x^2(\gamma-1)-2E_{\mathrm{T}}(\gamma-1)\rho}{2\rho}\\ \frac{m_y\left(-m_x^2(\gamma-1)-m_y^2(\gamma-1)+2E_{\mathrm{T}}\gamma\rho\right)}{2\rho^2}\end{bmatrix}. \tag{1c}$$

Here ρ is density, $m_x=\rho u$ is the x-direction momentum, $m_y=\rho v$ is y-direction momentum, and E_{T} is total energy per unit volume (u and v are the velocity components). Although the indices are not included, ρ, m_x, m_y and E_{T} are functions of space $\mathbf{x}=(x,y)$ and time t. Since we assume an ideal gas for our applications, this equation set is closed using the ideal gas law, and γ is the ratio of specific heats. Boundary conditions vary based on the characteristic properties of the flow (i.e. subsonic, supersonic or transonic), and the proximity of the boundary to the dynamics of interest. Dirichlet conditions ($U_\Omega=U(\boldsymbol{x}_\Omega)$) are generally acceptable when the boundary dynamics are minimal. Notice that density ρ appears in the

denominator for each momentum term, and ρ^2 appears in the denominator for each energy term. In the numerators, notice that there are squared nonlinearities in the momentum terms, and cubic nonlinearities in the energy term.

The solution of the Euler equations can be approximated by a wide variety of time-integration techniques. To do this, the spatial domain is discretized, and the flow variables in $U(\mathbf{x}, t)$ at each discrete location are collocated into a column vector $\mathbf{U}(t)$ and computed with finite-difference, finite-volume, or finite-element techniques. Time integration across the computational mesh is used to obtain flow solutions. Since both accuracy and stability of the numerical technique generally requires a very large computational mesh, these solvers tend to be computationally expensive.

When the spatial derivatives of the flux terms E and F are grouped to form a nonlinear operator R acting on the set of fluid variables, the fluid dynamics from Eq. (1a) can be expressed as,

$$\frac{\mathrm{d}U(\mathbf{x}, t)}{\mathrm{d}t} = R(U(\mathbf{x}, t)). \tag{2}$$

When discretized this expression takes the following form,

$$\frac{\mathrm{d}\mathbf{U}(t)}{\mathrm{d}t} = R(\mathbf{U}(t)). \tag{3}$$

Eq. (3) is referred to as the full-system dynamics.

2.2. Proper orthogonal decomposition

POD is a technique to identify a small number of basis functions that adequately describe the behavior of the full-system dynamics (Eq. (3)) across some parameter space of interest. A summary of POD as it applies to a spatially-discretized flow field follows. A detailed description of POD is available in the literature [5,18]. For simplicity, consider only one fluid variable, $w(\mathbf{x}, t)$, which when spatially discritized using N nodes is denoted $\mathbf{w}(t)$. For this fluid variable, the full-system dynamics in Eq. (3) is represented by

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = R_w(\mathbf{w}). \tag{4}$$

Spectral methods approximate the solution $w(\mathbf{x}, t)$ as follows

$$w(\mathbf{x}, t) \approx \sum_{k=1}^{M} a_k(t)\phi_k(\mathbf{x}). \tag{5}$$

When the domain is spatially discretized, $\phi_k(\mathbf{x})$ becomes a vector, $\boldsymbol{\phi}_k$, and the following relation applies:

$$\mathbf{w}(t) \approx \sum_{k=1}^{M} a_k(t)\boldsymbol{\phi}_k. \tag{6}$$

The set of vectors $\{\boldsymbol{\phi}_k\}$ are discrete basis functions corresponding to the computational mesh defined for the numerical solver. The set $\{a_k\}$ are the modal coefficients. The discrete form can be represented using matrix algebra. The fluid modes comprise columns of a modal matrix Φ, and the coefficients are collocated into a column vector $\widehat{\mathbf{w}}(t)$. By convention, we denote the vector $\widehat{\mathbf{w}}(t)$ as the reduced-order solution, and the vector $\mathbf{w}(t)$ as the full-order solution. Using this syntax, POD produces a linear transformation Φ between the full-order solution, $\mathbf{w}$, and the reduced order solution, $\widehat{\mathbf{w}}$

$$\mathbf{w}(t) = \mathbf{W}_0 + \Phi\widehat{\mathbf{w}}(t). \tag{7}$$

The reduced-order variable $\widehat{\mathbf{w}}(t)$ represents deviations of $\mathbf{w}(t)$ from a base solution $\mathbf{W}_0$. The subtraction of $\mathbf{W}_0$ will result in zero-valued boundaries for the POD modes wherever Dirichlet boundary conditions occur on the domain.

Φ is constructed by collecting observations of the solution $\mathbf{w}(t) - \mathbf{W}_0$ at different time intervals throughout the time integration of the full-system dynamics. These observations are called snapshots [35] and are generally collected in a way that provides a good variety of flow field dynamics minimizing linear dependence. The snapshot generation procedure is sometimes referred to as POD training [5].

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_w. The linear space is defined by the span of the snapshots [30]. POD identifies a new basis for this linear space that is optimally convergent [18] in the sense that no other set of basis functions will capture as much energy in as few dimensions as the POD basis functions. To identify the POD basis, the snapshots are compiled into an $N \times Q$ matrix S, known as the snapshot matrix. The mapping function Φ is then developed using

$$S^T S V = V \Lambda, \tag{8a}$$

$$\Phi = SV. \tag{8b}$$

Here V is the matrix of eigenvectors of $S^T S$, and Λ is the corresponding 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 Φ will be an $N \times M$ matrix. It determines the coordinates of $\mathbf{w}(t)$ in terms of the M remaining basis functions, $\boldsymbol{\phi}_k$.

The reduced order mappings for each fluid variable are developed separately, and individual S and V arrays are collocated as blocks into a larger set of arrays, also denoted S and V to form

$$\mathbf{U}(t) \approx \mathbf{U}_0 + \Phi \widehat{\mathbf{U}}(t), \tag{9a}$$

$$\Phi = SV. \tag{9b}$$

These versions of Eqs. (7) and (8b), respectively, apply to the entire set of fluid variables.

2.3. Reduced order model generation

Once the POD basis functions have been identified using the method of snapshots, the Euler equations must be recast to solve for the modal coefficients $\widehat{\mathbf{U}}(t)$ in lieu of the full system variables $\mathbf{U}(t)$. For compressible flows, this has generally been accomplished using a non-Galerkin method.

2.3.1. Subspace projection

The non-Galerkin approach, also known as the subspace projection method [6], uses the full system dynamics and a forward difference approximation to yield the following reduced order flow solver.

$$\widehat{\mathbf{U}}^{n+1} = \widehat{\mathbf{U}}^n + \Delta t \Lambda^{-1} (V^T V)^{-1} V^T S^T R(SV \widehat{\boldsymbol{U}}^n) \tag{10}$$

The pseudo inverse of V is shown, assuming modal truncation is employed. The inverse of Λ and pseudo inverse of V exist assuming modal truncation is employed to eliminate the zero valued eigenvalues of $S^T S$ and their corresponding eigenvectors [23]. Notice that $\Lambda^{-1}(V^T V)^{-1} V^T S^T$ from Eq. (10) is equivalent to $(\Phi^T \Phi)^{-1} \Phi^T$, and $(\Phi^T \Phi)^{-1} \Phi^T = \Phi^T$ when the modes are normalized.

The explicit subspace projection method from Eq. (10) relies on the full system function evaluation R at each time integration step. As such, the order of each integration step is not actually reduced. Computational improvement occurs because subspace projection can greatly increase the time step size allowed for stability [5]. The total number of time steps required for the explicit solver can be significantly reduced using this approach.

An implicit subspace projection method can also be generated for the time accurate case [24] using a chord method implementation with numerically approximated Jacobians. Although the equations governing the reduced-order operator $\widehat{R}(\widehat{\boldsymbol{U}})$ are never explicitly obtained, the value of the reduced-order operator at any time can be obtained from Eq. (10), using the full-system function evaluation R:

$$\widehat{R}(\widehat{\mathbf{U}}) = \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\widehat{\boldsymbol{U}}). \tag{11}$$

Consider the implicit, time-integration function $\widehat{F}$

$$\widehat{F}(\widehat{\boldsymbol{U}}^{n+1}) = \widehat{\boldsymbol{U}}^{n+1} - \widehat{\boldsymbol{U}}^{n} - \Delta t \widehat{R}(\widehat{\boldsymbol{U}}^{n+1}). \tag{12}$$

The value of $\widehat{\boldsymbol{U}}^{n+1}$ that results in $\widehat{F}(\widehat{\boldsymbol{U}}^{n+1}) = 0$ is the solution for the flow field at time $t_n + \Delta t$ from $\widehat{\boldsymbol{U}}^{n}$. The solution is readily obtained from Newton iterations. The full-system function call is required for each Newton iteration since R is used to evaluate $\widehat{R}(\widehat{U})$ in Eq. (11). The Jacobian term for F is related to the Jacobian term for R as follows:

$$\frac{\mathrm{d}F(\boldsymbol{U})}{\mathrm{d}\widehat{\boldsymbol{U}}} = \boldsymbol{\Phi} - \Delta t \frac{\mathrm{d}R(\boldsymbol{U})}{\mathrm{d}\widehat{\boldsymbol{U}}}, \tag{13a}$$

$$\frac{\mathrm{d}\widehat{F}(\widehat{\boldsymbol{U}})}{\mathrm{d}\widehat{\boldsymbol{U}}} = \Lambda^{-1}(V^T V)^{-1} V^T S^T \frac{\mathrm{d}F(\boldsymbol{U}))}{\mathrm{d}\widehat{\boldsymbol{U}}}. \tag{13b}$$

For computational purposes it is efficient to obtain the reduced-order Jacobian, $\mathrm{d}\widehat{F}(\widehat{\boldsymbol{U}})/\mathrm{d}\widehat{\boldsymbol{U}}$, numerically from $\mathrm{d}R(\boldsymbol{U})/\mathrm{d}\widehat{\boldsymbol{U}}$ in Eq. (13a). The Jacobian $\mathrm{d}R(\boldsymbol{U})/\mathrm{d}\widehat{\boldsymbol{U}}$ is obtained using a central-difference approximation, and only requires $2M$ function calls.

2.4. Projection methods

We use both Galerkin and direct projection to recast the governing equations in terms of $\widehat{\mathbf{U}}(t)$. This is done using the modal expansion given in Eq. (9a), along with the methods of linear functional analysis.

First, consider the set of all functions that are potential outcomes of the evolution equation (Eq. (2)) for each of the fluid variables. These functions must have sufficient smoothness such that the spatial derivatives within the operator R are defined. The set of functions that are Lebesgue integrable when taken to the second power meet this condition. Otherwise known as the L^2 functions, this set forms a complete linear space that admits spatial functions with any number of discontinuities (a countably infinite number) [30,36]. The domain $\mathscr{D}(R)$ and range $\mathscr{R}(R)$ of the operator R from Eq. (2) are the subset of L^2 functions that meet the boundary conditions posed to give the problem of interest a unique result.

Operating within L^2 is important, because when coupled with an inner product, L^2 provides a geometric structure that gives meaning to the notion of projection. The inner product provides a measure of "closeness" between two functions, and in L^2 the inner product between two functions $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$ is defined as follows [30]

$$\langle \psi_1, \psi_2 \rangle = \int_{\Omega} \psi_1(\mathbf{x}) \psi_2(\mathbf{x}) \, \mathrm{d}\mathbf{x} \quad \text{for } \psi_1, \psi_2 \in L^2. \tag{14}$$

The $\langle , \rangle$ operator is an inner product because it has the properties of additivity, homogeneity, symmetry and positive definiteness [30]. L^2 is a Hilbert space when endowed with the inner product from Eq. (14) [30].

The geometric structure of the inner product space allows us to to introduce the notion of orthogonality. Two functions ψ_1 and ψ_2 within an inner product space are orthogonal when

$$\langle \psi_1, \psi_2 \rangle = 0. \tag{15}$$

This is denoted as $\psi_1(\mathbf{x}) \perp \psi_2(\mathbf{x})$. Two subsets A and B are orthogonal when all the functions within A are orthogonal to all the functions within B. This is denoted $A \perp B$.

2.4.1. Galerkin projection

The geometric structure of L^2 provides the means to minimize errors due to the uniqueness of orthogonal projections in Hilbert spaces. Galerkin projection seeks to minimize error resulting in the use of the modal approximation described earlier in Eq. (5). The residual (E_r) is defined using the dynamics in Eq. (2) as

$$E_\mathrm{r} \equiv \frac{\mathrm{d}U^M}{\mathrm{d}t} - R(U^M), \tag{16}$$

where $U^M = \sum_{k=1}^{M} a_k(t)\phi_k(\mathbf{x})$, and E_r is identically zero when $U(\mathbf{x}, t) = U^M$. The base-flow term in not included in U^M for simplicity. We can minimize the errors by forcing them to lie outside of $\mathscr{R}(R)$, which for this case means we want $E_\mathrm{r} \in \mathscr{R}(R)^\perp$. By the definition of orthogonality, the following must be true for any $\psi(\mathbf{x}) \in \mathscr{R}(R)$,

$$\langle E_\mathrm{r}, \psi(\mathbf{x}) \rangle = 0. \tag{17}$$

Since every function $\psi(\mathbf{x}) \in \mathscr{R}(R)$ can be expressed as a linear combination of the orthogonal basis functions $\phi_k(\mathbf{x})$, the requirement $E_\mathrm{r} \in \mathscr{R}(R)^\perp$ is completely specified by the series of inner products $\langle E_\mathrm{r}, \phi_k(\mathbf{x}) \rangle = 0$ given below, where $k = 1$ through M.

$$\left\langle \frac{\mathrm{d}U^M}{\mathrm{d}t} - R(U^M), \phi_k(\mathbf{x}) \right\rangle = 0. \tag{18}$$

Eq. (18) is called Galerkin projection. The inner products maps spatial functions to scalars, including the boundary data. What remains is a system of coupled ODEs describing the dynamics of the modal coefficients $a_k(t)$ in time, one for each of the M basis functions.

Once the spatial domain is discretized, $\phi_k(\mathbf{x})$ becomes $\boldsymbol{\phi}_k$, a vector of dimension N. The inner product becomes a summation as follows:

$$\langle \psi_1, \psi_2 \rangle \approx \sum_{i=1}^{I} \sum_{j=1}^{J} \psi_1(i,j)\psi_2(i,j)\Delta y(i,j)\Delta x(i,j), \tag{19}$$

where i and j are indices of the discretized mesh, the total number of discrete points is $N = I * J$, and typically $N \gg M$. Once the inner products have been evaluated, time integration of the reduced-order set of ODEs can be accomplished very quickly (order M^2 computations as opposed to order N^2 or more). However, evaluating the inner products is not trivial. Even though integrating Eq. (18) numerically using Eq. (19) might seem simple enough, the nonlinearities and spatial derivatives in the operator R complicate matters greatly.

2.4.2. Direct projection

Direct projection is an extension of the subspace projection method from Eq. (10). When the full-order function evaluation R is replaced with a general Taylor series expansion [27],

$$R(\mathbf{U}_0+\Delta\boldsymbol{U})=R(\boldsymbol{U}_0)+\sum_{i=1}^{N}\Delta u_i\frac{\partial R}{\partial u_i}(\boldsymbol{U}_\mathbf{0})+\frac{1}{2}\sum_{i,j=1}^{N}\Delta u_i\Delta u_j\frac{\partial^2 R}{\partial u_i\partial u_j}(\boldsymbol{U}_\mathbf{0})+\frac{1}{3!}\sum_{i,j,k=1}^{N}\Delta u_i\Delta u_j\Delta u_k \times\frac{\partial^3 R}{\partial u_i\partial u_j\partial u_k}(\boldsymbol{U}_\mathbf{0})+\text{H.O.T.}, \tag{20}$$

pre-multiplication by $(\Phi^T\Phi)^{-1}\Phi^T$ and post-multiplication by Φ will produce a set of nonlinear ODEs, with the linear portion in state-space form. The numerical computation of the Jacobians is done with central differences as described above for the implicit subspace formulation (Eq. (13b)). The number of full-order function evaluations to accomplish the projection is $(2M)^p$ where p is the order of the largest derivative term retained in the expansion.

3. Galerkin projection of the Euler equations

Nonlinearities that couple the conserved fluid variables make Galerkin projection of the Euler equations very difficult to evaluate. Consider the algebra involved inserting modal expansions for the conserved variables into E and F from Eqs. (1b) and (1c). If each of the four conserved variables had $M = 10$ fluid modes $\boldsymbol{\phi}_k$, then the numerator for the momentum term would contain several groupings of 100 terms mixing the various modal coefficients $a_k(t)$. Evaluation of these expressions would be encumbered by the presence of 10 linear terms in the denominator. Since we are interested in representing the fluxes in terms of the $a_k(t)$, the resulting expression is algebraically irreducible. One might hope that orthogonality would cancel many of these terms and greatly reduce the complexity when the inner product is evaluated. However, the presence of the squared nonlinearities in the numerator, the terms in the denominator, along with the need to differentiate the resulting terms eliminates this possibility. These problems are exacerbated in the energy expression where cubic nonlinearities make for 1000-term groupings in the numerator, and 100 nonlinear terms in the denominator. One can see how assuming incompressible flow reduces the number of terms, since density is constant in the incompressible case, and the energy expression is decoupled from momentum.

The first technique to make Galerkin projection manageable is to recast the Euler equations using the primitive velocity components (u and v). This will enable code implementations that allow numerical differentiation and integration. While the time derivative of $U(\mathbf{x},t)$ will be complicated by this modification, the flux expressions are simplified as follows:

$$E=\begin{bmatrix}\rho u\\ E_\mathrm{T}\gamma-E_\mathrm{T}+\frac{3u^2\rho}{2}+\frac{v^2\rho}{2}-\frac{1}{2}u^2\gamma\rho-\frac{1}{2}v^2\gamma\rho\\ \rho uv\\ E_\mathrm{T}u\gamma+\frac{u^3\rho}{2}+\frac{1}{2}uv^2\rho-\frac{1}{2}u^3\gamma\rho-\frac{1}{2}uv^2\gamma\rho\end{bmatrix}, \tag{21a}$$

$$F=\begin{bmatrix}\rho v\\ \rho uv\\ E_\mathrm{T}\gamma-E_\mathrm{T}+\frac{u^2\rho}{2}+\frac{3v^2\rho}{2}-\frac{1}{2}u^2\gamma\rho-\frac{1}{2}v^2\gamma\rho\\ E_\mathrm{T}v\gamma+\frac{1}{2}u^2v\rho+\frac{v^3\rho}{2}-\frac{1}{2}u^2v\gamma\rho-\frac{1}{2}v^3\gamma\rho\end{bmatrix}. \tag{21b}$$

In eliminating the denominator terms, we have increased the nonlinearities by one order of magnitude. Quartic nonlinearities in the energy terms now produce groupings of 10,000 terms instead of 1000 when $M = 10$, but since there is no modal expansion in the denominator, code implementation for this case is greatly simplified. For notational convenience, a special vector of flow variables $U_P(\mathbf{x},t)$ is introduced in addition to the original flow vector $U(\mathbf{x},t)$,

$$U(\mathbf{x},t)=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E_{\mathrm{T}}\end{bmatrix},\quad U_P(\mathbf{x},t)=\begin{bmatrix}\rho\\ u\\ v\\ E_{\mathrm{T}}\end{bmatrix}. \tag{22}$$

$U_P(\mathbf{x},t)$ resembles the vector of primitive flow variables except that E_{T} is used in lieu of pressure. Also, $\gamma=\gamma(\mathbf{x},t)$ will be treated as a variable and the flux terms will be denoted as $E(U_P,\gamma)$ and $F(U_P,\gamma)$. The Euler equations from Eq. (1a) are written below using this notation,

$$\frac{\partial U}{\partial t}+\frac{\partial E(U_P,\gamma)}{\partial x}+\frac{\partial F(U_P,\gamma)}{\partial y}=0. \tag{23}$$

3.1. Numerical dissipation

The Euler equations are neutrally stable, so some form of dissipation will be required to yield a stable system of ODEs. One way to incorporate dissipation is to add a diffusive term to the right-hand side of the Euler equations. A constant μ modifies this term, and its value can be set to provide the amount of dissipation required. We include this term in the development and will study the effects of numerical dissipation in the results section. Eq. (23) is modified as follows to include the dissipative terms

$$\frac{\partial U}{\partial t}+\frac{\partial E(U_P,\gamma)}{\partial x}+\frac{\partial F(U_P,\gamma)}{\partial y}=\mu\frac{\partial^2 U_P}{\partial x^2}+\mu\frac{\partial^2 U_P}{\partial y^2}. \tag{24}$$

3.2. Numerical differentiation in the transformed space

The next technique to enable evaluation of the Galerkin projection is to numerically differentiate the flux expressions. Since the POD process relies on a full-order numerical solver for snapshot generation, a spatial discretization of the flow field consistent with the fluid modes is readily available for finite-difference approximation of the spatial derivatives. To use finite differences, the physical domain is transformed to a computational domain using a mapping $\xi=\xi(x,y)$ and $\eta=\eta(x,y)$. The ξ,η mapping is defined such that the values of ξ,η are simply the i and j index numbers associated with each pair of x,y values in the physical domain, therefore $\Delta\xi$ and $\Delta\eta$ are unity everywhere. For a structured grid, this mapping is one-to-one and invertible [1].

To operate in $L(\xi,\eta)$ index space, the chain rule is used to recast Eq. (24) into the transformed expression. Multiplying through by

$$J=\frac{\partial x}{\partial\xi}\frac{\partial y}{\partial\eta}-\frac{\partial y}{\partial\xi}\frac{\partial x}{\partial\eta}$$

and manipulating the resulting expression produces the following [1]:

$$\frac{\partial(JU)}{\partial t}+\frac{\partial E_1}{\partial\xi}+\frac{\partial F_1}{\partial\eta}-D_\xi-D_\eta=0, \tag{25a}$$

$$E_1\equiv(Ey_\eta-Fx_\eta), \tag{25b}$$

$$F_1\equiv(-Ey_\xi+Fx_\xi), \tag{25c}$$

$$D_\xi \equiv J\mu \frac{\partial^2 U_P}{\partial x^2}, \tag{25d}$$

$$D_\eta \equiv J\mu \frac{\partial^2 U_P}{\partial y^2}. \tag{25e}$$

Here

$$x_\xi = \frac{\partial x}{\partial \xi}, \quad x_\eta = \frac{\partial x}{\partial \eta}, \quad y_\xi = \frac{\partial y}{\partial \xi} \text{ and } y_\eta = \frac{\partial y}{\partial \eta}$$

are the metrics of the transformation, and J is the Jacobian of the transformation. The metric terms are functions of $\mathbf{x}$ only, since we are not considering grids that either deform or move. Deforming grids can be used with POD if additional expressions are evaluated [2].

The dissipative terms in Eq. (25a), D_ξ and D_η, need to be transformed into ξ, η coordinates for evaluation. Consider the rth fluid variable in U_P, denoted U_{P_r}. After significant manipulations, the x-direction dissipation term becomes,

$$J\mu \frac{\partial^2 U_{P_r}}{\partial x^2} = \mu\left(\frac{\partial^2 U_{P_r}}{\partial \xi^2}\frac{1}{J}y_\eta^2 - 2\frac{\partial^2 U_{P_r}}{\partial \xi \eta}\frac{1}{J}y_\eta y_\xi + \frac{\partial^2 U_{P_r}}{\partial \eta^2}\frac{1}{J}y_\xi^2\right) + J\mu\left(\frac{\partial U_{P_r}}{\partial \xi}\frac{\partial^2 \xi}{\partial x^2}\right) + J\mu\left(\frac{\partial U_{P_r}}{\partial \eta}\frac{\partial^2 \eta}{\partial x^2}\right), \tag{26}$$

and the pure second-derivative terms reduce to the following form:

$$\begin{aligned}\frac{\partial^2 \xi}{\partial x^2} = \frac{1}{J^3}\Big(&- Jy_{\eta\eta}y_\xi + Jy_{\eta\xi}y_\eta - x_\xi y_{\eta\xi}y_\eta^2 - x_{\xi\xi}y_\eta^3 + y_\xi x_{\eta\xi}y_\eta^2 \\ &+ y_{\xi\xi}x_\eta y_\eta^2 + x_\xi y_{\eta\eta}y_\xi y_\eta + x_{\eta\xi}y_\eta^2 y_\xi - x_{\eta\eta}y_\xi^2 y_\eta - y_{\eta\xi}x_\eta y_\xi y_\eta\Big),\end{aligned} \tag{27a}$$

$$\begin{aligned}\frac{\partial^2 \eta}{\partial x^2} = \frac{1}{J^3}\Big(&Jy_{\xi\eta}y_\xi - Jy_{\xi\xi}y_\eta + x_\xi y_{\eta\xi}y_\eta y_\xi + x_{\xi\xi}y_\eta^2 y_\xi - y_\xi^2 x_{\eta\xi}y_\eta - y_{\xi\xi}x_\eta y_\eta y_\xi - x_\xi y_{\eta\eta}y_\xi^2 \\ &- x_{\eta\xi}y_\eta y_\xi^2 + x_{\eta\eta}y_\xi^3 + y_{\eta\xi}x_\eta y_\xi^2\Big).\end{aligned} \tag{27b}$$

A similar set of manipulations yields the corresponding term for dissipation in the y direction

$$J\mu \frac{\partial^2 U_{P_r}}{\partial y^2} = \mu\left(\frac{\partial^2 U_{P_r}}{\partial \xi^2}\frac{1}{J}x_\eta^2 - 2\frac{\partial^2 U_{P_r}}{\partial \xi \eta}\frac{1}{J}x_\eta x_\xi + \frac{\partial^2 U_{P_r}}{\partial \eta^2}\frac{1}{J}x_\xi^2\right) + J\mu\left(\frac{\partial U_{P_r}}{\partial \xi}\frac{\partial^2 \xi}{\partial y^2}\right) + J\mu\left(\frac{\partial U_{P_r}}{\partial \eta}\frac{\partial^2 \eta}{\partial y^2}\right). \tag{28}$$

The pure second-derivative terms for the y-direction dissipation are as follows:

$$\begin{aligned}\frac{\partial^2 \xi}{\partial y^2} = \frac{1}{J^3}\Big(&- Jx_{\eta\eta}x_\xi + Jx_{\eta\xi}x_\eta - x_\xi y_{\eta\xi}x_\eta^2 - x_{\xi\xi}y_\eta x_\eta^2 + y_\xi x_{\eta\xi}x_\eta^2 + y_{\xi\xi}x_\eta^3 + x_\xi^2 y_{\eta\eta}x_\eta + x_{\eta\xi}y_\eta x_\xi x_\eta \\ &- x_{\eta\eta}y_\xi x_\xi x_\eta - y_{\eta\xi}x_\eta^2 x_\xi\Big),\end{aligned} \tag{29a}$$

$$\begin{aligned}\frac{\partial^2 \eta}{\partial y^2} = \frac{1}{J^3}\Big(&Jx_{\xi\eta}x_\xi - Jx_{\xi\xi}x_\eta + x_\xi^2 y_{\eta\xi}x_\eta + x_{\xi\xi}y_\eta x_\eta x_\xi - y_\xi x_{\eta\xi}x_\eta x_\xi - y_{\xi\xi}x_\eta^2 x_\xi - x_\xi^3 y_{\eta\eta} \\ &- x_{\eta\xi}y_\eta x_\xi^2 + x_{\eta\eta}y_\xi x_\xi^2 + y_{\eta\xi}x_\eta x_\xi^2\Big).\end{aligned} \tag{29b}$$

3.3. Linearizing the transformed flux terms

Linearizing the transformed expression is a good way to reduce the number of terms that result when the modal expansion is applied to the cubic and quartic nonlinearities in the flux expressions. Many compressible flow problems are dominated by linear behavior, and should be well represented by the linearized expressions. When nonlinear terms are required, of-order analysis of the expanded nonlinear flux expression could be used to identify the dominant nonlinear terms to be added to the linearized expression. A vast majority of the nonlinear terms are very small, and can be eliminated prior to the evaluation of the inner products [18].

The linearized flux expressions require symbolic computation of Jacobian matrices; one for each flux expression. The Jacobians are evaluated at U_{P0} and γ_{xy}. U_{P0} is the base flow field used in the generation of the POD basis. γ_{xy} is the base gamma function used in the forcing term (forcing will be described in detail in Section 3.4). The linearized Jacobians for the flux terms are denoted A and B for $\partial E_1/\partial U_P$ and $\partial F_1/\partial U_P$, respectively. Likewise, the linearized forcing terms are denoted as $F_E \equiv \partial E_1/\partial\gamma$ and $F_F \equiv \partial F_1/\partial\gamma$.

Dividing through by the Jacobian of the transformation J will greatly simplify the linear time derivative evaluation in the Galerkin projection due to the orthogonality of the modes. J is not a function of time, and can be moved outside of the derivative. J is invertible, as long as the mapping to transformed space is one-to-one and onto [17] (which it will be for a structured grid).The linearized form of the transformed Euler equations is provided below

$$\frac{\mathrm{d}U}{\mathrm{d}t} + \frac{1}{J}A\frac{\partial U_P}{\partial\xi} + \frac{1}{J}B\frac{\partial U_P}{\partial\eta} + \frac{1}{J}F_e\frac{\partial\gamma}{\partial\xi} + \frac{1}{J}F_F\frac{\partial\gamma}{\partial\eta} + \frac{1}{J}D_\xi + \frac{1}{J}D_\eta = 0. \tag{30}$$

3.4. Forcing terms

Forcing is provided by spatially varying γ with time. Forcing enters Eq. (30) through the $\partial\gamma/\partial\xi$ and $\partial\gamma/\partial\eta$ terms. This type of forcing was selected to provide unsteady behavior without the added complication of a dynamic boundary condition. γ is a time varying spatial function, formed by modulating $f_{\mathrm{mod}}(\mathbf{x})$ with a time-varying scalar function $g(t)$. $f_{\mathrm{mod}}(\mathbf{x})$ is a spatial function that is discretized using the computational mesh for the full-order solver. $f_{\mathrm{mod}}(\mathbf{x})$ is read in with the data, and integrated in the Galerkin projection.

$$\gamma(\mathbf{x},t) = \gamma_{xy}(\mathbf{x}) + g(t)f_{\mathrm{mod}}(\mathbf{x}). \tag{31}$$

The function $\gamma_{xy}(\mathbf{x})$ is the base value γ assumes when no forcing is applied. We used a constant value of $\gamma_{xy}(\mathbf{x}) = 1.4$ for our research. The spatial derivatives of $\gamma(\mathbf{x},t)$ in index space,

$$\frac{\partial\gamma}{\partial\xi} = g(t)\frac{\partial f_{\mathrm{mod}}}{\partial\xi}, \tag{32a}$$

$$\frac{\partial\gamma}{\partial\eta} = g(t)\frac{\partial f_{\mathrm{mod}}}{\partial\eta}, \tag{32b}$$

are input into Eq. (30) as forcing during time integration. Here $\partial f_{\mathrm{mod}}/\partial\xi$ and $\partial f_{\mathrm{mod}}/\partial\eta$ are evaluated in index space using central differences. Specific choices for $g(t)$ and $f_{\mathrm{mod}}(\mathbf{x})$ will be described later when the model problem is discussed in detail.

3.5. Inserting the modal expansion

A modal expansion for each primitive fluid variable is obtained through POD. The task remaining is to insert modal coefficients into Eq. (30) and take the inner product. The modal expansion is given below. Notice that a base flow term is added, as specified in Eq. (9a).

$$\rho = \rho_0 + \sum_{m_\rho=1}^{M_\rho} a_{m_\rho}(t)\phi^\rho_{m_\rho}, \tag{33a}$$

$$u = u_0 + \sum_{m_u=1}^{M_u} b_{m_u}(t)\phi^u_{m_u}, \tag{33b}$$

$$v = v_0 + \sum_{m_v=1}^{M_v} c_{m_v}(t)\phi^v_{m_v}, \tag{33c}$$

$$E_\mathrm{T} = E_{\mathrm{T}0} + \sum_{m_{E_\mathrm{T}}=1}^{M_{E_\mathrm{T}}} d_{m_{E_\mathrm{T}}}(t)\phi^{E_\mathrm{T}}_{m_{E_\mathrm{T}}}. \tag{33d}$$

The set of vectors $\{\phi^\rho\}$, $\{\phi^u\}$, $\{\phi^v\}$ and $\{\phi^{E_\mathrm{T}}\}$ represent the POD basis functions for each of the four fluid variables: discretized according to the full-order computational mesh, and collocated into vectors. The total number of modes M is given by $M = M_\rho + M_u + M_v + M_{E_\mathrm{T}}$. Since the basis vectors for each fluid variable are generated independently using POD, only the basis vectors within each $\{\phi^r\}$ are orthogonal. In other words, $\{\phi^\rho\}$ is an orthogonal set of vectors, but the set $\{\phi^\rho\}$ is not orthogonal to $\{\phi^u\}$, $\{\phi^v\}$ or $\{\phi^{E_\mathrm{T}}\}$. When the individual fluid variable vectors are collocated as blocks to create the mapping given in Eq. (9a), then the modal vectors for the fluid system (the columns of Φ) are orthogonal by virtue of the block collocation.

The modal expansion enters Eq. (30) through the first derivative terms $\partial U_P/\partial\xi$ and $\partial U_P/\partial\eta$, and the derivative terms contained in D_ξ and D_η. Due to the linearity of the derivative operator, the first derivatives in ξ and η operate on the base flow terms and the modes. This is shown below for the ξ derivative of the rth fluid variable

$$\frac{\partial U_{P_\mathrm{r}}}{\partial\xi} = \frac{\partial U_{P_{\mathrm{r}0}}}{\partial\xi} + \sum_{m=1}^{M_\mathrm{r}} a_\mathrm{r}(t)\frac{\partial\phi^r_m}{\partial\xi}. \tag{34}$$

The derivatives for all the modes and the base flow terms are computed numerically using central differences i.e., without upwinding. These derivatives are input as arrays into the Galerkin projection algorithm (we will summarize all of the input required at the end of this section).

The presence of a non-uniform base flow term is a complicating factor, because if the base flow is included in the modal expansion for U_P in D_ξ and D_η (Eqs. (25d) and (25e)), the base flow terms produce very large constant gain terms (modified by μ) under Galerkin projection. These terms act like a step input and pollute the reduced-order response. Since dissipation stabilizes the time integration of the dynamics, only the time varying portion of the modal expansion requires dissipation. Consequently, the base flow terms are removed from

$$\frac{\partial U_P}{\partial\xi}, \quad \frac{\partial U_P}{\partial\eta}, \quad \frac{\partial^2 U_P}{\partial\xi^2}, \quad \frac{\partial^2 U_P}{\partial\eta^2}, \quad \frac{\partial^2 U_P}{\partial\xi\eta}$$

prior to insertion of the modal expansion into D_ξ and D_η.

3.6. Galerkin projection

3.6.1. Taking the inner product

The inner product introduced in Eq. (14), modified for operation in index space, is as follows

$$\langle f,g\rangle = \int_{\Omega_\xi}\left(\int_{\Omega_\eta} f(\xi,\eta)g(\xi,\eta)\,\mathrm{d}\eta\right)\mathrm{d}\xi \quad \text{for } f,g\in L^2. \tag{35}$$

This expression will be numerically approximated by summation over the computational domain, and since $\Delta\xi$, and $\Delta\eta$ are defined to be identically one, this becomes the following expression

$$\langle f,g\rangle \approx \sum_{i=1}^{I}\sum_{j=1}^{J} f(i,j)g(i,j)\Delta\eta(i,j)\Delta\xi(i,j) = \sum_{i=1}^{I}\sum_{j=1}^{J} f(i,j)g(i,j). \tag{36}$$

The Galerkin projection is obtained by taking the inner product of the linearized Euler equations (Eq. (30)) with each mode. Eq. (30) is repeated below, and some notation is defined to aid in presenting the Galerkin projection.

$$\frac{\mathrm{d}U}{\mathrm{d}t}+\frac{1}{J}A\frac{\partial U_P}{\partial\xi}+\frac{1}{J}B\frac{\partial U_P}{\partial\eta}+\frac{1}{J}F_E\frac{\partial\gamma}{\partial\xi}+\frac{1}{J}F_F\frac{\partial\gamma}{\partial\eta}-\frac{1}{J}D_\xi-\frac{1}{J}D_\eta=0,$$

$$A=\begin{bmatrix}a_{11}&a_{12}&a_{13}&a_{14}\\ a_{21}&a_{22}&a_{23}&a_{24}\\ a_{31}&a_{32}&a_{33}&a_{34}\\ a_{41}&a_{42}&a_{43}&a_{44}\end{bmatrix}=\begin{bmatrix}A_\rho\\ A_u\\ A_v\\ A_{E_\mathrm{T}}\end{bmatrix},\quad B=\begin{bmatrix}B_\rho\\ B_u\\ B_v\\ B_{E_\mathrm{T}}\end{bmatrix}.$$

Galerkin projection of Eq. (30) is given below for the jth mode on each fluid variable. The subscripts ρ,u,v and E_T on the Jacobian terms A and B represent row vectors from the Jacobian matrices corresponding to the four fluid variables. Operations such as $A_\rho(\partial U_P/\partial\xi)$ are the vector dot product. Likewise, the Jacobian vectors F_E and F_F, along with the dissipation vectors D_ξ and D_η are broken into scalar components.

Even though the flux terms are a mix of fluid variables, as are the time derivatives in the momentum equations, a single primitive fluid variable is associated with each expression in the Euler equations. Continuity is associated with density (ρ), x-momentum is associated with x-velocity (u), y-momentum is associated with y-velocity (v) and the energy expression is associated with total energy (E_T). This permits orthogonality to simplify the linear time-derivative terms.

$$\left\langle \frac{\mathrm{d}(\rho)}{\mathrm{d}t}+\frac{1}{J}A_\rho\frac{\partial U_P}{\partial\xi}+\frac{1}{J}B_\rho\frac{\partial U_P}{\partial\eta}+\frac{1}{J}F_{\rho_\xi}\frac{\partial\gamma}{\partial\xi}+\frac{1}{J}F_{\rho_\eta}\frac{\partial\gamma}{\partial\eta}-\frac{1}{J}D_{\rho_\xi}-\frac{1}{J}D_{\rho_\eta},\phi_j^\rho\right\rangle=0, \tag{37a}$$

$$\left\langle \frac{\mathrm{d}(\rho u)}{\mathrm{d}t}+\frac{1}{J}A_u\frac{\partial U_P}{\partial\xi}+\frac{1}{J}B_u\frac{\partial U_P}{\partial\eta}+\frac{1}{J}F_{u_\xi}\frac{\partial\gamma}{\partial\xi}+\frac{1}{J}F_{u_\eta}\frac{\partial\gamma}{\partial\eta}-\frac{1}{J}D_{u_\xi}-\frac{1}{J}D_{u_\eta},\phi_j^u\right\rangle=0, \tag{37b}$$

$$\left\langle \frac{\mathrm{d}(\rho v)}{\mathrm{d}t}+\frac{1}{J}A_v\frac{\partial U_P}{\partial\xi}+\frac{1}{J}B_v\frac{\partial U_P}{\partial\eta}+\frac{1}{J}F_{v_\xi}\frac{\partial\gamma}{\partial\xi}+\frac{1}{J}F_{v_\eta}\frac{\partial\gamma}{\partial\eta}-\frac{1}{J}D_{v_\xi}-\frac{1}{J}D_{v_\eta},\phi_j^v\right\rangle=0, \tag{37c}$$

$$\left\langle \frac{\mathrm{d}(E_\mathrm{T})}{\mathrm{d}t}+\frac{1}{J}A_{E_\mathrm{T}}\frac{\partial U_P}{\partial\xi}+\frac{1}{J}B_{E_\mathrm{T}}\frac{\partial U_P}{\partial\eta}+\frac{1}{J}F_{E_\mathrm{T}\xi}\frac{\partial\gamma}{\partial\xi}+\frac{1}{J}F_{E_\mathrm{T}\eta}\frac{\partial\gamma}{\partial\eta}-\frac{1}{J}D_{E_\mathrm{T}\xi}-\frac{1}{J}D_{E_\mathrm{T}\eta},\phi_j^{E_\mathrm{T}}\right\rangle=0. \tag{37d}$$

The M modes for the set of four equations will yield a set of M coupled ODEs in time, with the unknowns being the time history of the modal coefficients.

3.6.2. Flux terms

The flux portion of the Galerkin projection can be separated from the time derivative. Since the evaluation of the flux terms is general for the four fluid variables, the subscript r will be used to represent any of the four equations associated with ρ, u, v or E_T.

$$R^{\xi}_{r,j}(U_P) = -\left\langle \frac{1}{J}A_r\frac{\delta U_P}{\delta\xi} + \frac{1}{J}F_{r_\xi}\frac{\delta\gamma}{\delta\xi} - \frac{1}{J}D_{r_\xi}, \phi^r_j \right\rangle, \tag{38}$$

$$R^{\eta}_{r,j}(U_P) = -\left\langle \frac{1}{J}B_r\frac{\delta U_P}{\delta\eta} + \frac{1}{J}F_{r_\eta}\frac{\delta\gamma}{\delta\eta} - \frac{1}{J}D_{r_\eta}, \phi^r_j \right\rangle. \tag{39}$$

The ODE expression for the jth mode of the rth fluid variable is,

$$\left\langle \frac{\mathrm{d}U[r]}{\mathrm{d}t}, \phi^r_j \right\rangle = R^{\xi}_{r,j}(U_P) + R^{\eta}_{r,j}(U_P), \tag{40}$$

where U_P is approximated using Eqs. (33a)–(33d).

At this point in the analysis, the portion of $R^{\xi}_{r,j}(U_P)$ and $R^{\eta}_{r,j}(U_P)$ inside the inner product symbols is evaluated symbolically using *Mathematica* or some suitable symbolic-math software package. Prior to integration, the symbolic expressions within $R^{\xi}_{r,j}(U_P)$ and $R^{\eta}_{r,j}(U_P)$ consist of a long additive string of terms. Since integration is a linear operator, the inner product of each term can be computed individually and combined later to produce the final expression. These individual terms will consist of a grouping of spatial functions multiplied together. Each grouping will be multiplied by either a single modal coefficient (since we are using linearized dynamics), or a single modal coefficient times either the dissipation modifier μ or $g(t)$ from the forcing term. The groupings of spatial functions will consist of some combination of the metrics $(x_\xi, x_\eta, y_\xi, y_\eta)$ and $1/J$, the modes $(\{\phi^\rho\}, \{\phi^u\}, \{\phi^v\}, \{\phi^{E_T}\})$, the first and second derivatives of the modes, the base flow (ρ_0, u_0, v_0, E_T0), the first derivatives of the base flow, the base gamma function γ_{xy}, and the derivatives of the modulation function $(\delta f_{\mathrm{mod}}/\delta\xi, \delta f_{\mathrm{mod}}/\delta\eta)$. Since we have discretized the domain, these spatial functions are arrays. The first step in efficiently integrating the flux expressions is isolating the array terms from the scalar time functions and μ using the symbolic-math software. Once these terms are isolated, all the array data should be loaded into memory and each group of arrays should be reduced to a single array using element by element multiplication. Finally, the elements of each condensed array term should be summed. This is the inner product operation in index space, and it produces one scalar value for each grouping. The scalar coefficient terms are recombined with their corresponding grouping of spatial functions, but now these grouping have been reduced to scalar modifiers. Once like terms are combined, what remains is a linear combination of the scalar coefficients and the scalar coefficients times either μ or the forcing term $g(t)$. This small string of terms represents the evaluation of $R_{r,j}(U_P)$.

3.6.3. Continuity and energy time derivatives

The time derivative portion of the continuity and energy expressions can be simplified because the POD modes are orthogonal. The time derivatives transfer to the modal coefficients when the expansion is evaluated. For continuity and energy, the set of ODEs resulting from the Galerkin procedure is as follows, with the right-hand side being the flux expressions defined in Eqs. (38) and (39).

$$\frac{\mathrm{d}r_j(t)}{\mathrm{d}t}\left\langle \phi^r_j, \phi^r_j \right\rangle = R^{\xi}_{r,j}(U_P) + R^{\eta}_{r,j}(U_P). \tag{41}$$

The modes used in this research were not normalized. If the modes are normalized $\langle \phi^r_j, \phi^r_j \rangle$ is unity.

3.6.4. Momentum time derivatives

Because density is multiplied by velocity within the time derivatives for the x and y-momentum equations, orthogonality does not simplify the time derivatives in this case. The x-momentum time derivative is evaluated below

$$\begin{aligned}\left\langle \frac{\mathrm{d}(\rho u)}{\mathrm{d}t}, \phi_j^u \right\rangle &= \left\langle \rho \frac{\mathrm{d}u}{\mathrm{d}t} + u \frac{\mathrm{d}\rho}{\mathrm{d}t}, \phi_j^u \right\rangle = \left\langle \rho \frac{\mathrm{d}u}{\mathrm{d}t}, \phi_j^u \right\rangle + \left\langle u \frac{\mathrm{d}\rho}{\mathrm{d}t}, \phi_j^u \right\rangle \\ &= \left\langle \left(\rho_0 + \sum_{m_\rho=1}^{M_\rho} a_{m_\rho}(t) \phi_{m_\rho}^\rho \right) \sum_{m_u=1}^{M_u} \frac{\mathrm{d}(b_{m_u}(t))}{\mathrm{d}t} \phi_{m_u}^u, \phi_j^u \right\rangle \\ &\quad + \left\langle \left(u_0 + \sum_{m_u=1}^{M_u} b_{m_u}(t) \phi_{m_u}^u \right) \sum_{m_\rho=1}^{M_\rho} \frac{\mathrm{d}(a_{m_\rho}(t))}{\mathrm{d}t} \phi_{m_\rho}^\rho, \phi_j^u \right\rangle. \end{aligned} \tag{42}$$

The y-momentum time derivative is similar. For the momentum expressions, the set of ODEs resulting from the Galerkin procedure involves coupled time derivative terms in the left-hand-side combined with the flux expressions previously defined Eqs. (38) and (39). The coupled time derivative expressions are evaluated using the same procedure outlined for evaluating the flux terms in Section 3.6.2.

3.7. Input data

To evaluate the inner products, data must be obtained from the full-order simulation. This data is in the form of arrays, with array indexing corresponding to an ordered mapping of locations on the computational mesh. Data for all of the non-time varying spatial functions appearing in the previous expressions are required to evaluate the Galerkin projection. The pure second derivatives of the metric terms $\partial^2\xi/\partial x^2$ and $\partial^2\eta/\partial x^2$ from Eqs. (27a) and (27a), and $\partial^2\xi/\partial y^2$ and $\partial^2\eta/\partial y^2$ from Eqs. (29a) and (29a) were computed within the full-order simulation, multiplied by J and provided as arrays of data. A summary of the complete list of array terms to be computed is provided in Table 1.

3.8. System of ODEs

One the terms are segregated, the arrays are multiplied together and summed, and all the terms are collected, the resulting expressions for the fluxes and the time derivatives can be combined into a system of ODEs. This is the evaluation of Eq. (40) for all dependent variables, which will consist of M total

Table 1
Array input data

Modes	$\{\phi^\rho\}$, $\{\phi^u\}$, $\{\phi^v\}$, $\{\phi^{E_T}\}$
First derivatives of the modes	$\{\delta\phi^\rho/\delta\xi\}$, $\{\delta\phi^u/\delta\xi\}$, $\{\delta\phi^v/\delta\xi\}$, $\{\delta\phi^{E_T}/\delta\xi\}$, $\{\delta\phi^\rho/\delta\eta\}$, $\{\delta\phi^u/\delta\eta\}$, $\{\delta\phi^v/\delta\eta\}$, $\{\delta\phi^{E_T}/\delta\eta\}$
Second derivatives of the modes	$\{\delta^2\phi^\rho/\delta\xi^2\}$, $\{\delta^2\phi^u/\delta\xi^2\}$, $\{\delta^2\phi^v/\delta\xi^2\}$, $\{\delta^2\phi^{E_T}/\delta\xi^2\}$, $\{\delta^2\phi^\rho/\delta\eta^2\}$, $\{\delta^2\phi^u/\delta\eta^2\}$, $\{\delta^2\phi^v/\delta\eta^2\}$, $\{\delta^2\phi^{E_T}/\delta\eta^2\}$ $\{\delta^2\phi^\rho/\delta\eta\xi\}$, $\{\delta^2\phi^u/\delta\eta\xi\}$, $\{\delta^2\phi^v/\delta\eta\xi\}$, $\{\delta^2\phi^{E_T}/\delta\eta\xi\}$
Metrics and Jacobian of the transformation	x_ξ, x_η, y_ξ, y_η, J
Second derivative metric terms	$J(\partial^2\xi/\partial x^2)$, $J(\partial^2\eta/\partial x^2)$, $J(\partial^2\xi/\partial y^2)$, $J(\partial^2\eta/\partial y^2)$
Base flow terms	ρ_0, u_0, v_0, E_{T0}
First derivatives of base flow terms	$\delta\rho_0/\delta\xi$, $\delta u_0/\delta\xi$, $\delta v_0/\delta\xi$, $\delta E_{T0}/\delta\xi$, $\delta\rho_0/\delta\eta$, $\delta u_0/\delta\eta$, $\delta v_0/\delta\eta$, $\delta E_{T0}/\delta\eta$,
Forcing terms	γ_{xy}, f_{mod}

expressions. The left-hand side terms are the time derivatives of the coefficients, and the right-hand side terms come from the evaluation of $R_{r,j}(U_P)$. Since the nonlinear terms in the time derivative expressions from the momentum equations are very small, they can generally be neglected, leaving a linear system

$$B_{\text{sys}} \frac{\mathrm{d}\widehat{\mathbf{U}}}{\mathrm{d}t} = A_{\text{sys}}(\mu)\widehat{\mathbf{U}} + F_{\text{sys}}. \tag{43}$$

This can be placed in the more standard form,

$$\frac{\mathrm{d}\widehat{\mathbf{U}}}{\mathrm{d}t} = G(\mu)\widehat{\mathbf{U}} + F_{\text{aug}}, \tag{44}$$

where $G(\mu) = (B_{\text{sys}})^{-1}A_{\text{sys}}(\mu)$, and $F_{\text{aug}} = (B_{\text{sys}})^{-1}F_{\text{sys}}$. This system has M states, where M is typically 8–12. Stability of the linear system will be addressed in detail later. The system is easily and quickly integrated, and the time history of the modal coefficients $\widehat{\mathbf{U}}$ can be expanded to provide the time history of the flow field using Eq. (9a).

4. Direct projection of the Euler equations

Direct projection begins with a Taylor series expansion (see Eq. (20)) of the continuum form of the full-system dynamics (from Eq. (2)),

$$\frac{\partial \boldsymbol{U}}{\partial t} = R(\boldsymbol{U}_{\mathbf{0}}, \Delta_0) + \frac{\partial R}{\partial \boldsymbol{U}}(\boldsymbol{U}_0, \Delta_0)(\boldsymbol{U} - \boldsymbol{U}_0) + \frac{\partial R}{\partial \Delta}(\boldsymbol{U}_0, \Delta_0)(\Delta - \Delta_0) + \text{H.O.T.}, \tag{45}$$

where Δ is a scalar variable replacing $g(t)$ from Eq. (31). For this research, the second-order terms and higher are truncated, leaving an approximation for the full system dynamics linearized about the base flow term $\boldsymbol{U}_0$. Nonlinear terms could be included in future applications by retaining some of the higher order terms. The Jacobian terms $\partial R/\partial \boldsymbol{U}(\boldsymbol{U}_0, \Delta_0)$ and $\partial R/\partial \Delta(\boldsymbol{U}_0, \Delta_0)$ are constant. The reduced order mapping from (9a) is inserted into Eq. (45), and the base flow terms fall out. The result is pre-multiplied by $\boldsymbol{\Phi}^T$, and the left-inverse is applied to produce

$$\frac{\partial \widehat{\mathbf{U}}}{\partial t} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T R(\boldsymbol{U}_{\mathbf{0}}, \Delta_0) + (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \boldsymbol{U}}(\boldsymbol{U}_0, \Delta_0)\boldsymbol{\Phi}\widehat{\boldsymbol{U}} + (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \Delta}(\boldsymbol{U}_0, \Delta_0)(\Delta - \Delta_0). \tag{46}$$

When the base flow $\boldsymbol{U}_0$ is a steady state, unforced flow condition, then Δ_0 is zero and $R(\boldsymbol{U}_0) = 0$. For this case, a simplified expression is obtained

$$\frac{\partial \widehat{\boldsymbol{U}}}{\partial t} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \boldsymbol{U}}(\boldsymbol{U}_0)\boldsymbol{\Phi}\widehat{\boldsymbol{U}} + (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \Delta}(\boldsymbol{U}_0)\Delta. \tag{47}$$

Eq. (47) is seen to be in state-space form.

$$A_{DP} \equiv (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \boldsymbol{U}}(\boldsymbol{U}_0)\boldsymbol{\Phi}$$

is an $M \times M$ matrix, and

$$B_{DP} \equiv (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T \frac{\partial R}{\partial \Delta}(\boldsymbol{U}_0)$$

is an $M \times 1$ matrix, so that Eq. (47) can be written as

$$\frac{d\widehat{\mathbf{U}}}{dt} = A_{DP}\widehat{\mathbf{U}} + B_{DP}g(t). \tag{48}$$

The coefficient matrix A_{DP} will be close to but not equal to $G(\mu = 0)$ from Eq. (44) since they are generated by different approximations. For direct projection, the Jacobian terms are approximated with the discrete Euler equations using central differences about $\widehat{\boldsymbol{U}}_0$. The dynamics of the numerical scheme will be characterized and contained within A_{DP}, where $G(\mu = 0)$ comes directly from the continuum Euler equations. This is true for the forcing terms to a lesser extent, and $B_{DP}g(t) \approx F_{\text{aug}}$ from Eq. (44).

5. Two-dimensional flow over a bump with forcing

The sample problem described below was used to explore the implementation of both projection techniques. A full-order numerical solver was developed to provide data for the POD basis, as well as a reliable solution for comparison with the reduced-order model results. The full-order model is described below, along with the key features of the resulting flow field that we will use to assess performance.

5.1. Problem description

We considered 2-D, inviscid flow along a solid wall containing a small circular arc. Unsteadiness was introduced by varying γ as described in Eq. (31). $f_{\text{mod}}(\mathbf{x})$ was a 2-D Gaussian distribution as depicted graphically in Fig. 1. A small bump of length 1 and height 0.001 was centered at $x = 0, y = 0$, and the Gaussian curve was centered directly above it at $x = 0, y = 2.5$. The covariance terms determine the spread of the Gaussian curve, and were $\sigma_x = \sigma_y = 0.25$. The time varying function $g(t)$ was defined as,

$$g(t) = \alpha \sin(\omega t). \tag{49}$$

For the results that follow, the amplitude was fixed at $\alpha = 0.07$, and frequency ω was fixed at 0.2145. This type of forcing was selected to provide unsteady behavior without the added complication of a dynamic boundary condition.

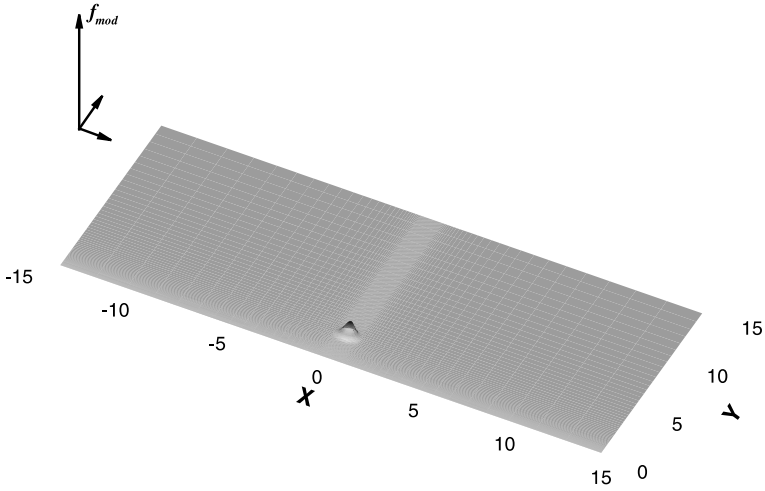


Fig. 1. Modulating function for γ.

5.2. Grid

A rectangular domain of dimension $-23.668 < x < 23.668$ and $0 < y < 24.6$ was used for this research. A structured grid was generated for this domain using 141 nodes along the solid surface and 116 nodes extending to the far field boundary.

5.3. Solver

A cell-centered, finite-volume scheme was the basis for the full-order, numerical fluid solver used in this research. The flux terms were computed using first-order Roe averaging, and the flow variables were evaluated as cell averages. Time integration across the computational mesh was used to obtain flow solutions. This was accomplished with a first-order accurate, forward Euler approximation. The full-order model for this problem had 64,400 DOFs. The Euler equations were non-dimensionalized using Reynolds number, and all the results were obtained in non-dimensional form (including time units). The finite-volume fluid solver was validated by comparison with experimental data [20]. The reader is referred to a number of texts on CFD for detailed description of these techniques (i.e. see [37]).

5.4. Full-order behavior

The full-order model was time integrated using the baseline forcing frequency $\omega = 0.2145$, the baseline amplitude variation of $\alpha = 0.07$, and a free stream Mach number $M_\infty = 0.75$. The time histories of the pressure at the $1/2$-chord and $3/4$-chord positions on the bump surface are shown in Fig. 2. The flow was initialized with steady-state conditions obtained without forcing.

Once the flow was fully developed, the unsteady pressure on the surface of the bump varied based upon the amplitude and frequency of the forcing term. The difference between the maximum and minimum surface pressures (non-dimensional, for fully developed flow) at the $1/2$-chord location on the bump surface are denoted ΔP_{mid}. The response of ΔP_{mid} from the full system is shown in Fig. 3 for variations in both frequency and amplitude. At the baseline amplitude, forcing frequencies greater than 0.1 introduced nonlinearities into the dynamics as indicated by the resonance peak in Fig. 3. At forcing frequencies between 0.1 and 0.5 the nonlinearities were weak. The average mid-point pressure was 1.265, which was slightly less than the free stream pressure $P_\infty = 1/\gamma M^2 = 1.27$. This difference was 33% of the pressure amplitude variation, and provides a measure of nonlinearity. At frequencies between 0.5 and 3 the nonlinearities were stronger, and at frequencies greater than 3, the fluid system did not react quickly enough for forcing to affect pressure on the bump surface, and the frequency response quickly fell off. At the baseline

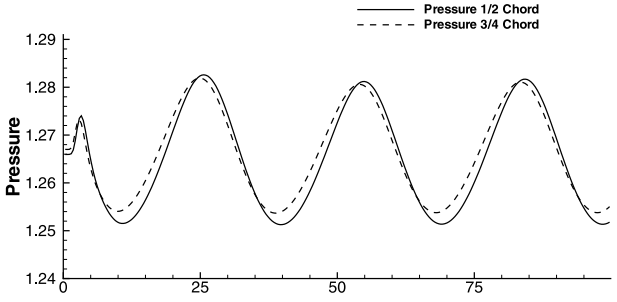


Fig. 2. Full-order model bump pressures.

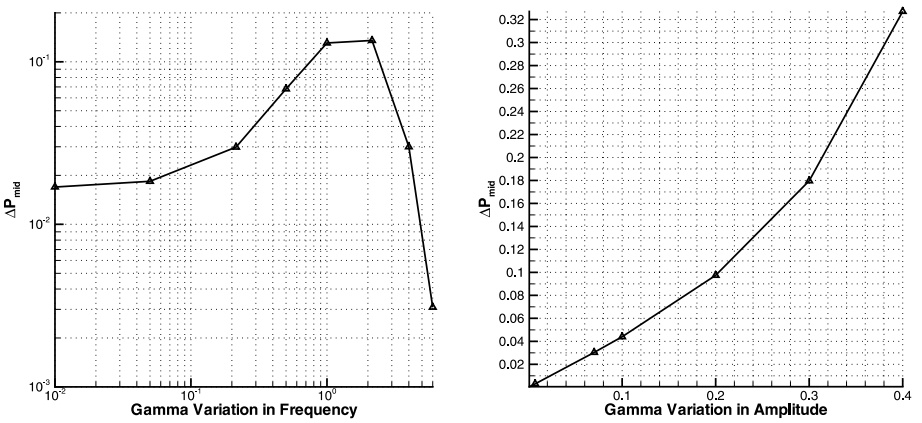


Fig. 3. Full-order pressure response (L) in frequency (rad/s) (R) in amplitude.

frequency, amplitude variation greater than 0.1 introduced nonlinearities as well. The $\alpha = 0.07$ baseline amplitude was well within the linear range at this frequency.

6. Results

The Galerkin and direct projection methods were used to identify a system of ODEs for the model problem at the baseline amplitude and frequency. Additional dissipation was required to stabilize the ODEs, and varying types and amounts of dissipation were explored. The ODEs generated from both methods were compared by examining the eigenvalues of the state-space dynamics matrix. Accuracy was assessed by comparing the modal amplitude responses. The time histories of $\widehat{\mathbf{U}}$, once expanded to $\mathbf{U}$ using Eq. (9a), were also compared with the pressure time history of the full-order system shown in Fig. 2.

6.1. Stability of the reduced-order system

Galerkin and direct projection were used to obtain two reduced-order linear systems (Eqs. (44) and (48)). The POD basis functions were the same for both ROMs, and they were obtained using snapshots from time integration of the full order model at the baseline frequency and amplitude. A set of 50 snapshots was taken at even intervals from start-up through a single cycle of oscillation (approximately 28 s, where s refers to non-dimensional time units). The first two modes for each fluid variable contained over 99% of the energy content, and system realization was performed using a total of $M = 8$ fluid modes (2 modes per fluid variable).

The forcing terms from both methods were nearly equal:

$$F_{\text{aug}} = \begin{bmatrix} 0 \\ 0 \\ -2.19753\alpha\sin(\omega t) \\ 8.19537\alpha\sin(\omega t) \\ -8.92715\alpha\sin(\omega t) \\ 29.7541\alpha\sin(\omega t)) \\ 0.153992\alpha\sin(\omega t) \\ 2.29104\alpha\sin(\omega t) \end{bmatrix}, \quad B_{DP}g(t) = \begin{bmatrix} 0 \\ 0 \\ -2.2007\alpha\sin(\omega t) \\ 8.2122\alpha\sin(\omega t) \\ -8.9350\alpha\sin(\omega t) \\ 29.7799\alpha\sin(\omega t)) \\ 0.1542\alpha\sin(\omega t) \\ -2.2959\alpha\sin(\omega t) \end{bmatrix}. \tag{50}$$

Galerkin projection also yielded small, constant forcing terms (of order 10^{-4}) on each element of $\widehat{\boldsymbol{U}}$, which were dropped. The eigenvalues of the resulting dynamics matrices for both methods, $G(\mu = 0)$ and A_{DP} are shown in Fig. 4. The frequency response of both systems was similar, as reflected in the similarities in the imaginary portion of the eigenvalues. Direct projection was more dissipative, as evidenced by the larger, negative magnitude in the real portion of the complex pairs. Notice that both systems contained one or more complex pairs with positive real parts, an indication of an unstable linear system. This was expected from Galerkin projection since no dissipation was provided by the neutrally stable Euler equations. The instability in the direct projection dynamics was somewhat surprising, since the Jacobian term was approximated using function evaluations from a very dissipative first-order solver. We suspect the instability occurred because the central-difference technique used a small number of function evaluations. Since these evaluations were perturbations about a steady-state solution, the sampling of the full-order response did not excite a great deal of dissipation in the full-order solver. A variety of dissipation schemes were explored in an attempt to stabilize both schemes while preserving the accuracy of the reduced-order model.

6.1.1. Primitive variable based dissipation

The term μ modulated an artificial viscosity term introduced into the flow physics in Eq. (24). Galerkin projection placed μ into the dynamics matrix to enact dissipation in a manner consistent with the projection of artificial viscosity. Fig. 5 shows the behavior of the system roots as μ was varied from 0 to 0.1. Stability of the system required a value of $\mu = 0.075$ to bring the high-frequency eigenvalue pair onto the left-half plane. This magnitude of artificial dissipation is consistent with that required for a full-order simulation based on a similar numerical scheme.

Notice that for $\mu > 0.075$, the eigenvalues associated with lower frequency eigenmodes are pushed deep into the left half plane. The additional dissipation blunted the low frequency response of the Galerkin POD/ROM, and produced an inaccurate ROM (accuracy will be quantified in detail later). A much smaller amount of dissipation ($\mu \approx 0.025$) was sufficient to stabilize the low frequency response of the system.

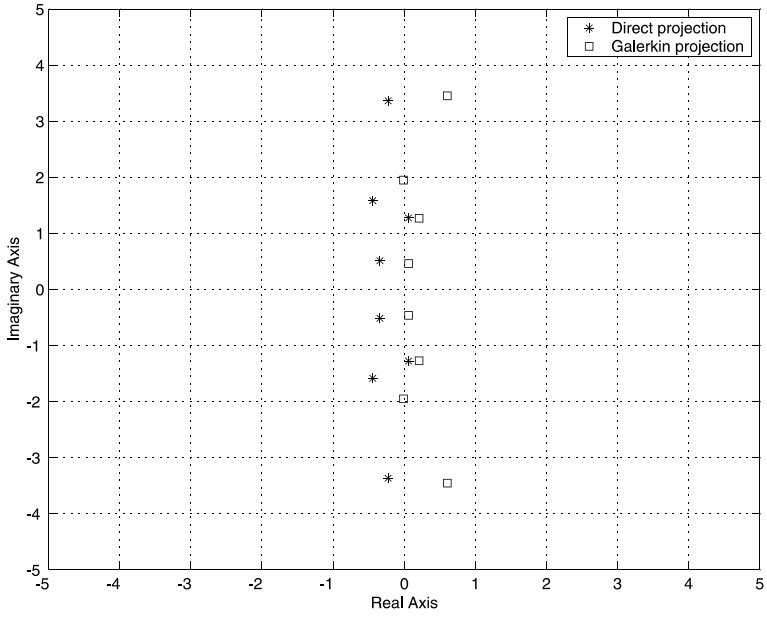


Fig. 4. Modal energy in POD basis functions for density.

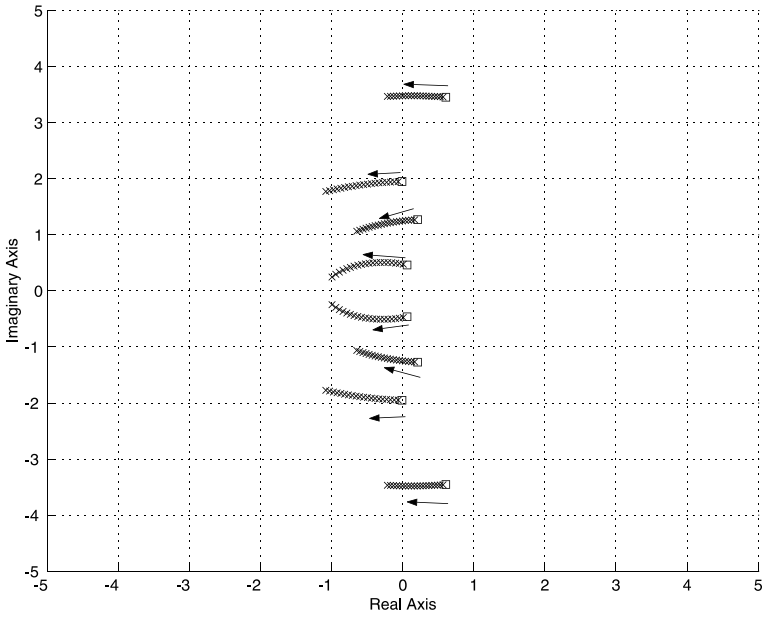


Fig. 5. Root locus $\mu = 0 \rightarrow 0.1$ (the squares are the poles of the uncontrolled system).

Essentially, a hybrid method was required that could damp the high frequency response, independently of the lower frequency component.

6.1.2. Linear quadratic regulation

Linear quadratic regulation (LQR) was used in addition to artificial viscosity to stabilize the Galerkin reduced-order system. LQR was also able to stabilize the direct projection POD/ROM, which contained no artificial viscosity term.

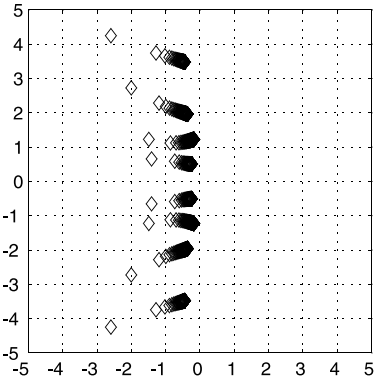
LQR required a control input $\mathbf{u}$ in the state dynamics expression, and Eqs. (44) and (48) were modified as follows:

$$\frac{\mathrm{d}\widehat{\mathbf{U}}}{\mathrm{d}t} = A\widehat{\mathbf{U}} + B\mathbf{u}, \tag{51}$$

where $\mathbf{u}$ was a vector of control inputs with the same length as $\widehat{\mathbf{U}}$ (8 for this model problem), and B was the matrix B_{DP} from Eq. (48). A was simply the dynamics matrix from either method. LQR provided the optimal gain matrix K such that the state-feedback law $\mathbf{u} = -K\widehat{\mathbf{U}}$ minimized the cost function,

$$J = \int (\widehat{\mathbf{U}}^T Q \widehat{\mathbf{U}} + \mathbf{u}^T R \mathbf{u} + 2\widehat{\mathbf{U}}^T N \mathbf{u})\,\mathrm{d}t. \tag{52}$$

N was set equal to 0 because $\widehat{\mathbf{U}}^T N\mathbf{u}$ represents the cost of correlating control inputs with responses, which was not a concern. Both Q and R were diagonal matrices formed by multiplying the identity matrix by a scalar value. Large values of Q placed a high cost on non-zero system responses, and large values of R placed a high cost on using control inputs. Consequently, large values of R and small values of Q reduced the effects of control. With $Q = 1$, the root locus for R ranging from 1 to 1000 is shown in Fig. 6 for both the Galerkin and direct projection POD/ROMs. As R gets large the unstable eigenvalues approach their reflection over the imaginary axis. One benefit of LQR is the guarantee of a stable system, regardless of the



weights [28]. At large R, the stable eigenvalues are relatively unaffected. For the Galerkin POD/ROM, the high frequency term is stabilized without modifying the low frequency response.

The LQR stabilized POD/ROMs with $Q = 1$ and $R = 1000$ will be the baseline for the performance analysis that follows. The controlled system has the following state-space form.

$$\begin{aligned} \dot{\mathbf{x}} &= A_c \widehat{\mathbf{x}} + B\mathbf{u}, \\ \mathbf{y} &= C\widehat{\mathbf{x}} + D\mathbf{u}. \end{aligned} \tag{53}$$

Here $\mathbf{x} = \widehat{\mathbf{U}}$, C is the identity matrix, and $D = 0$. A_c is defined as $A_c = A - BK$ with A and B taken from Eq. (51), and K is the LQR controller gain matrix.

6.2. Coefficient time history

POD/ROM modal responses were benchmarked against the full-order solution by projecting the full-system response (see Fig. 2) onto the POD basis functions using the left-inverse of Eq. (9a) (which is simply $\widehat{\boldsymbol{U}}(t) = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T(\boldsymbol{U}(t) - \boldsymbol{U}_0)$). The full-order modal response is shown in Fig. 7, and is compared to a variety of POD/ROM results. All the POD/ROMs used the same set of basis functions. The performance of the non-Galerkin methods was represented by an implicit subspace projection POD/ROM (see Section 2.3.1). Once the flow was fully developed, the time responses for the first modes were replicated exactly; however, the non-Galerkin solvers exhibit an error predicting the second modes for each fluid variable. These modal-response errors translated to a physical flow field error that will be detailed later.

Direct and Galerkin projection POD/ROM results are also included in Fig. 7. Both POD/ROMs showed excellent agreement with the full order results for the first modes. The direct projection also predicted all the second mode behavior accurately. The Galerkin POD/ROM tracked all the second modes closely, with the exception of the second mode for y-momentum. The dissipation model was modified by varying μ and R values to eliminate the error evident in Fig. 7, but this introduced new errors in the responses for other modes. The impact of the second y-momentum mode on the overall solution was insignificant, and eliminating this mode from the reduced-order mapping had no noticeable affect on the physical flow field reproduced from the modes.

The Galerkin POD/ROM shown in Fig. 7 was stabilized by combining primitive variable dissipation with $\mu = 0.045$ and LQR with $Q = 1$, and $R = 1000$. This combination of parameters was found to produce the best results via trial and error. Iterations were not computationally intensive since they involved the

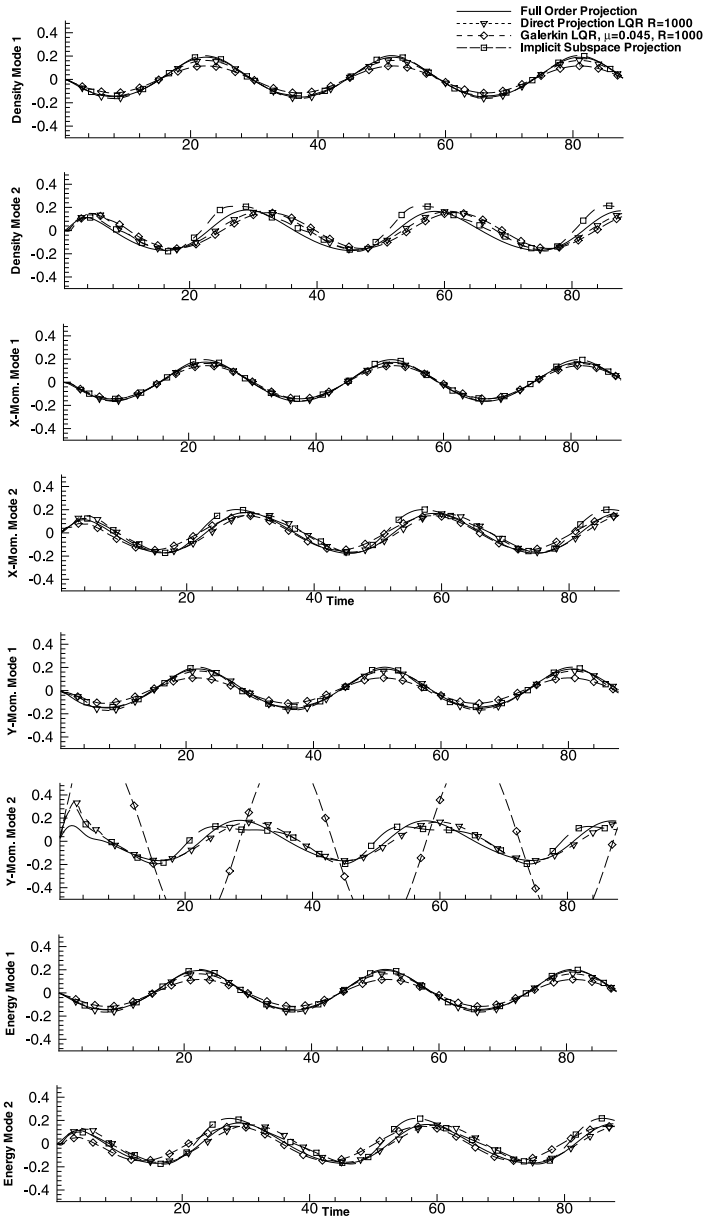
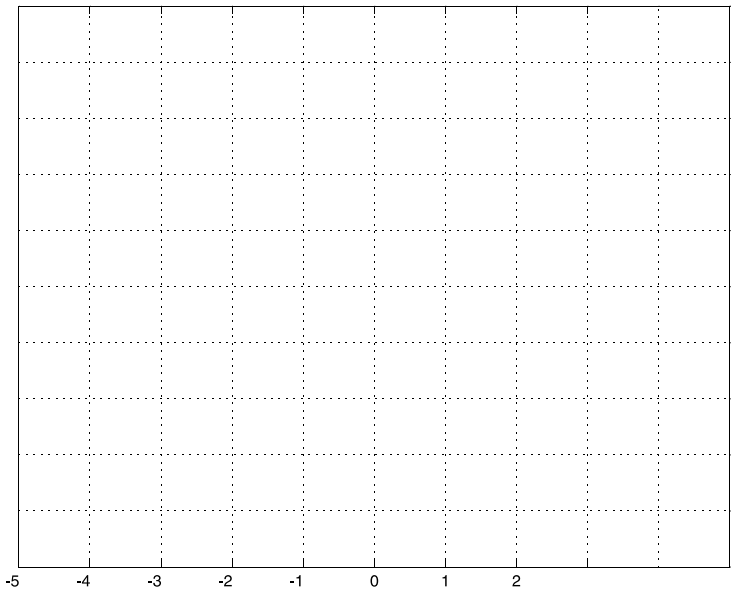


Fig. 7. Time history of reduced-order coefficients.

reduced-order system. The full-order data from snapshot collection was used, so no additional full-order simulations were required. Since R was so large, LQR only served to stabilize the high-frequency response. The manipulation of μ drove the amplitude of the modal response curves. The eigenvalue plots of the controlled system for both projection methods are provided in Fig. 8. Notice that the eigenvalue pairs are nearly identical, demonstrating that the two fundamentally different approaches produced essentially the same result.



6.3. Time history

Time responses of surface pressure for various POD/ROMs are compared to the full-order system in Fig. 9, which provides the time history for three cycles of fully developed flow. The reduced-order results using the non-Galerkin approach are considered along with the Galerkin and direct projection POD/ROMs.

The explicit subspace projection POD/ROM introduced a slow growing instability after the third cycle. These instabilities were consistent with other instances of subspace projection reported in the literature [5]. Eliminating the second y-momentum mode improved stability as shown in Fig. 9. Otherwise, the implicit and explicit subspace projection POD/ROMs produced identical results, with a bias error in the pressure response. This bias error was the result of the peculiar behavior of the response for the second modes as described in Fig. 7.

Neither the Galerkin nor the direct projection POD/ROMs suffered any instabilities, and average errors were less than 5% for both. The Galerkin projection POD/ROM, using $\mu = 0.045$ and LQR with $R = 1000$ for dissipation, matched the full-system response better than the direct projection results. The direct projection POD/ROM (using LQR $R = 1000$) had a slightly muted response. This difference is insignificant, since the direct projection response could be tuned to match the full-order results more closely by iterating on the value of R. No POD/ROM matched the transient period ($t < 15$ s) within 10% average error, but this degraded performance was expected since the transient flow field data was not sufficiently sampled by the snapshots.

6.4. Computational performance

The motivation for employing projection methods with POD/ROM was to realize a computational performance improvement consistent with the reduction in the number of DOFs. Computational performance, summarized in Table 2, was assessed by measuring the wall clock time for each solver to provide an 88 s time history of the flow. All computations were run on a 800 MHz Pentium-based, personal computer.

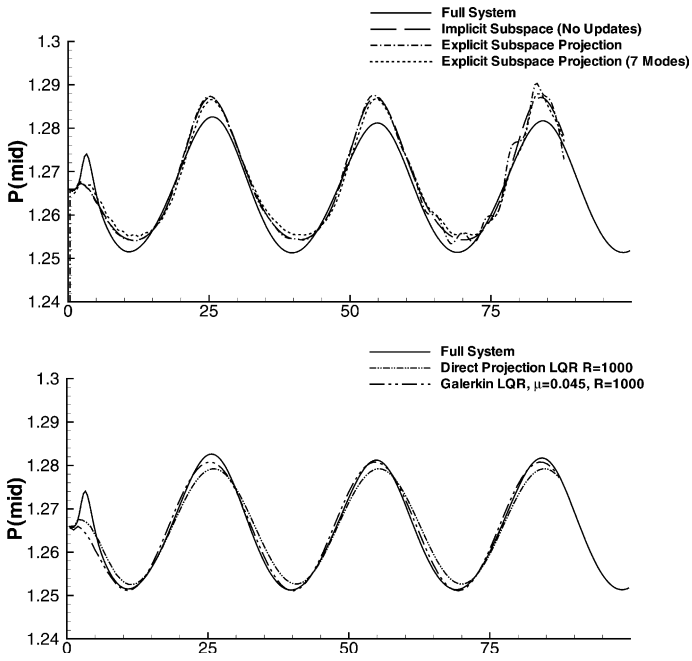


Fig. 9. Time history of panel pressures for various POD/ROMs.

Table 2
Computational performance

Flow solver	Number of DOFs	Wall clock time (s)
Full order	64,400	310.256
Subspace projection POD/ROM	8	77.902
Galerkin or direct projection POD/ROM	8	0.08

The explicit non-Galerkin method (the fastest of the two subspace projection methods) was not nearly as efficient as the Galerkin or direct projection approach, because the subspace projection method used the flux evaluation of the full-order model. Subspace projection truncated high-frequency oscillations and permitted stable integrations at a Courant–Friedrichs–Lewy (CFL) condition of 5 (the full-order model required CFL < 1 for stability). The larger time step associated with the large CFL number yielded the reduction in wall-clock time shown in Table 2. Notice that subspace projection only provided one-order-of-magnitude reduction in computation time to accompany four-orders-of-magnitude reduction in DOFs. In contrast, Galerkin and direct projection both reduced compute time by four-orders-of-magnitude, and realized an improvement in performance consistent with the DOF reduction.

The cost of computing the Galerkin projection in *Mathematica* was also measured using wall clock time. Symbolic manipulation of the equations required about 3 s of compute time, as did loading the data from the full system solver into memory. The element by element multiplications took the most time, and the array summations of the results went quickly. Together these took 38 s. These combine for a total of 44 s in compute time to generate the reduced-order model using Galerkin projection. This was one order of magnitude less time than a single run of the full-order solver, which generally took over 300 s. Direct projection required even less time, about 5 s, since only $2M$ full-system function calls (or 16 with $M = 8$)

were required to compute the linear Jacobian using central differences. The cost of direct projection would increase greatly if nonlinear terms were included.

6.5. Boundary enforcement

Both projection methods provide an operator $\widehat{R}$ on Euclidean M-space, where M is the number of reduced-order variables. Each vector in M-space maps to a unique function in L^2 through the linear transformation of $\psi(\mathbf{x}) = \sum_{k=1}^{M} a_k \phi_k(\mathbf{x})$, where $\{\phi_k(\mathbf{x})\}$ are the POD basis functions. Therefore, M-space is isomorphic to the linear subspace ℓ on L^2, where ℓ is defined as the algebraic span of the POD basis [17]. The evolution of L^2 functions based on the operator R from Eq. (2) are propagated in time, subject to the condition that the domain and range are both the linear subspace ℓ. The accuracy of this condition depends on the data content of the snapshots. The data must be collected such that the actual range of the operator R from Eq. (2) is well approximated by ℓ. During the evolution of the system, $\widehat{R}$ continually projects the solution onto ℓ. Regardless of the data collection, some error is inherent because the range of R is not a linear space within L^2 due to the boundary constraints. This permits either projection method to yield a solution that violates the boundary conditions when the solution vector $\psi(\mathbf{x}) \in \{\ell \cap \mathscr{R}(R)\}$.

Use of the POD basis functions, along with the base flow condition, precludes serious violations of boundary constraints under either projection method. This assertion applies as long as the boundary behavior is well represented in the snapshot data. The POD basis vectors are simply the principal components of the sampled data, and as such they will conform to the boundary behavior in the set of snapshots. Each POD basis function will meet the boundary conditions. However, ℓ is formed by all possible linear combinations of the basis functions, and all the elements of ℓ will not meet the boundary conditions. Even so, the first POD basis function contains a majority of the flow energy (e.g. 97% of the flow energy in the first density mode). Since the first mode obeys the boundary conditions, a one-mode solution would not violate the boundary conditions at all. Addition of the remaining modes introduces the potential for about 3% of the flow energy to violate the boundary condition. These percentages refer to the perturbation to the base flow condition. Since the base flow conforms to the boundary conditions and generally contains a majority of the flow energy, using a base flow in the Galerkin formulation further reduces any concern. Boundary violations using this method are small relative to the 5% errors introduced through numerical approximation, modal truncation, data collection, operator linearization, and artificial dissipation.

For the bump flow problem, the maximum percentage of the localized flow momentum through any individual cell on the solid wall was observed as a function of time. This error metric was averaged over time to measure violation of the solid-surface boundary condition. The max-percent momentum through the wall averaged 0.28% for either the Galerkin or direct projection based POD/ROM, and 0.27% for subspace projection. The perturbation was about 5% of the base flow, so these findings were consistent with the 3% of perturbation estimate described above.

7. Conclusions

Two projection methods, Galerkin projection and direct projection, were developed in conjunction with POD to obtain reduced-order models of a compressible flow field. While fundamentally different in their theoretical development, the two approaches were similar in that they yielded very low-dimensional systems of ODEs to describe the dynamics of coefficients that modify the POD basis functions. The coefficients were reduced-order variables that were used to reconstruct the entire flow field for a time-accurate solution. For the linear case, we demonstrated that the two methods produce nearly identical systems of ODEs when dissipative parameters were introduced, and properly tuned.

Evaluation of the Galerkin projection was made tractable through the use of primitive variables. Primitive variables simplified the flux expressions and enabled symbolic manipulations in *Mathematica* to provide for an implementation that was general for any number of POD basis functions, and any 2-D application of the Euler equations with stationary boundaries. The spatial domain was discretized in a manner suitable for the full-order solver, and spatial derivatives within the Galerkin projection were evaluated numerically using central differences. The Galerkin projection was approximated using numerical integration and could be evaluated efficiently, using less than 10% of the computational expense necessary for a full-order run.

Direct projection was introduced as a new technique to obtain a low-dimensional, state-space model from the POD reduced-order mapping. In direct projection, the flux evaluation in the Euler equations was approximated with a truncated Taylor series expansion that was projected into low-order, state-space form using the reduced-order mapping. The method relied on the efficient computation of Jacobian terms, enabled by the small number of POD reduced-order variables. Reduced-order Jacobians were approximated from a small number of full-system function evaluations using the chord method.

Both Galerkin and direct projection were demonstrated using linearized Euler equations with oscillatory forcing. They yielded POD/ROMs four orders-of-magnitude faster than the full system. Boundary conditions were enforced by the POD basis functions. The use of a base flow combined with the optimal convergence of the POD basis functions virtually eliminated boundary violations.

The dissipation from the full-order model was not retained in either the Galerkin or direct projection techniques, since upwinding of the spatial derivatives in the full-system was not adequately captured in either projection. As a result, the ODEs produced by both methods were unstable. Dissipation was introduced to stabilize the system by two means: the first involved adding dissipative terms to the Euler equations prior to the Galerkin projection. The second involved using a linear quadratic regulator to stabilize the system. A combination of primitive variable dissipation and LQR provided the most accurate Galerkin POD/ROM. Use of LQR alone with direct projection provided an equally accurate POD/ROM. Careful selection of the amount of artificial viscosity, and the weights associated with the LQR process produced ODEs that accurately replicated full-system behavior. In general, LQR weights were chosen to stabilize the unstable eigenmodes of the linear system, without affecting stable eigenmodes. For the Galerkin POD/ROM, low frequency eigenmode responses were tuned using artificial viscosity, then the high-frequency eigenmodes were stabilized with LQR. Direct projection results required no such tuning, once the proper LQR weights were identified. These tuning procedures were accomplished using the reduced-order models, and had little computational cost.

These promising results for the linearized Euler equations with fixed boundaries motivates the extension of this projection methods to nonlinear problems with moving boundaries. The computations required to evaluate Galerkin projection of the nonlinear Euler equations would not be prohibitive following the approach described in this paper. Of-order analysis could be used to reduce the number of nonlinear terms prior to data collection and integration. This might favor the Galerkin approach over direct projection, where computing higher-order terms might become prohibitive, even with reduced-order Jacobians. The extension to moving boundaries would require coupling the boundary operator with the flow field to produce a single operator describing the entire system. Either Galerkin or direct projection of this new operator could use modes describing both the interior and the boundary to produce very efficient reduced-order models for free boundary problems.

Acknowledgements

The authors gratefully acknowledge the Air Force Office of Scientific Research (Dr. Dean Mook) for supporting this work through Grant 99VA01COR.

References

[1] J. Anderson, Computational Fluid Dynamics, McGraw-Hill, Inc, New York, 1995.
[2] J.S.R. Antonnen, Techniques for reduced-order modeling of aeroelastic structures, PhD Thesis, Air Force Institute of Technology, School of Engineering and Management, 2001.
[3] S. Banerjee, J.V. Cole, K.F. Jensen, Nonlinear model reduction strategies for rapid thermal processing systems, IEEE Transactions on Semiconductor Manufacturing 11 (1998) 266–275.
[4] P. Beran, C. Pettit, Prediction of nonlinear panel response using proper orthogonal decomposition, in: AIAA 2001–1292, 42nd Structural, Structural Dynamics, and Material Conference, Denver, CO, April 2001.
[5] P. Beran, W. Silva, Reduced order modeling: new approaches for computational physics, in: AIAA 2001–0853, 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2001.
[6] P.S. Beran, L.J. Huttsell, B.J. Buxton, C. Noll, G. Osswald, Computational aeroelastic techniques for viscous flow, in: CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA, June 22–25, 1999.
[7] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods in Fluid Dynamics, Springer, Berlin, 1988.
[8] E.H. Dowell, K.C. Hall, Modeling of fluid-structures interaction, Annual Reviews of Fluid Mechanics 33 (2001) 445–490.
[9] E.H. Dowell, K.C. Hall, J. Thomas, R. Florea, B. Epureanu, J. Heeg, Reduced order models in unsteady aerodynamics, in: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, St. Louis, MO, April 12–15, 1999.
[10] B.I. Epureanu, E.H. Dowell, K.C. Hall, A parametric analysis of reduced order models of potential flows in turbomachinery using proper orthogonal decomposition, 2001-GT-0434, in: Proceedings of ASME Turbo-Expo 2001, New Orleans, Louisiana, June 4–7, 2001.
[11] B.I. Epureanu, K.C. Hall, E.H. Dowell, Reduced order models of unsteady transonic viscous flows in turbomachinery, Journal of Fluids and Structures 18 (2000) 1215–1235.
[12] B.I. Epureanu, K.C. Hall, E.H. Dowell, Reduced order models in turbomachinery using inviscid–viscous coupling, Journal of Fluids and Structures 15 (2001) 255–276.
[13] R. Florea, K.C. Hall, E.H. Dowell, Eigenmode analysis and reduced order modeling of unsteady transonic full potential flow around isolated airfoils, in: CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA, June 22–25, 1999.
[14] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, New York, NY, 1996.
[15] D. Gottlieb, L. Lustman, C. Streett, Spectral methods for two-dimensional shocks, in: Spectral Methods for Partial Differential Equations, Symposium Proceedings, NASA Langley Research Center, VA, 16–18 August, 1984.
[16] K.C. Hall, J.P. Thomas, E.H. Dowell, Reduced-order modeling of unsteady small-disturbance flows using a frequency-domain proper orthogonal decomposition technique, in: AIAA 99-0655, 37th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 11–15, 1999.
[17] K. Hoffman, R. Kunze, Linear Algebra, second ed., Prentice-Hall, Upper Saddle River, NJ, 1971.
[18] P. Holmes, J. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, 1996.
[19] K. Ito, S.S. Ravindran, A reduced-order method for simulation and control of fluid flows, Journal of Computational Physics 143 (1998) 403–425.
[20] C. Kaplan, The flow of a compressible fluid past a curved surface, in: Report 768, National Advisory Committee for Aeronautics, 1944.
[21] K. Kunisch, S. Volkwein, Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition, Journal of Optimization Theory and Applications 102 (1999) 345–371.
[22] P.A. Legresley, J.J. Alonso, Airfoil design optimization using reduced order models based on proper orthogonal decomposition, in: AIAA 2000-2545, Fluids 2000 Conference and Exhibit, Denver, CO, June 2000.
[23] D.J. Lucia, Reduced order modeling for high speed flows with moving shocks, PhD Thesis, Air Force Institute of Technology, School of Engineering and Management, November 2001.
[24] D.J. Lucia, P.S. Beran, P.I. King, Reduced order modeling of an elastic panel in transonic flow, in: AIAA 2002-1594, 43rd AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, Denver, CO, April 22–25, 2002.
[25] D.J. Lucia, P.I. King, P.S. Beran, Reduced order modeling of a two-dimensional flow with moving shocks, Computers and Fluids 32 (2003) 917–938.
[26] D.J. Lucia, P.I. King, P.S. Beran, M.E. Oxley, Reduced order modeling for a one-dimensional nozzle flow with moving shocks, in: AIAA 2001-2602, 19th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, June 2001.
[27] J. Marsden, A. Tromba, Vector Calculus, W.H. Feeman and Company, New York, 1981.
[28] P.S. Maybeck, in: Stochastic Estimation and Control, vol. 3, Academic Press, Inc., Ill Fifth Avenue, New York, 1982, p. 10003.
[29] S.A. Mortara, J.C. Slater, P.S. Beran, An optimal proper orthogonal decomposition technique for the computation of nonlinear panel flutter, in: AIAA 2000-1936, 2000.

[30] A.W. Naylor, G.R. Sell, Linear Operator Theory in Engineering and Science, Springer, New York, 1982.
[31] H.M. Park, M.W. Lee, An efficient method of solving the Navier–Stokes equation for flow control, International Journal for Numerical Methods in Engineering 41 (1998) 1133–1151.
[32] C.L. Pettit, P.S. Beran, Reduced-order modeling for flutter prediction, in: AIAA 2000-1446-CP, 41st AIAA/ASCE/AHS/ASC Structural, Structural Dynamics and Matrerials Conference, Atlanta, GA. Accepted for publication in December 2001, International Journal for Numerical Methods in Engineering, April 2000.
[33] O.K. Rediniotis, J. Ko, X. Yue, A.J. Kurdila, Synthetic jets, their reduced order modeling and applications to flow control, in: AIAA 99-1000, 37th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 11–15, 1999.
[34] S.Y. Shvartsman, I.G. Kevrekidis, Nonlinear model reduction for control of distributed systems: a computer-assisted study, American Institute of Chemical Engineering 44 (1998) 1579–1595.
[35] L. Sirovich, Turbulence and the dynamics of coherent structures, part 1: Coherent structures, Quarterly of Applied Mathematics 45 (3) (1987) 561–571.
[36] I. Stakgold, Green's Functions and Boundary Value Problems, second ed., John Wiley and Sons, Inc., Washington, DC, 1998.
[37] J.C. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Company, Washington, DC, 1997.