Received: 29 March 2017 | Revised: 23 August 2017 | Accepted: 5 September 2017
DOI: 10.1002/fld.4468

RESEARCH ARTICLE

WILEY

Convergence acceleration of continuous adjoint solvers using a reduced-order model

Andrew L. Kaminsky | Reza Djeddi | Kivanc Ekici

Department of Mechanical, Aerospace, and Biomedical Engineering, The University of Tennessee, Knoxville, Tennessee 37996, USA

Correspondence
Andrew L. Kaminsky, Department of Mechanical, Aerospace, and Biomedical Engineering, University of Tennessee, Knoxville, TN 37996, USA.
Email: akamins1@vols.utk.edu

Funding information
Mechanical, Aerospace, and Biomedical Engineering Department; Bredesen Center for Interdisciplinary Research and Graduate Education; National Science Foundation, Grant/Award Number: CBET-1150332

Summary

A novel acceleration technique using a reduced-order model is presented to speed up convergence of continuous adjoint solvers. The acceleration is achieved by projecting to an improved solution within an iterative process solely using early solution results. This is achieved by forming basis vectors from early iteration adjoint solutions to perform model order reduction of the adjoint equations. The reduced-order model of the adjoint equations is then substituted into the full-order discretized governing equations to determine weighting coefficients for each basis vector. With these coefficients, a linear combination of the basis vectors is used to project to an improved solution. The method is applied to 3 inviscid quasi-1D nozzle flow cases including fully subsonic flow, subsonic inlet to supersonic outlet flow, and transonic flow with a shock. Significant cost reductions are achieved for a single application as well as repeated applications of the convergence acceleration technique.

KEYWORDS

convergence acceleration, continuous adjoint, proper orthogonal decomposition, POD, reduced-order model, ROM

1 | INTRODUCTION

Adjoint sensitivity, or gradient, calculation is a topic of considerable interest within computational fluid dynamics and aerodynamic design. In the framework of aerodynamic shape optimization, gradients were initially used to guide design through gradient-based optimization procedures,[1-3] but more recently, gradient usage has been extended to error analysis and grid adaptation,[4,5] constraint aggregation,[6,7] flow stability,[8] and control problems.[9]

Traditionally, the discretized adjoint equations are obtained either through discretization of the governing flow equations and subsequent linearization in the discrete approach or through linearization of the flow equations, resulting in the analytic adjoint equations, followed by discretization in the continuous approach. Sensitivity values found using the 2 approaches might vary, since discretization and linearization are generally noncommutative.[10] The discrete approach delivers the exact gradients of the discrete objective function, which are therefore consistent with gradients found through a finite difference approximation. Additionally, development of a discrete adjoint solver is simplified by

Nomenclature: c, speed of sound; E, total energy; $\mathbf{F}$, inviscid flux vectors; h, cross-sectional area; H, total enthalpy; I, objective function; M, number of basis vectors or Mach number; p, static pressure; $\mathbf{R}$, flow solver residual vector; $\mathbf{U}$, vector of conservation variables; u, axial velocity; γ, specific heat ratio; Φ, basis vector matrix; $\boldsymbol{\psi}$, adjoint snapshot vector; ρ, density; ξ, weighting coefficients; τ, pseudo-time

 Copyright © 2017 John Wiley & Sons, Ltd.

automatic differentiation.[11-13] Conversely, the continuous adjoint approach conveys the physical significance of the adjoint variables and the boundary conditions in a more meaningful manner, although increased effort is required to develop the complementary adjoint solver. It must be noted that solvers developed using the continuous adjoint approach benefit from reduced memory requirements and the ability to consider different objective functions at a lower cost compared to the discrete adjoint approach. It is frequently debated which approach has more merit,[14] but the continuous adjoint approach is adopted within this work.

Due to the similarity between the discretized adjoint equations and the discrete flow equations, essentially, the same numerical scheme can be used to solve both sets of equations.[15] For steady problems, multistage time-stepping schemes are frequently used to iteratively march the governing equations through pseudo-time to a steady state.

The iterative scheme is typically halted when the convergence error, approximated by the magnitude of the global residual, is driven to a sufficiently small value. Convergence acceleration techniques are used to decrease the convergence error as rapidly as possible. As mentioned earlier, since the solution methodologies for the flow and adjoint solutions are essentially identical, many flow convergence acceleration techniques, including local time-stepping,[16] multigrid,[17,18] and implicit residual smoothing,[19] have been applied successfully to accelerate adjoint sensitivity solutions.

As an alternative (or in addition) to the traditional acceleration techniques listed above, flow solution convergence acceleration has also been performed by transforming, or projecting, a slowly converging sequence to a new sequence that converges more rapidly to the same limit. These methods, called extrapolation methods, were first pioneered by Wynn's scalar and vector epsilon algorithms.[20] More recently, the minimal polynomial extrapolation has been used to approximate the eigenvalues of the flow solver.[21] The eigenvalues are used in the characteristic polynomial to extrapolate a solution to minimize the convergence error. Alternatively, Jespersen and Buning accelerated an iterative process for the Euler equations by annihilating the unstable eigenvalues.[22] Ekici et al extended this concept to the Navier-Stokes equations by using proper orthogonal decomposition (POD) to approximate the unstable eigenvalues.[23] Model order reduction has also been performed to find an initial guess for iterative solvers. Clemens et al used Gram-Schmidt orthogonalization to form a basis to extrapolate an initial guess in their subspace projection extrapolation scheme.[24] Similarly, Markovinović and Jansen used POD to project initial guesses from previous time steps.[25] The present authors have also used a similar reduced-order model (ROM)–based acceleration scheme to approximate converged flow solutions with considerable success.[26] Use of such techniques has not been reported for adjoint solution acceleration in the literature to date.

Model order reduction techniques are used to extract simplified models that capture the essential features of the high-fidelity models.[27] Order reduction is achieved by decomposing the flow field into a set of modes. Proper orthogonal decomposition provides these alternative modes in the form of orthogonal basis vectors derived from a set of system observations.[28] The resulting basis maximizes the average energy in the projection of the data onto the new subspace spanned by the modes, which minimizes the error associated with the projection.[29] Within fluid mechanics, the POD method was first applied to the study of turbulence,[30] but more recently, it has been extended to flow projection following basis creation,[31] sensitivity projection,[32,33] and even inverse design optimization.[34] A weakness of using POD to develop a ROM is that it typically requires the generation of several high-fidelity solutions. The generation of these solutions can be intractable.

In the current study, a novel ROM-based sensitivity convergence acceleration technique is presented. The technique is applied to a continuous adjoint solver where the basis vectors were formed from a sampling of the solution behavior by collecting snapshots of the adjoint solution at a set of preconvergent iterations within a single-solution procedure. Proper orthogonal decomposition was performed to construct an orthogonal basis, which was used to approximate the converged adjoint solution. Performance of the acceleration technique was evaluated by drops in the global residual following the projection. In addition, the technique is used in conjunction with traditional acceleration schemes including local time-stepping and multigrid for even greater acceleration.

It must be noted that since the adjoint equations are linear, the POD/ROM technique presented herein lends itself very nicely to extrapolation, or projection, to the converged solution. Similarly, the method could be readily extended to time-linearized unsteady flow solution acceleration due to the linear nature of the time-linearized governing equations. The presented acceleration technique is straightforward to implement within existing solvers, requiring only minor modifications to the code.

2 | GOVERNING EQUATIONS

The continuous adjoint sensitivity procedure is performed in 3 stages: First, the flow equations are solved; then, the flow solution is used to solve the adjoint equations; and finally, the resulting adjoint vector is used to calculate the sensitivities of interest. The mathematical derivation of each step will be outlined here.

2.1 | Flow equations: quasi–one-dimensional Euler equations

To present the adjoint equation acceleration technique, a flow regime must first be considered. Within this work, the quasi–one-dimensional Euler equations were used to model inviscid flow through a nozzle. The quasi-1D Euler equations were used because they are simple enough to have exact analytic solutions, while retaining complex flow features, such as shocks.[35] Although the method is presented for a quasi-1D problem, it is extendable to 2D and 3D computational fluid dynamics–based adjoint flow solvers, which is the focus of our ongoing research. The quasi-1D Euler equations are written as

$$\mathbf{R}\left(\mathbf{U}, h\right) = \frac{\mathrm{d}}{\mathrm{d}x}\left(h\mathbf{F}\right) - \mathbf{S}\frac{\mathrm{d}h}{\mathrm{d}x} = \mathbf{0}\ . \tag{1}$$

Here, h denotes the cross-sectional area, and the vector of conservation variables $\mathbf{U}$, the convective flux vector $\mathbf{F}$, and the source term $\mathbf{S}$ are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho H u \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix}\ .$$

The pressure and total enthalpy are related to the conservation variables through

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}u^2\right]$$
$$H = \frac{\rho E + p}{\rho}\ .$$

The governing equations were discretized on a structured grid using a dual control-volume method, and a central difference scheme with scalar artificial dissipation was used. A pseudo-time term was added to the steady governing equations, and the resulting equations were marched to steady state using an explicit scheme that can be written as

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{R}\left(\mathbf{U}^n\right)\ , \tag{2}$$

where $\mathbf{R}$ is the residual for a given iteration, n. Note that Equation 2 is left in a generic form to indicate that the proposed scheme is not limited to a given explicit scheme. However, in this work, a Jameson-Schmidt-Turkel -based 5-stage Runge-Kutta scheme was used with local time-stepping.[36,37]

2.2 | Adjoint equations for the quasi–one-dimensional Euler equations

The derivation of the continuous adjoint equations presented here closely follows the approaches of Lozano and Ponsin[10] and Giles and Pierce[35] and is included here for completeness. The derivation begins by defining the cost function, I. The cost function quantifies the quality of a given design and is generally defined so that either its minimization or maximization corresponds to an improved design. Following Lozano and Ponsin, the cost function is the integral of pressure along the nozzle,[10]

$$I = \int_{\Omega} p\ \mathrm{d}x\ , \tag{3}$$

where Ω represents the flow domain. This cost function is chosen, because it is analogous to the lift integral that is of considerable importance in aerodynamic design. From this definition, it should be observed that the cost function depends on the flow solution. Therefore, a perturbation in the flow solution, $\delta\mathbf{U}$, causes a change in the cost function, which is expressed by

$$\delta I = \int_{\Omega} \frac{\partial p}{\partial \mathbf{U}} \delta \mathbf{U} dx \ . \tag{4}$$

To calculate the change in the cost function, Equation 1 must be solved for each perturbation, $\delta\mathbf{U}$. Note that even for simple solvers such as the one developed in this work, this technique quickly becomes intractable if the number of design variables is large.

Therefore, the method of Lagrange multipliers is used to develop a technique to calculate the cost function variation, Equation 4, without the need to solve each independent perturbation. The adjoint state vector, $\boldsymbol{\psi}$, is introduced as a Lagrange multiplier to enforce the flow equations, Equation 1. This results in the augmented cost function, written as

$$I = \int_{\Omega} p \ \mathrm{d}x - \int_{\Omega} \boldsymbol{\psi}^T \mathbf{R}\left(\mathbf{U}, h\right) \ dx \ . \tag{5}$$

Note that the augmented cost function, Equation 5, is equivalent to Equation 3 because $\mathbf{R}\left(\mathbf{U}, h\right) = 0$ for a converged flow solution. Linearization of the new augmented cost function with respect to changes in the flow results in

$$\delta I = \int_{\Omega} \frac{\partial p}{\partial \mathbf{U}} \delta \mathbf{U} dx - \int_{\Omega} \boldsymbol{\psi}^T \left[\frac{\mathrm{d}}{\mathrm{d}x} \left(h \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} \right) - \frac{\mathrm{d}h}{\mathrm{d}x} \frac{\partial \mathbf{S}}{\partial \mathbf{U}} \delta \mathbf{U} - \frac{\mathrm{d}\delta h}{\mathrm{d}x} \mathbf{S} + \frac{\mathrm{d}}{\mathrm{d}x} \left(\delta h \mathbf{F} \right) \right] dx \ . \tag{6}$$

Integrating by parts and subsequently isolating the terms dependent on changes in the flow solution, $\delta\mathbf{U}$, yield

$$\begin{aligned} \delta I = & \int_{\Omega} \boldsymbol{\psi}^T \left[\frac{\mathrm{d}\delta h}{\mathrm{d}x} \mathbf{S} - \frac{\mathrm{d}}{\mathrm{d}x} \left(\delta h \mathbf{F} \right) \right] \mathrm{d}x \\ & - \int_{\Omega} \delta \mathbf{U}^T \left[-h \frac{\partial \mathbf{F}}{\partial \mathbf{U}}^T \frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}x} - \frac{\mathrm{d}h}{\mathrm{d}x} \frac{\partial \mathbf{S}}{\partial \mathbf{U}}^T \boldsymbol{\psi} - \frac{\partial p}{\partial \mathbf{U}}^T \right] \mathrm{d}x - \left[h \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} \right]_{x_i}^{x_o} \ . \end{aligned} \tag{7}$$

Here, x_i and x_o are the inlet and outlet locations, respectively. This equation is valid for any value of $\boldsymbol{\psi}$. As a result, Equation 7 can be simplified so that the variation of the cost function, δI, no longer depends on the variation of the flow solution, $\delta\mathbf{U}$, by choosing $\boldsymbol{\psi}$ so that

$$-h \frac{\partial \mathbf{F}}{\partial \mathbf{U}}^T \frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}x} - \frac{\mathrm{d}h}{\mathrm{d}x} \frac{\partial \mathbf{S}}{\partial \mathbf{U}}^T \boldsymbol{\psi} - \frac{\partial p}{\partial \mathbf{U}}^T = \mathbf{0} \tag{8}$$

and

$$\left[h \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} \right]_{x_i}^{x_o} = \mathbf{0} \ . \tag{9}$$

With these conditions satisfied, the variation of the cost function can be calculated, independent of $\delta\mathbf{U}$, through the remaining terms:

$$\delta I = \int_{\Omega} \boldsymbol{\psi}^T \left[\frac{\mathrm{d}\delta h}{\mathrm{d}x} \mathbf{S} - \frac{\mathrm{d}}{\mathrm{d}x} \left(\delta h \mathbf{F} \right) \right] \mathrm{d}x \ . \tag{10}$$

The adjoint equations, Equation 8 with boundary conditions defined by Equation 9, can be solved by pseudo time-marching for the appropriate adjoint vector, $\boldsymbol{\psi}$. Finally, the adjoint vector can be used to calculate the sensitivity of the cost function with respect to any design variable.

2.3 | Adjoint boundary conditions for the quasi–one-dimensional Euler equations

The mathematical formulation of appropriate boundary conditions for the adjoint equations is one of the main challenges associated with the continuous adjoint method. The inlet and outlet adjoint boundary conditions must be formulated so that Equation 9 is satisfied. In doing so, one must keep in mind that the adjoint characteristics are inversely related to the flow characteristics. For the quasi–one-dimensional Euler equations, if the flow equations have n incoming characteristics, then the adjoint equations have $(3 - n)$ incoming characteristics, and an equivalent number of boundary conditions must be specified.[38]

2.3.1 | Subsonic inlet boundary conditions

In the flow regime, a subsonic inlet has 2 incoming characteristics (corresponding to right-traveling pressure and entropy waves), with the eigenvalues $(u + c)$ and u, and one outgoing characteristic (corresponding to a left-traveling pressure wave), with the eigenvalue, $(u - c)$. As a result, the flow equations require 2 physical and one numerical boundary condition. One possibility is to specify the stagnation enthalpy, H, and the total pressure, p_t, while numerically extrapolating the Mach number, M, from the interior.

Lozano and Ponsin developed the inlet adjoint boundary conditions for these inlet flow variables, which are also used in this work and are provided here for completeness.[10] Recall that the adjoint boundary condition must satisfy Equation 9. Expanding the adjoint boundary condition in terms of the inlet flow variables (H, p_t, M) yields

$$\boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} = \boldsymbol{\psi}^T \left.\frac{\partial \mathbf{F}}{\partial H}\right|_{p_t, M} \delta H + \boldsymbol{\psi}^T \left.\frac{\partial \mathbf{F}}{\partial p_t}\right|_{H, M} \delta p_t + \boldsymbol{\psi}^T \left.\frac{\partial \mathbf{F}}{\partial M}\right|_{H, p_t} \delta M = \mathbf{0} \,. \tag{11}$$

Here, the change in total enthalphy, δH, and total pressure, δp_t, is both zero if no energy is added or removed from the system. This leaves only the term dependent on the change in Mach number, δM, which can be eliminated by imposing the adjoint boundary condition:

$$\boldsymbol{\psi}^T \left.\frac{\partial \mathbf{F}}{\partial M}\right|_{H, p_t} = \mathbf{0} \,. \tag{12}$$

The inlet boundary condition in the continuous adjoint solver is implemented by imposing the adjoint boundary condition given by Equation 12 and then extrapolating the 2 additional adjoint variables from the interior of the domain. Lozano and Ponsin[10] accomplish this by defining adjoint inlet variables, $\tilde{\boldsymbol{\psi}}_{in} = \left[\tilde{\psi}_M \ \tilde{\psi}_H \ \tilde{\psi}_{p_t}\right]^T$, as follows:

$$\begin{aligned}
\tilde{\psi}_M &= \frac{M\left(1 + \frac{\gamma-1}{2}M^2\right)}{\rho u \left(1 - M^2\right)} \left(\frac{\partial \mathbf{F}}{\partial M}\right)^T_{H, p_t} \boldsymbol{\psi} = \psi_1 + u\psi_2 + H\psi_3, \\
\tilde{\psi}_H &= \frac{2H}{\rho u} \left(\frac{\partial \mathbf{F}}{\partial H}\right)^T_{M, p_t} \boldsymbol{\psi} = -\psi_1 + H\psi_3, \\
\tilde{\psi}_{p_t} &= p_t \left(\frac{\partial \mathbf{F}}{\partial p_t}\right)^T_{H, M} \boldsymbol{\psi} = \rho u \psi_1 + \left(\rho u^2 + p\right)\psi_2 + \rho u H \psi_3 \,,
\end{aligned} \tag{13}$$

where $\tilde{\psi}_M = 0$, while $\tilde{\psi}_H$ and $\tilde{\psi}_{p_t}$ are extrapolated from the interior. Inversely, the adjoint variables at the inlet, $\boldsymbol{\psi}_{in} = [\psi_1 \ \psi_2 \ \psi_3]^T_{in}$, are related to $\tilde{\boldsymbol{\psi}}_{in}$ through

$$\boldsymbol{\psi}_{in} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}_{in} = \begin{bmatrix} \frac{\rho u^2 + p}{2p} & -\frac{1}{2} & -\frac{u}{2p} \\ -\frac{\rho u}{p} & 0 & \frac{1}{p} \\ \frac{\rho u^2 + p}{2pH} & \frac{1}{2H} & -\frac{u}{2pH} \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{\psi}_H^{(e)} \\ \tilde{\psi}_{p_t}^{(e)} \end{bmatrix} , \tag{14}$$

where the superscript (e) indicates extrapolated values from the interior.

2.3.2 | Subsonic outlet boundary conditions

A subsonic outlet in the flow domain has one incoming characteristic (left-traveling pressure), corresponding to the eigenvalue $(u - c)$, and 2 outgoing characteristics (right-traveling pressure and entropy), corresponding to the eigenvalues $(u + c)$ and u, respectively. As such, the flow equations require one physical and 2 numerical boundary conditions. One possibility is setting the exit pressure, p, and then extrapolating 2 other variables such as the density, ρ, and the velocity, u, numerically from the interior.

Lozano and Ponsin also present the derivation of the outlet adjoint boundary conditions for these outlet flow variables.[10] The outlet adjoint boundary conditions must be defined to enforce the exit condition of Equation 9, which can be expanded in terms of the exit flow variables (ρ, u, p) yielding

$$\boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} = \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \rho}\bigg|_{p,u} \delta\rho + \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial u}\bigg|_{p,\rho} \delta u + \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial p}\bigg|_{\rho,u} \delta p \ . \tag{15}$$

Note that δp is zero if the back pressure is held constant, which eliminates the dependence of the adjoint outlet boundary condition, Equation 15, on the pressure. To remove the dependence of the adjoint outlet boundary condition, Equation 15, on $\delta\rho$ and δu, the adjoint boundary conditions require

$$\begin{aligned} \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \rho}\bigg|_{p,u} &= \mathbf{0}, \\ \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial u}\bigg|_{p,\rho} &= \mathbf{0} \ . \end{aligned} \tag{16}$$

The outlet boundary condition in the continuous adjoint solver is thus implemented by imposing the adjoint boundary condition given by Equation 16 and then extrapolating the remaining independent adjoint variable from the interior of the domain. Thus, the adjoint outlet variables, $\tilde{\boldsymbol{\psi}}_{out} = \left[\tilde{\psi}_\rho \ \tilde{\psi}_u \ \tilde{\psi}_p\right]^T$, are

$$\tilde{\boldsymbol{\psi}}_{out} = \begin{bmatrix} \tilde{\psi}_\rho \\ \tilde{\psi}_u \\ \tilde{\psi}_p \end{bmatrix} = \begin{bmatrix} \frac{1}{u}\left(\frac{\partial \mathbf{F}}{\partial \rho}\right)^T \boldsymbol{\psi} \\ \frac{1}{\rho}\left(\frac{\partial \mathbf{F}}{\partial u}\right)^T \boldsymbol{\psi} \\ \left(\frac{\partial \mathbf{F}}{\partial p}\right)^T \boldsymbol{\psi} \end{bmatrix} = \begin{bmatrix} 1 & u & \frac{u^2}{2} \\ 1 & 2u & H + u^2 \\ 0 & 1 & \frac{\gamma}{\gamma-1} u \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} \ . \tag{17}$$

The adjoint exit boundary conditions impose that $\tilde{\psi}_\rho = 0$ and $\tilde{\psi}_u = 0$, while $\tilde{\psi}_p$ is extrapolated from the interior of the domain. The relation of these adjoint inlet variables can be inverted, which gives the final relation for the adjoint variables

$$\boldsymbol{\psi}_{out} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}_{out} = \frac{1}{\frac{\gamma+1}{\gamma-1}\frac{u^2}{2} - H} \begin{bmatrix} \frac{u^2(\gamma+1)-H(\gamma-1)}{\gamma-1} & -\frac{u^2}{2}\frac{\gamma+1}{\gamma-1} & uH \\ -\frac{\gamma u}{\gamma-1} & \frac{\gamma u}{\gamma-1} & -H - \frac{u^2}{2} \\ 1 & -1 & u \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \tilde{\psi}_p^{(e)} \end{bmatrix} \ . \tag{18}$$

2.3.3 | Supersonic outlet boundary conditions

In the flow regime, a supersonic outlet has no incoming characteristics and 3 outgoing characteristic, corresponding to the eigenvalues of u, $(u - c)$, and $(u + c)$. Therefore, the flow equations require 3 numerical boundary conditions, and the 3 exit flow variables, p, ρ, and u, are extrapolated numerically from the interior.

The outlet boundary conditions again must be defined to enforce the exit condition of Equation 9, which again can be expanded in terms of the exit flow variables (ρ, u, p) yielding

$$\boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U} = \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \rho}\bigg|_{p,u} \delta\rho + \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial u}\bigg|_{p,\rho} \delta u + \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial p}\bigg|_{\rho,u} \delta p \ . \tag{19}$$

For this condition to be satisfied, the adjoint boundary conditions require

$$\begin{aligned} \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial \rho}\bigg|_{p,u} &= \mathbf{0} \\ \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial u}\bigg|_{p,\rho} &= \mathbf{0} \quad . \\ \boldsymbol{\psi}^T \frac{\partial \mathbf{F}}{\partial p}\bigg|_{u,\rho} &= \mathbf{0} \end{aligned} \tag{20}$$

As a result, the adjoint variables for a supersonic outlet will all equal zero. Examining the adjoint outlet boundary condition given by Equations 17 and 18 reaffirms this according to

$$\boldsymbol{\psi}_{out} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}_{out} = \frac{1}{\frac{\gamma+1}{\gamma-1}\frac{u^2}{2} - H} \begin{bmatrix} \frac{u^2(\gamma+1)-H(\gamma-1)}{\gamma-1} & -\frac{u^2}{2}\frac{\gamma+1}{\gamma-1} & uH \\ -\frac{\gamma u}{\gamma-1} & \frac{\gamma u}{\gamma-1} & -H-\frac{u^2}{2} \\ 1 & -1 & u \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{21}$$

since $\tilde{\psi}_\rho$, $\tilde{\psi}_u$, and $\tilde{\psi}_p$ must all equal zero.

3 | ROM-BASED ADJOINT VECTOR CONVERGENCE ACCELERATION

For convenience, the system of adjoint equations, Equation 8, is solved iteratively by introducing a pseudo-time term:

$$\frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}\tau} - h\frac{\partial \mathbf{F}}{\partial \mathbf{U}}^T \frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}x} - \frac{\mathrm{d}h}{\mathrm{d}x}\frac{\partial \mathbf{S}}{\partial \mathbf{U}}^T \boldsymbol{\psi} - \frac{\partial p}{\partial \mathbf{U}}^T = \mathbf{0}\,. \tag{22}$$

For an explicit scheme, this iterative process can be written as

$$\boldsymbol{\psi}^{n+1} = \boldsymbol{\psi}^n + \mathbf{R}^{\mathrm{adj}}\left(\boldsymbol{\psi}^n\right)\,, \tag{23}$$

where n is the iteration number and $\mathbf{R}^{\mathrm{adj}}$ is the residual at each stage of the iteration process. Note that the adjoint equations are developed through linearization of the augmented equation, Equation 5. Therefore, the residual vector can be approximated as a linear function of the adjoint vector, ie,

$$\mathbf{R}^{\mathrm{adj}} \approx \mathbf{A}\boldsymbol{\psi} - \mathbf{b}\,, \tag{24}$$

which is solved by finding the adjoint state vector, $\boldsymbol{\psi}$, that drives the residual of the adjoint equation solver to zero. Here, $\mathbf{A}$ is the Jacobian matrix and $\mathbf{b}$ is the right hand side vector, which are functions of the adjoint equation. This is a full-order system with N degrees of freedom, equal to the number of grid points times the number of state variables.

Projection-based model order reduction methods can be used to project the high-order adjoint equations onto a subspace of the original space. This is equivalent to transforming the original high-order system of adjoint equations to a system of much lower order, by retaining only the most dominant portions of the solution dynamics.[39] The projection-based model effectively "compresses" the system's state information by projecting the state behavior onto a lower dimensional subspace thereby rewriting the governing equations in a compressed representation.[25]

To build a ROM, the dynamics or behavior of the iterative algorithm must be captured. The underlying behavior of the solution is extracted by samples of the solution at various pseudo-timesteps, called snapshots. These snapshots decompose the solution behavior into a set of modes.

3.1 | ROM development

For the ROM approach used in this work, M snapshots are collected and arranged in a matrix column by column, ie,

$$\Psi = \begin{bmatrix} | & | & | & & | \\ \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 & \boldsymbol{\psi}_3 & \cdots & \boldsymbol{\psi}_M \\ | & | & | & & | \end{bmatrix}_{N\times M}\,. \tag{25}$$

Proper orthogonal decomposition is then used to form an orthogonal basis from these snapshots. This is done by solving the eigenvector problem defined by

$$\Psi^T\Psi\mathbf{v}_k = \lambda_k\mathbf{v}_k, \quad \text{for} \quad k = 1, \ldots, M\,, \tag{26}$$

where λ_k and $\mathbf{v}_k$ are the eigenvalues and right eigenvectors of the symmetric matrix $\Psi^T\Psi$. Since the matrix is symmetric, each eigenvector corresponding to a unique eigenvalue will be orthogonal. These eigenvectors are then multiplied by the snapshot matrix to form an orthogonal basis so that

$$\phi_k = \Psi \mathbf{v}_k \ . \tag{27}$$

The new basis vectors are then collected column by column to form the matrix:

$$\Phi = \begin{bmatrix} | & | & | & & | \\ \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \boldsymbol{\phi}_3 & \dots & \boldsymbol{\phi}_M \\ | & | & | & & | \end{bmatrix}_{N \times M} \ . \tag{28}$$

Although POD is used here to create an optimal set of orthogonal basis vectors to capture the adjoint modes, Ekici and Hall state that it is also viable to use the nonorthogonal solution snapshots, Ψ, as alternative basis vectors by setting $\Phi = \Psi$ if a large number of snapshot vectors is used,[31] and prior work has shown this to be the case.[26,40]

3.2 | ROM convergence acceleration

Convergence acceleration is achieved by using a linear combination of the orthogonal basis vectors, consisting of the reduced adjoint modes, to approximate the converged adjoint solution

$$\boldsymbol{\psi}_{projected} = \sum_{i=1}^{M} \boldsymbol{\phi}_i \xi_i = \Phi \boldsymbol{\xi} \ , \tag{29}$$

so that

$$\mathbf{R}^{\text{adj}} \left(\boldsymbol{\psi}_{projected} \right) \approx \mathbf{0} \ . \tag{30}$$

This requires that the span of the basis vectors contains the fully converged (or at least a better) solution, ie,

$$\boldsymbol{\psi}|_{\mathbf{R}^{\text{adj}}(\boldsymbol{\psi})=0} \ \in \ \text{span}\,(\Phi) = \left\{ \sum_{i=1}^{M} \boldsymbol{\phi}_i \xi_i \ \mid \ \xi_i \in R \text{ and } \phi_i \in \Phi \right\} \ . \tag{31}$$

The projected solution may be found by substituting this ROM approximation from Equation 29 into the full-order governing equations, Equation 24, yielding

$$\mathbf{R}^{\text{adj}} = \mathbf{A}\Phi\boldsymbol{\xi} - \mathbf{b} \ , \tag{32}$$

which can be used to determine the appropriate weighting coefficients that drive the residual of the adjoint solver to machine accuracy, ie,

$$\mathbf{R}^{\text{adj}} = \mathbf{A}\Phi\boldsymbol{\xi} - \mathbf{b} = \mathbf{0} \ . \tag{33}$$

Note the above system is order N, which can be computationally demanding to solve as the problem size grows. To alleviate this concern, the system can be premultiplied by the transpose of Φ, thereby reducing the order to M, where ($M \ll N$). This results in

$$\Phi^T \mathbf{A}\Phi\boldsymbol{\xi} = \Phi^T \mathbf{b} \ . \tag{34}$$

The solution of Equation 34 begins by calculating the right hand side vector, $\mathbf{b}$, first. This is done by initializing the adjoint solution vector to zero and performing a single iteration. The residual of the adjoint solver can then be used to find $\mathbf{b}$ according to

$$\mathbf{R}^{\text{adj}} \left(\boldsymbol{\psi} = \mathbf{0} \right) = -\mathbf{b} \ . \tag{35}$$

Rather than explicitly calculating the Jacobian, $\mathbf{A}$, in Equation 34 directly, the matrix-free approach outlined by Ekici and Hall[31] is followed to compute $\mathbf{A}\Phi$ column by column. This is accomplished by initializing the adjoint solver with the first basis vector and calculating the resulting residual of the adjoint solver by running it for a single iteration. The matrix-vector product $\mathbf{A}\boldsymbol{\phi}_1$ is then obtained from

$$\mathbf{A}\boldsymbol{\phi}_1 = \mathbf{R}^{\text{adj}} \left(\boldsymbol{\phi}_1 \right) + \mathbf{b} \ . \tag{36}$$

The remaining matrix-vector products, $\mathbf{A}\boldsymbol{\phi}_2, \ \dots \ , \mathbf{A}\boldsymbol{\phi}_M$, are computed in a similar manner.

FIGURE 1 The cross-sectional area of the nozzle is defined by a sinusoidal decrease near the throat

With all other terms known, the weighting coefficients, ξ, can be determined by solving the ROM, Equation 34. The full-order projected adjoint vector can then be approximated through the linear combination given by Equation 29. As a result, the converged adjoint vector can be approximated from a number of early iteration unconverged adjoint vector values. It must be noted that, for each application of the acceleration technique, the additional computational cost of forming and solving the reduced-order system is on the order of M iterations. Therefore, this approach proves to be a very efficient way to accelerate the convergence of continuous adjoint solvers.

4 | RESULTS

The proposed ROM-based acceleration technique was applied to a continuous adjoint solver considering inviscid quasi–one-dimensional flow through a nozzle. The geometric definition of the nozzle is given by

$$h(x) = \begin{cases} 2 & 0 \leq x \leq 0.5 \\ 1 + \sin^2(\pi(x - 1.0)) & 0.5 < x < 1.5 \\ 2 & 1.5 \leq x \leq 2\,, \end{cases} \tag{37}$$

and it can be seen in Figure 1.

Three cases with increasingly complex flow fields were considered. These cases were chosen because each had been previously examined, thus enabling both internal and external verification of the accelerated results.[10,35,41] For each case, the adjoint vector, $\boldsymbol{\psi}^T$, was determined using both the traditional continuous adjoint method and the continuous adjoint method with the proposed acceleration technique.

4.1 | Case 1: fully subsonic nozzle

4.1.1 | Flow solution

To begin the sensitivity calculation procedure, the nozzle flow field solution must be determined. The first case consists of a nozzle with a fully subsonic flow field. The case is defined by a total inlet pressure of 1.0, a total inlet temperature of 1.0, and an exit pressure of 0.9899. This corresponds to the cases of Lozano[41] and Giles and Pierce[35] defined by an inlet Mach number of $M_{in} = 0.12$. The numeric flow solution was obtained using a dual control-volume discretization and a central scheme with scalar artificial dissipation on 299-node uniform grid. The Mach number distribution is shown to agree with the analytic flow solution[10] in Figure 2B. The resulting flow field is entirely subsonic and in the incompressible regime. As a result, the density-based compressible flow solver converges rather slowly as shown in Figure 2A.

4.1.2 | Traditional adjoint solution

Once the flow solution is obtained, the results can be used to solve for the adjoint vector using a continuous adjoint solver. A traditional (nonaccelerated) continuous adjoint solver was used to solve for the adjoint variables. The convergence rate

FIGURE 2 Fully subsonic nozzle solution A, convergence history and B, Mach number distribution [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 3 Adjoint solver A, convergence history and B, comparison of the adjoint solution vector with that of Lozano and Ponsin[10] [Colour figure can be viewed at wileyonlinelibrary.com]

of the adjoint solution was similar to that found for the flow solution and is included in Figure 3A. The adjoint vector is verified in Figure 3B using the results presented by Lozano and Ponsin.[10] As can be seen, the adjoint solution obtained using the traditional adjoint solver agrees well with Lozano and Ponsin.[10] With verification completed, the investigation turns to the evaluation of the proposed ROM convergence acceleration techniques.

4.1.3 | Convergence acceleration with snapshot basis vectors, $\Phi = \Psi$

The correlation-based acceleration, the simplest of the proposed techniques, uses the snapshots directly as the basis vectors to project the converged solution. Several parameters including snapshot quantity, first snapshot placement, iteration interval between snapshots, and snapshot span are used to control the convergence acceleration technique.

The acceleration technique's dependence on the snapshot quantity is evaluated first. To isolate the snapshot count as the sole variable of interest, the snapshots were collected over the same iteration span by varying the number of iterations between each snapshot for each case. This ensured that all the snapshots started and ended on the same iteration. Each time the snapshot count is halved, the acceleration technique effectively samples every other snapshot. The global residual convergence rates for acceleration using increasing snapshot quantities are presented in Figure 4.

Table 1 includes the snapshot collection parameters and the number of iterations to reach convergence, $[R(\boldsymbol{\psi}) = 1 \times 10^{-16}]$, for the cases considering increasing snapshot counts. Table 1 also presents the reduction of iterations required to reach convergence for each case. An idea of the cost associated with the ROM-based acceleration at different snapshot quantities can be observed by examining the correlation between the reduction in iterations and the time. As the snapshot count increases, the computational time reductions are diminished relative to the reduction in iteration count due to the time associated with the ROM-based convergence acceleration.

FIGURE 4 Convergence history achieved by varying snapshot count: Increased snapshot quantities improve convergence acceleration [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Iteration reduction for increasing snapshot count

Case	First snapshot (Iteration)	Snapshot span (Iterations)	Converge (Iteration)	Reduction (%)	Time (Normalized)
Baseline	-	-	31 175	-	1.0
11 snapshots	500	320	28 790	7.65	0.935
21 snapshots	500	320	26 361	15.44	0.859
41 snapshots	500	320	20 230	35.11	0.663
81 snapshots	500	320	14 656	52.99	0.499
161 snapshots	500	320	15 595	49.98	0.550

For this case, it is clear that collecting additional snapshots causes increasingly large global residual drops that correlate to convergence acceleration. This is likely due to the increased span of the solution space resulting from the additional degrees of freedom in Equation 31. It should be noted that all the acceleration cases reach machine accuracy before the traditional method. This occurs despite the fact that the 11-snapshot case residual actually jumps to a value higher than that found by the traditional method. The convergence rate increases after the jump so that even the 11-snapshot case is able to catch up to and eventually surpass the convergence of the traditional method. However, it is apparent that for the case considered here, 11 snapshots are insufficient to provide enough information to achieve a drop in the adjoint residual after the projection.

Next, a comparison of the projected adjoint vectors with the fully converged solution and the adjoint vector at the last snapshot is presented in Figure 5. It is clear that all of the projected adjoint solutions are closer to the fully converged solution; the 41-, 81-, and 161-snapshot cases are visually indistinguishable from the fully converged profile. Examining the 11-snapshot profile, it can be seen that the residual jump in the 11-snapshot case is produced by the steep gradient oscillation in the projected solution near $x = 0.25$. After the oscillations are attenuated, the profile matches the converged solution more closely that leads to an increased convergence rate. The difference of each projection from the converged value is also included in Figure 6 on a log scale, which demonstrates how accurately the projections match the converged solution.

Thus far, only a single application of the convergence acceleration technique has been considered. However, the proposed method can be applied multiple times during a solution procedure to further improve the convergence rate. The performance of repeated application of the acceleration technique is presented for each snapshot quantity using double precision in Figure 7A and quadruple precision in Figure 7B. Examining the convergence rates in Figure 7A for multiple applications of the acceleration, it can be seen that all but the 11-snapshot case reach a convergence limit at a global residual value between 1×10^{-8} and 1×10^{-10} using double-precision floating point numbers. The correlation-based acceleration technique is unable to project to a solution with improved accuracy at this point. This results from the fact that as the solution converges, the snapshot values will be increasingly similar and thus become less linearly independent making the ROM ill-conditioned. As a result, the potential solution space given by Equation 31 diminishes to a point where

FIGURE 5 Projected adjoint vector profiles offer considerable improvement over the adjoint vector profile of the most converged snapshot [Colour figure can be viewed at wileyonlinelibrary.com]

solution improvement is no longer attainable. However, if application of the acceleration technique is ceased once the residual convergence limit (1×10^{-10}) is reached, the solution will continue to converge until machine accuracy.

Next, to test our assertion that the method fails due to nearly linearly dependent snapshots, we run the solver using quadruple precision. Figure 7B shows that using quadruple precision eliminates the artificial convergence limit. However, it also increases the computational cost as well as the memory requirement by a factor of 2. The additional precision helps to distinguish between the snapshots as they become increasingly linearly dependent. However, even quadruple precision fails to add additional convergence acceleration for repeated application of the 11- and 21-snapshot cases with

FIGURE 6 The differences between the projected adjoint vector profiles and converged values show that the projections significantly improve the present solution [Colour figure can be viewed at wileyonlinelibrary.com]

these parameters, so stabilization of the method is desirable. While quadruple precision succeeds in reducing the total number of iterations for the 41-, 81-, and 161-snapshot cases, the upto fourfold cost increase of each iteration is untenable. Thus, another method is desired to make the acceleration tractable for convergence to machine accuracy.

4.1.4 | Convergence acceleration with orthogonal basis vectors

To circumvent problems with ill-conditioned ROMs, POD can be used to maximize the span of the solution space, Equation 31, by forming orthogonal basis vectors. The convergence acceleration for application of the correlation-based

FIGURE 7 Multiple applications of the convergence acceleration technique: A, Using double precision reaches an artificial convergence limit. B, Using quadruple precision eliminates this limit [Colour figure can be viewed at wileyonlinelibrary.com]

acceleration technique with orthogonal basis vectors is presented for cases using increasing quantities of snapshots in Figure 8A. The orthogonal basis vectors used here were formed from the same snapshots used for the convergence acceleration shown in Figure 4.

It can be seen that for a single application, the orthogonal basis vectors only result in a slight improvement in the convergence drop. The most notable drop occurs for the 161-snapshot case, which now outperforms the 81-snapshot case. The 161-snapshot case projection also converges past the artificial convergence limit found in the prior case using double-precision arithmetic.

Thus far, sampling of the first snapshot has been delayed until iteration 500. Figure 8B presents convergence acceleration for the same snapshot selection procedure but with snapshot sampling beginning at iteration 2 instead of iteration 500. Examining Figure 8, it is evident that delayed snapshot collection improves the performance of the acceleration technique. Table 2 presents the snapshot collection parameters and the relative acceleration improvement offered by delayed snapshot selection. Delaying snapshot collection offers considerable improvement, particularly for the cases with higher snapshot counts. Even this small delay achieves an additional cost reduction of up to 68%. By delaying snapshot collection, sampling of the initial transient behavior of the solver is avoided. This leads to less oscillatory basis vectors that leads to improved convergence acceleration. Delaying the first snapshot location even further conveys improved convergence acceleration until an eventual plateau is reached, as shown in Figure 9, for the 41-snapshot case.

While it is clear an optimal delay for the first snapshot location exists, all of the locations provide convergence acceleration. Additionally, for the best performance out of a single application of the convergence acceleration technique, it may be desirable to delay the application. However, it could be argued that earlier improvement in the global residual could actually be more significant.

FIGURE 8 Convergence acceleration using orthogonal basis vectors with snapshot collection. A, Beginning at iteration 500. B, Beginning at iteration 2 [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Acceleration performance for increasing snapshot counts

Case	First snapshot (Iteration)	Interval (Iterations)	Convergence (Iteration)	Reduction (%)	Relative reduction (%)
Baseline	-	-	31 175	-	-
11 snapshots	2	32	33 144	−6.32	-
11 snapshots	500	32	28 792	7.64	13.13
21 snapshots	2	16	27 239	12.14	-
21 snapshots	500	16	26 361	15.44	3.75
41 snapshots	2	8	26 296	15.65	-
41 snapshots	500	8	20 230	35.11	23.07
81 snapshots	2	4	27 666	11.26	-
81 snapshots	500	4	10 613	65.96	61.64
161 snapshots	2	2	25 403	18.51	-
161 snapshots	500	2	7946	74.51	68.72

The concern over when to begin snapshot sampling is largely alleviated by applying the convergence acceleration technique multiple times. Convergence acceleration achieved through multiple applications of the POD-based technique with snapshot sampling starting at iterations 2 and 500 is shown in Figure 10A and 10B, respectively.

Orthogonal basis vectors convey considerable improvement in convergence rates for cases that make use of repeated application of the acceleration technique. Again, the convergence limit found when using snapshot basis vectors is no longer present. Delaying sampling of the first snapshot still yields improved performance but to a much smaller degree,

FIGURE 9 Delaying the first iteration location for the 41-snapshot case improves projections until a gradual plateau is reached [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 10 Convergence acceleration using multiple applications of the proper orthogonal decomposition–based correlation technique with snapshot collection: A, Beginning at iteration 2. B, Beginning at iteration 500 [Colour figure can be viewed at wileyonlinelibrary.com]

and both repeated application cases significantly outperform even the very best single application case. The 81-snapshot delayed application case achieved the greatest speedup, reducing the total computational cost by 90%.

Only the 41-, 81-, and 161-snapshot cases were presented in Figure 10, because despite the orthogonal basis, the 11- and 21-snapshot cases defined using the same snapshot selection parameters were unsuccessful in delivering additional convergence acceleration. Multiple applications of the convergence acceleration technique using fewer snapshots result in

FIGURE 11 Convergence acceleration using multiple applications of the proper orthogonal decomposition–based correlation technique stabilized by increasing the delay between acceleration cycles for 21 snapshots [Colour figure can be viewed at wileyonlinelibrary.com]

unstable projections that can sometimes result in global residual drops but more frequently cause undesirable jumps. Two different techniques have been found to stabilize the performance of repeated application cases that use fewer snapshots. Both techniques were developed by recalling that the efficacy of a ROM is dependent upon how much of the solution behavior is captured.[25,42]

The first approach is performed by increasing the space between consecutive acceleration spans. Previously, a one-span delay between the last projection and the first snapshot of the following projection was used. An increased delay allows the solution to converge even further beyond the solution that was last projected. By sampling snapshots later, any transient behavior from oscillations in the previous projection has been dampened out, and new solution behavior is captured from the next cycle of snapshots. The stabilization provided by an increased delay is demonstrated for the 21-snapshot case in Figure 11. Looking closely, it appears that there is a trade-off between the frequency with which the acceleration technique is applied and the stabilization provided by increasing the delay between acceleration cycles. Performance improves from the increased stability through the 9-span delay. For larger delays, the increased delay leads to fewer applications of the acceleration technique that leads to less acceleration comparatively.

Occasionally, using this approach, the convergence rate jumps following the projection. This is likely due to small oscillations within the projected solution, like those shown in Figure 5. These oscillations occur because the limited basis vectors provide fewer degrees of freedom to completely smooth out the profile. Using more basis vectors eliminates these jumps as shown in Figure 12, where the 41-snapshot case is presented for comparison. However, it should be noted that despite the jumps, repeated application after the stabilization can outperform the single application case. The repeated application offers an improved slope following each projection. Examining the convergence history of the single application case, in Figure 4, the initially steep slope following the projection deteriorates as the solution progresses leading to a concave shape. The repeated application ensures that the steep convergence history slope is maintained. Finally, it is easy to evaluate a jump in the global residual, and if it is deemed too large, it is easy to reset the solution to the pre-projection value.

The second approach is the increased interval approach. Before, the iteration interval between snapshots was selected so that the span of case was kept constant. This was done merely for evaluative purposes. By increasing the span, the ROM can capture more of the solution behavior since the snapshots are collected over a longer pseudo-time. The improved stability conferred by increased iteration intervals for a 21-snapshot case is presented in Figure 13.

Again, there is a trade-off between the convergence acceleration application frequency and the stabilization provided by increasing the interval between snapshots, and the ideal snapshot interval for this case occurs at 32 iterations and achieves a 50% cost reduction. It is also notable that the increased interval approach slightly outperforms the increased delay approach. However, either approach can be used to stabilize multiple applications of convergence acceleration with fewer snapshots.

Lower snapshot counts are appealing because of the reduced memory requirements. For problems with more degrees of freedom, storing 161, 81, or even 41 snapshots within the random access memory (RAM) could become unfeasible. However, the acceleration technique can be performed using either an input-output (I/O)–efficient or a memory-efficient approach. In the I/O-efficient approach, the solution and their residuals are written into external files at each snapshot.

FIGURE 12 Convergence acceleration using multiple applications of the proper orthogonal decomposition–based acceleration technique delayed 13 spans [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 13 Convergence acceleration using multiple applications of the proper orthogonal decomposition–based correlation technique with 21 snapshots varying the iteration interval between snapshots [Colour figure can be viewed at wileyonlinelibrary.com]

After all of the snapshots have been collected, the snapshot solutions and residuals are all read into the RAM. This requires enough RAM to store $N \times (M + 1)$ values, where N is the number of degrees of freedom and M is the number of snapshots. This limits the number of I/O operations to only $2M$, since each snapshot and residual is written and then read only once. In the memory-efficient approach, the left and right hand sides of Equation 34 are built term by term by loading only 2 snapshots at a time, instead of loading all of the snapshots at once. This approach limits the memory requirements to $2N$ but requires $M^2 + 2M$ I/O operations. The additional I/O operations slow down the projection procedure but provide extension to problems with considerably more degrees of freedom.[26]

4.2 | Nozzle with subsonic inlet supersonic outlet

4.2.1 | Flow solution and adjoint vector verification

The second case consists of a nozzle with a subsonic inlet flow that is accelerated through the converging diverging nozzle so that the flow is supersonic at the outlet. The case is specified by a total inlet pressure of 1.0, a total inlet temperature of 1.0, and an exit pressure of 0.52. This case corresponds to the case of Giles and Pierce[35] designated by $H_{in} = 4$ and $p_{t_{in}} = 2$. The Mach number profile and adjoint solution using the traditional approach are compared with those of Giles and Pierce[35] in Figure 14A and 14B, respectively. In this case, a finer 899-node grid was used to show convergence acceleration for a case with a reduced convergence rate due to grid resolution. Inspecting the adjoint vector solution, it is interesting to note the asymptotic behavior at the nozzle throat. This is expected since a small change at the throat within a transonic flow field can have large ramifications.

FIGURE 14 Transonic nozzle A, Mach number distribution and B, adjoint vector solution both match the values reported by Giles and Pierce[35] [Colour figure can be viewed at wileyonlinelibrary.com]

4.2.2 | Convergence acceleration with orthogonal basis vectors

The correlation-based convergence acceleration technique using orthogonal basis vectors was applied to this transonic flow problem. This case was used to further examine the importance of the interval between snapshots. Application of the convergence acceleration using increasing spans of 80, 160, 320, and 640 iterations is shown in Figure 15. For these comparisons, the initial snapshot was taken at iteration 1000 to encourage stability among even the smallest span. As the span of the snapshots is increased, performance of the acceleration technique is improved. The convergence behavior for this case is smoother than the prior case that leads to large drops even for very small spans. The typical global residual drop is increased from 3 orders of magnitude to almost 10 by quadrupling the span. Overall, a single application of the acceleration technique can reduce the computation cost by over 50% for this case.

4.3 | Nozzle with shocked flow

4.3.1 | Flow solution and adjoint vector verification

The third case considers a nozzle with a transonic shocked flow field with a total inlet pressure of 1.0, a total inlet temperature of 1.0, and an exit pressure of 0.84317. This case is consistent with the $H_{in} = 4$, $p_{t_{in}} = 2$, and $p_{ex} = 1.6$ case presented by Giles and Pierce.[35] Comparisons of the Mach number profile, adjoint solution, and those reported by Giles and Pierce[35] are presented in Figure 16. The adjoint and flow solutions were obtained using a 1025-node grid.

The Mach number profile, shown in Figure 16A, clearly shows a shock in the flow field. Traditionally, shocks and other jump discontinuities can be difficult to resolve using a ROM. However, despite the shock in the flow field, the corresponding adjoint vector, Figure 16B, is actually quite similar to the prior subsonic-supersonic case. The adjoint vector

FIGURE 15 Convergence acceleration using orthogonal basis vectors with snapshot spans of A, 80 iterations; B, 160 iterations; C, 320 iterations; and D, 640 iterations [Colour figure can be viewed at wileyonlinelibrary.com]

field again behaves asymptotically near the throat but remains smooth and continuous at the shock location. This behavior is expected, and an in depth explanation is presented by Giles and Pierce.[43]

4.3.2 | Convergence acceleration with orthogonal basis vectors

As previously mentioned, the ROM-based acceleration scheme is easily implemented with existing convergence acceleration techniques. Here, performance of the correlation-based acceleration technique with orthogonal basis vectors applied to a solver accelerated with a multigrid scheme is evaluated using the shocked flow condition. To establish a baseline, a single application of the ROM acceleration technique is performed without multigrid, and the convergence histories are compared with those of the varying levels of multigrid schemes in Figure 17.

The cost reduction associated with each level of multigrid and the ROM convergence acceleration technique applied separately is included in Table 3. As found in the previous 2 cases, increasing the snapshot quantity again leads to larger global residual drops. It is interesting to note that there is no additional acceleration added by increasing the snapshot count from 21 to 41 snapshots. This is a known drawback of POD-based ROMs. Increased modes do not guarantee improved performance, and they can sometimes perform worse.[29] However, the general trend is still established by the improved performance offered by the 81- and 161-snapshot case. The 81-snapshot case performs the best in terms of computational time, and it can be observed that the computational time increases significantly for the 161-snapshot case even though the iteration counts are similar. The time required for storage of the snapshots and the solution of the basis vector coefficients increases with the number of snapshots, and as a result, lower snapshot counts perform more efficiently.

Figure 17 also demonstrates that both multigrid levels significantly reduce the number of iterations required to reach convergence, but the 1-level multigrid slightly outperforms the 2-level multigrid scheme when considering computational time as shown in Table 3. It should be noted that for this case, the multigrid scheme outperforms all the ROM-based acceleration applications except the 81-snapshot case, with respect to computational time.

An advantage of the proposed scheme is that it can easily be compounded with other acceleration schemes, and application of the ROM-based acceleration scheme to a solver using one level of multigrid is presented in Figure 18. A single

FIGURE 16 Shocked nozzle A, Mach number distribution and B, adjoint vector solution match those of Giles and Pierce[35] [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 17 Convergence acceleration using orthogonal basis vectors applied once [Colour figure can be viewed at wileyonlinelibrary.com]

application of the ROM-based convergence acceleration technique leads to a further reduction in required iterations and more importantly computational time for each snapshot quantity as shown in Table 4. The ROM-based acceleration applied to the multigrid scheme provides additional computational cost reduction relative to the multigrid scheme alone, and the 81-snapshot case almost halves the computational time.

TABLE 3 Iteration reduction for increasing snapshot count with a single application

Case	First snapshot (Iteration)	Snapshot span (Iterations)	Converge (Iteration)	Reduction (%)	Time (Normalized)
Baseline	-	-	11 267	-	1.0
Multigrid (1 level)	-	-	4755	57.80	0.674
Multigrid (2 levels)	-	-	3967	64.80	0.756
11 snapshots	1500	640	10 971	2.63	0.991
21 snapshots	1500	640	9514	15.56	0.883
41 snapshots	1500	640	9368	16.85	0.893
81 snapshots	1500	640	5899	47.65	0.659
161 snapshots	1500	640	6236	44.65	0.788

FIGURE 18 Convergence acceleration using orthogonal basis vectors applied once to the multigrid scheme [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Iteration reduction for increasing snapshot count with a single application applied to the 1-level multigrid scheme

Case	First snapshot (Iteration)	Snapshot span (Iterations)	Converge (Iteration)	Reduction (%)	Time (Normalized)
Baseline	-	-	11 267	-	1.0
Multigrid (1 level)	-	-	4755	57.80	0.674
Multigrid (2 levels)	-	-	3967	64.80	0.756
11 snapshots	500	320	4139	63.26	0.588
21 snapshots	500	320	3235	71.29	0.476
41 snapshots	500	320	3075	72.71	0.484
81 snapshots	500	320	1856	83.53	0.366
161 snapshots	500	320	2143	80.98	0.502

Note that for the multigrid case, the snapshot collection was started earlier and over a smaller span. This was done for 2 reasons. First, if application of the convergence technique was delayed until iteration 1500 on the multigrid scheme, the solution would have been nearly converged and thus not a fair assessment of the ROM-based acceleration technique. Secondly and more interestingly, the multigrid scheme actually appears to stabilize the performance of the ROM-based acceleration technique. Multigrid schemes remove low-frequency oscillation components from the solution behavior by solving the solution on a coarser grid, where they act as higher-frequency oscillations. When prolongation is performed to extend the coarse residuals to the fine grid, interpolation is required where data are not present. This interpolation leads to an inherent residual smoothing. The removal of the low-frequency oscillations coupled with the inherent residual smoothing allows the ROM-based acceleration technique to be applied earlier with smaller snapshot spans, because the residuals and basis vectors are less oscillatory.

FIGURE 19 Convergence acceleration using orthogonal basis vectors applied multiple times to the multigrid scheme [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Iteration reduction for increasing snapshot count with multiple applications applied to the 1-level multigrid scheme

Case	First snapshot (Iteration)	Snapshot span (Iterations)	Converge (Iteration)	Reduction (%)	Time (Normalized)
Baseline	-	-	11 267	-	1.0
Multigrid (1 level)	-	-	4755	57.80	0.674
Multigrid (2 levels)	-	-	3967	64.80	0.756
11 snapshots	500	320	2778	75.34	0.454
21 snapshots	500	320	2183	80.62	0.379
41 snapshots	500	320	2342	79.21	0.466
81 snapshots	500	320	1766	84.33	0.465
161 snapshots	500	320	2172	80.72	0.878

Finally, the ROM-based acceleration technique was applied multiple times to the multigrid scheme, and the convergence history is included in Figure 19 and Table 5. Multiple applications using 81 and 161 snapshots reduce the number of iterations required, but actually increase the computational time. From Figure 19, it is clear that after the first application, there is little gained from additional applications of the ROM-based acceleration technique. Additionally, the cost associated with the ROM-based acceleration technique increases with the number of snapshots used. As a result, it is recommended that application of the ROM-based scheme be halted after a global residual level around 1×10^{-12} when using large snapshot quantities in double precision, due to the diminishing returns.

While performance for the higher snapshot counts deteriorated when considering multiple applications, the performance of cases considering fewer snapshots improved. Smaller snapshot quantities require less RAM or storage space and are thus preferred to larger snapshot quantities. Multiple applications of the small-quantity snapshot cases to the multigrid scheme, in particular the 21-snapshot case, perform nearly as well as the best case and thus are a valuable alternative when memory is a concern.

5 | CONCLUSIONS

In this work, a ROM-based convergence acceleration technique is introduced for continuous adjoint solvers. A ROM is built by collecting snapshots of the solution behavior prior to convergence. This ROM is substituted into the full-order discretized governing equations to solve for the appropriate weighting coefficients. The weighting coefficients are then used to project to the "converged" solution. The acceleration technique was used to accelerate the adjoint equation for a quasi-1D problem. The method is also readily applicable to other linear flow problems including time-linearized flow solvers.

The acceleration technique was evaluated using both orthogonal and snapshot basis vectors. The orthogonal basis vectors were found to considerably improve the performance of the convergence acceleration technique, particularly for cases

with multiple applications. The roles of snapshot selection parameters including quantity, starting point, and span were then evaluated during application to test cases. The convergence acceleration technique was applied to 3 increasingly complex flows-an incompressible fully subsonic case, a subsonic to supersonic very fine grid case, and a transonic shocked flow case-to evaluate the efficacy of the acceleration when the convergence rate is limited by different causes. The convergence acceleration technique offered a considerable speedup for all of the cases.

The accelerated continuous adjoint solutions matched the solutions obtained using the traditional approach but required significantly fewer iterations. The projected solutions were also shown to offer considerable improvement over the most converged snapshot solution. A single application of the convergence technique can result in iteration reductions of up to 70%, while multiple applications can lead to over a 90% reduction. Furthermore, the acceleration technique was shown to work collectively with the multigrid and local time-stepping techniques and was able to reduce the cost to 0.37 times the original cost. Due to the efficacy and ease of implementation, the proposed convergence acceleration technique is found to be an effective approach to accelerate continuous adjoint solvers.

ACKNOWLEDGMENTS

This material is based upon work supported by the Mechanical, Aerospace, and Biomedical Engineering Department and the Bredesen Center for Interdisciplinary Research and Graduate Education at the University of Tennessee, Knoxville, and by the National Science Foundation under grant no. CBET-1150332. The authors greatly appreciate this support.

ORCID

Andrew L. Kaminsky http://orcid.org/0000-0001-8535-8455

REFERENCES

1. Jameson A. Aerodynamic design via control theory. *J Sci Comput*. 1988;3(3):233-260.
2. Huang H, Ekici K. A discrete adjoint harmonic balance method for turbomachinery shape optimization. *Aerosp Sci Technol*. 2014;39:481-490.
3. Kaminsky AL, Ekici K. Sensitivity and stability derivative analysis using an efficient adjoint harmonic balance technique. In: 54th AIAA Aerospace Sciences Meeting. San Diego, CA; 2016:1-23. AIAA-2016-0808.
4. Venditti DA, Darmofal DL. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. *J Comput Phys*. 2000;164(1):204-227.
5. Pierce NA, Giles MB. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Rev*. 2000;42(2):247-264.
6. Poon NM, Martins JR. An adaptive approach to constraint aggregation using adjoint sensitivity analysis. *Struct Multi Optim*. 2007;34(1):61-73.
7. Martins J, Poon NM. On structural optimization using constraint aggregation. In: VI World Congress on Structural and Multidisciplinary Optimization WCSMO6. Rio de Janeiro, Brasil; 2005:1-10 .
8. Luchini P, Bottaro A. Adjoint equations in stability analysis. *Annu Rev Fluid Mech*. 2014;46(1):493-517.
9. Kim J, Bewley TR. A linear systems approach to flow control. *Annu Rev Fluid Mech*. 2007;39:383-417.
10. Lozano C, Ponsin J. Remarks on the numerical solution of the adjoint quasi-one-dimensional Euler equations. *Int J Numer Methods Fluids*. 2012;69(5):966-982.
11. Hascoet L, Pascual V. The Tapenade automatic differentiation tool: principles, model, and specification. *ACM Trans Math Software (TOMS)*. 2013;39(3):1-43.
12. Giering R, Kaminski T, Slawig T. Generating efficient derivative code with TAF: adjoint and tangent linear Euler flow around an airfoil. *Future Gener Comput Syst*. 2005;21(8):1345-1355.
13. Mader CA, RA Martins J, Alonso J, Der Weide EV. ADjoint: an approach for the rapid development of discrete adjoint solvers. *AIAA J*. 2008;46(4):863-873.
14. Giles MB, Pierce NA. An introduction to the adjoint approach to design. *Flow Turbul Combust*. 2000;65(3-4):393-415.
15. Jameson A, Shankaran S, Martinelli L. Continuous adjoint method for unstructured grids. *AIAA J*. 2008;46(5):1226-1239.
16. Wang D, He L. Adjoint aerodynamic design optimization for blades in multistage turbomachines, part i: methodology and verification. *J Turbomach*. 2010;132(2):1-14.
17. Mavriplis DJ. Multigrid solution of the discrete adjoint for optimization problems on unstructured meshes. *AIAA J*. 2006;44(1):42-50.
18. Kim S, Alonso JJ, Jameson A. Multi-element high-lift configuration design optimization using viscous continuous adjoint method. *J Aircr*. 2004;41(5):1082-1097.
19. Reuther JJ, Jameson A, Alonso JJ, Rimlinger MJ, Saunders D. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *J Aircr*. 1999;36(1):51-60.
20. Wynn P. Acceleration techniques for iterated vector and matrix problems. *Math Comput*. 1962;16(79):301-322.

21. Cabay S, Jackson L. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM J Numer Anal*. 1976;13(5):734-752.
22. Jespersen DC, Buning PG. Accelerating an iterative process by explicit annihilation. *SIAM J Sci Stat Comput*. 1985;6(3):639-651.
23. Ekici K, Hall KC, Huang H, Thomas JP. Stabilization of explicit flow solvers using a proper-orthogonal-decomposition technique. *AIAA J*. 2013;51(5):1095-1104.
24. Clemens M, Wilke M, Schuhmann R, Weiland T. Subspace projection extrapolation scheme for transient field simulations. *IEEE Trans Magn*. 2004;40(2):934-937.
25. Markovinović R, Jansen J. Accelerating iterative solution methods using reduced-order models as solution predictors. *Int J Numer Methods Eng*. 2006;68(5):525-541.
26. Djeddi R, Kaminsky A, Ekici K. Convergence acceleration of fluid dynamics solvers using a reduced-order model. *AIAA J*. 2017;9:3059-3071.
27. Rowley CW, Dawson ST. Model reduction for flow analysis and control. *Annu Rev Fluid Mech*. 2017;49:387-417.
28. Lucia DJ, Beran PS, Silva WA. Reduced-order modeling: new approaches for computational physics. *Prog Aerosp Sci*. 2004;40:51-117.
29. Rowley CW. Model reduction for fluids, using balanced proper orthogonal decomposition. *Int J Bifurcation Chaos*. 2005;15(03):997-1013.
30. Lumley JL. The structure of inhomogeneous turbulent flows. *Atmos Turbul Radio Wave Propag*. 1967;790:166-178.
31. Ekici K, Hall KC. Fast estimation of unsteady flows in turbomachinery at multiple interblade phase angles. *AIAA J*. 2006;44(9):2136-2142.
32. Li J, Qu K, Cai J, Cao C. Adjoint approach based on reduced-order model for steady PDE systems. In: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Washington D.C.; 2016:1-11.
33. Kaminsky A, Djeddi R, Ekici K. An efficient reduced-order-model for accurate projection of discrete adjoint sensitivities. In: 55th AIAA Aerospace Sciences Meeting. Grapevine, TX; 2016:1-17. AIAA-2017-0037.
34. Bui-Thanh T, Damodaran M, Willcox KE. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J*. 2004;42(8):1505-1516.
35. Giles MB, Pierce NA. Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *J Fluid Mech*. 2001;426:327-345.
36. Djeddi R, Howison J, Ekici K. A fully coupled turbulent low-speed preconditioner for harmonic balance applications. *Aerosp Sci Technol*. 2016;53:22-37.
37. Jameson A, Schmidt W, Turkel E, et al. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA Pap*. 1981;1259:1-14.
38. Giles M, Pierce N. Adjoint equations in CFD-duality, boundary conditions and solution behaviour. 13th Computational Fluid Dynamics Conference. Snowmass Village, CO; 1997:182-198. AIAA-1997-1850.
39. Baur U, Benner P, Feng L. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Arch Comput Meth Eng*. 2014;21(4):331-358.
40. Djeddi R, Kaminsky AL, Ekici K. Convergence acceleration of fluid dynamics solvers using a reduced-order-model. In: 55th AIAA Aerospace Sciences Meeting. Grapevine, TX; 2017:1-16. AIAA-2017-0726.
41. Lozano C. A note on the dual consistency of the discrete adjoint quasi-one-dimensional Euler equations with cell-centered and cell-vertex central discretizations. *Comput Fluids*. 2016;134:51-60.
42. Bui-Thanh T, Willcox K, Ghattas O. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J Sci Comput*. 2008;30(6):3270-3288.
43. Giles MB, Pierce NA. On the properties of solutions of the adjoint Euler equations. *Numer Methods Fluid Dyn VI. ICFD*. 1998:1-16.

How to cite this article: Kaminsky AL, Djeddi R, Ekici K. Convergence acceleration of continuous adjoint solvers using a reduced-order model. *Int J Numer Meth Fluids*. 2018;86:582–606. https://doi.org/10.1002/fld.4468