AIAA JOURNAL
Vol. 57, No. 8, August 2019

Technical Notes

Flutter Boundary Identification from Time-Domain Simulations Using the Matrix Pencil Method

Jan F. Kiviaho,* Kevin E. Jacobson,* and Graeme J. Kennedy†
Georgia Institute of Technology, Atlanta, Georgia 30332

DOI: 10.2514/1.J058072

Nomenclature

c_k	=	complex amplitude in a Prony series
I_α	=	sectional moment of inertia
L	=	matrix pencil parameter
M	=	model order
$\boldsymbol{M}$	=	Prony series extraction matrix; $\boldsymbol{\Psi\Lambda\Phi}$
m	=	mass per unit span
N	=	number of interpolated samples
S_α	=	static unbalance
s_k	=	complex exponent in a Prony series
V_s	=	flutter speed index
$\boldsymbol{Y}$	=	matrix of interpolated sample values; $\boldsymbol{U\Sigma V}^T$
$\hat{\boldsymbol{y}}$	=	aeroelastic modal amplitude samples
$\boldsymbol{y}$	=	interpolated samples
α_k	=	damping of mode k in the Prony series
$\boldsymbol{\Lambda}$	=	diagonal matrix of eigenvalues/exponential terms
ρ	=	aggregation parameter
ω_k	=	frequency of mode k in the Prony series
$(\cdot)^+$	=	Moore–Penrose inverse
$\circ$	=	Hadamard componentwise product

I. Introduction

NUMERICAL flutter identification involves finding the dynamic pressure at which the minimum damping of an aeroelastic system is zero. Although frequency-domain methods can be more computationally efficient [1], time-domain flutter identification remains an important computational tool [2,3]. In this technical note, we formulate a minimum damping estimate based on the matrix pencil method and demonstrate how to compute its derivative. The matrix pencil method is a technique that can be used to robustly estimate the coefficients in a Prony series (a series of damped sinusoids) from a time-dependent signal [4,5]. These Prony series coefficients can then be used to evaluate an estimate of the minimum aeroelastic damping. However, a damping estimate alone is not sufficient to implement an efficient method to find the flutter boundary. The derivative of the damping with respect to input parameters, such as dynamic pressure, is required for Newton's method or for gradient-based aeroelastic optimization with damping constraints. Because the derivative of the damping estimate depends on the samples obtained from time history of an aeroelastic simulation, the proposed technique requires a fully coupled aeroelastic sensitivity method. Several research groups have developed high-fidelity aeroelastic tools that employ adjoint-based gradient evaluation techniques for both steady and unsteady aeroelastic problems [6–15] that meet this requirement. The proposed matrix pencil method is demonstrated on a pitch-plunge airfoil modeled using FUN3D within the recently developed FUNtoFEM framework for aeroelastic simulations [11,16,17].

Presented as Paper 2018-2932 at the 2018 Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, 25–29 June 2018; received 2 November 2018; revision received 7 May 2019; accepted for publication 9 May 2019; published online 10 June 2019.

*Graduate Research Assistant, School of Aerospace Engineering. Student Member AIAA
†Assistant Professor, School of Aerospace Engineering. Senior Member AIAA.

II. Minimum Damping Estimates via Prony Series

The matrix pencil method is a signal processing technique that is based on a Prony series approximation of a scalar signal. A Prony series of a time-varying signal sampled at evenly spaced intervals can be written as

$$y_n = \sum_{k=1}^{M} c_k e^{s_k n} + w_n, \qquad n = 1, \ldots, N \tag{1}$$

where the model order is M. In the present analysis, the samples y_n are known values extracted as the amplitude of a fixed structural mode from an aeroelastic simulation, and the complex amplitude and exponent, c_k and s_k, respectively, are to be determined to best represent the sampled values. The exponent $s_k = (-\alpha_k + i\omega_k)\Delta t$ contains both the damping α_k and the frequency ω_k of signal mode k as well as the uniform sampling interval Δt. The last term in the series, w_n, is the unknown signal noise.

To motivate the matrix pencil method, consider the Hankel matrix

$$\boldsymbol{A}_i = \begin{bmatrix} y_i & y_{i+1} & \cdots & y_{i+M} \\ y_{i+1} & y_{i+2} & \cdots & y_{i+M+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i+M} & y_{i+M+1} & \cdots & y_{i+2M} \end{bmatrix}$$

where the samples are arranged sequentially in each column of the matrix, with each new column starting from the next sample. Assuming the Prony series representation (1), the Hankel matrix can be written in the matrix form

$$\boldsymbol{A}_i = \boldsymbol{S}^T \boldsymbol{C} \boldsymbol{\Lambda}^i \boldsymbol{S} + \boldsymbol{W}_i$$

where $\boldsymbol{S}$ is an iteration-independent matrix containing the terms $e^{s_k n}$ defined as

$$\boldsymbol{S} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{s_1} & e^{s_2} & \cdots & e^{s_M} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_1(M-1)} & e^{s_2(M-1)} & \ddots & e^{s_M(M-1)} \end{bmatrix}$$

$\boldsymbol{C}$ is a diagonal matrix containing the coefficients c_k

$$\boldsymbol{C} = \mathrm{diag}(c_1, c_2, \ldots, c_M)$$

and the matrix $\boldsymbol{\Lambda}$ is diagonal and contains the exponential terms

$$\boldsymbol{\Lambda} = \mathrm{diag}(e^{s_1}, e^{s_2}, \ldots, e^{s_M})$$

While the function of the matrix pencil method is to reconstruct a given signal as a sum of M damped sinusoids, our primary interest is in the damping, α_k, of the signal modes, which can be extracted from the exponent s_k.

In the absence of noise, such that $\boldsymbol{W}_i = 0$, and assuming that $\boldsymbol{A}_1$ is invertible, we can compute the matrix $\boldsymbol{M} \in \mathbb{R}^{M\times M}$

$$\boldsymbol{M} \triangleq \boldsymbol{A}_1^{-1}\boldsymbol{A}_2 = \boldsymbol{S}^{-1}\boldsymbol{\Lambda}\boldsymbol{S} \tag{2}$$

Since the right-hand side is a similarity transformation, the entries of $\boldsymbol{\Lambda}$ are the eigenvalues of the matrix $\boldsymbol{M}$, which can be formed from the samples. By taking the natural logarithm of the kth eigenvalue and dividing by the sampling interval, we obtain the complex number $-\alpha_k + j\omega_k$, in which the damping is the real part

$$\alpha_k = -\frac{Re[\ell_n(e^{s_k})]}{\Delta t} \tag{3}$$

Unfortunately, it is not possible to neglect the presence of the noise term and obtain robust damping estimates, even for signals obtained from numerical simulations. As a result, a more sophisticated approach is required.

The matrix pencil method offers a robust approach that is less sensitive to numerical noise. In our implementation, the damping estimation method differs from the simple technique outlined above in three key ways. First, the number of terms, M, in the Prony series is selected based on a criterion on the singular values. Second, the matrix $\boldsymbol{M}$ is computed indirectly via a singular value decomposition (SVD) of a Hankel-like matrix while discarding low-energy components corresponding to noise. Third, a damping aggregate is used to conservatively approximate the minimum damping from among all modes using the Kreisselmeier–Steinhauser (KS) function [18,19].

In this technique, we first compute the evenly spaced samples from an interpolation of the true aeroelastic data to avoid numerical issues with large sample sets. This interpolation step is written as

$$\boldsymbol{y} = \boldsymbol{H}\hat{\boldsymbol{y}} \tag{4}$$

where $\hat{\boldsymbol{y}}$ are obtained from the structural components of the aeroelastic solution state vector as $\hat{y}_i = \boldsymbol{u}_i^T\boldsymbol{u}_{\text{mode}}$, where $\boldsymbol{u}_i$ are the structural state variables at aeroelastic step i, and $\boldsymbol{u}_{\text{mode}}$ is a fixed structural mode vector. Note that this linear extraction technique facilitates modular integration with an aeroelastic adjoint method. With the interpolated samples, $\boldsymbol{y}$, a truncated Hankel matrix $\boldsymbol{Y} \in \mathbb{R}^{(N-L)\times(L+1)}$ is formed

$$\boldsymbol{Y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_L & y_{L+1} \\ y_2 & y_3 & \cdots & y_{L+1} & y_{L+2} \\ \vdots & \vdots & \ddots & \vdots & \\ y_{N-L} & y_{N-L+1} & \cdots & y_{N-1} & y_N \end{bmatrix}$$

Here, L is the matrix pencil parameter and is typically chosen between a value of $N/2$ and $N/3$ [4]. In this work we select $L = N/2–1$ for all results. Next, we compute the SVD of the matrix $\boldsymbol{Y}$, such that

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

The number of terms, M, in the Prony series is selected by imposing a relative tolerance on the retained singular values such that $\sigma_i/\max_i \sigma_i < 0.01$. The first M right singular vectors are kept and placed in the new matrix $\hat{\boldsymbol{V}} \in \mathbb{R}^{(L+1)\times M}$, while the remaining singular values and singular vectors are discarded. Next, two matrices, $\hat{\boldsymbol{V}}_1$ and $\hat{\boldsymbol{V}}_2$, are extracted from $\hat{\boldsymbol{V}}$. The matrix $\hat{\boldsymbol{V}}_1 \in \mathbb{R}^{L\times M}$ is extracted from the first L rows of $\hat{\boldsymbol{V}}$, and $\hat{\boldsymbol{V}}_2 \in \mathbb{R}^{L\times M}$ is extracted from the last L rows of $\hat{\boldsymbol{V}}$. With these new matrices, the noise-tolerant matrix $\boldsymbol{M}$ is computed as

$$\boldsymbol{M} = \hat{\boldsymbol{V}}_2^T[\hat{\boldsymbol{V}}_1^T]^+ \tag{5}$$

where $[\hat{\boldsymbol{V}}_1^T]^+ = \hat{\boldsymbol{V}}_1(\hat{\boldsymbol{V}}_1^T\hat{\boldsymbol{V}}_1)^{-1}$ denotes the right pseudoinverse of $\hat{\boldsymbol{V}}_1^T$. This pseudoinverse is well defined since the columns of $\hat{\boldsymbol{V}}_1$ are linearly independent. Note that this pseudoinverse appears on the right-hand side in the filtered case (5), whereas the inverse of $\boldsymbol{A}_1$ appears on the left-hand side in noise-sensitive simple method (2). It can be shown that pre- or postmultiplying $\hat{\boldsymbol{V}}_2^T$ by the pseudoinverse $[\hat{\boldsymbol{V}}_1^T]^+$ produces matrices with the same nonzero eigenspectrum. Using the postmultiplication matrix results in a smaller $M \times M$ matrix with only the relevant M eigenvalues. Next, the eigenvalues and left and right eigenvectors of the matrix $\boldsymbol{M}$ are computed:

$$\boldsymbol{M} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Phi}$$

where $\boldsymbol{\Lambda} = \text{diag}\{\boldsymbol{\lambda}\}$. Once the eigenvalue decomposition of $\boldsymbol{M}$ is computed, the damping estimates are obtained using Eq. (3) and the damping aggregate is computed with the KS function

$$c(\boldsymbol{\alpha}, \rho) = \alpha_{\min} - \frac{1}{\rho}\ell_n\left[\sum_{k=1}^{M} e^{-\rho(\alpha_k - \alpha_{\min})}\right]$$

where $\boldsymbol{\alpha}$ is a vector of the damping of the modes, $\alpha_{\min} = \min(\boldsymbol{\alpha})$ is the minimum damping, and ρ is the KS parameter that controls aggregation error.

This variant of the matrix pencil method offers two primary advantages over similar techniques, such as polynomial-based Prony methods. The first advantage is that the filtering of noise is embedded within the algorithm, whereas other similar techniques require that filtering be performed as a preprocessing step, adding complexity and compromising robustness. The second advantage of the method is that it finds the exponents using the solution of an eigenvalue problem rather than through a root-finding method, making this variant of the matrix pencil method faster and more robust.

III. Derivative of the Damping Estimate

To use the damping estimate developed in a Newton method or as a constraint within a gradient-based optimization problem, we must take the derivative of the estimate with respect to the input data, $\partial c/\partial \hat{\boldsymbol{y}}$. The aeroelastic adjoint method will then take this derivative as input to evaluate the gradient of the damping with respect to any desired parameters. The step-by-step process for computing this derivative is outlined notionally in Algorithm 1, with specific equations for each step given in the following section.

The method presented in Algorithm 1 is an application of the chain rule where the forward analysis process is reversed step by step. The first step at line 2 of Algorithm 1 evaluates the derivative of the KS function with respect to the modal damping estimates. The kth component of this derivative is

$$\frac{\partial c}{\partial \alpha_k} = \frac{e^{-\rho(\alpha_k - \alpha_{\min})}}{a} \tag{6}$$

where $a \triangleq \sum_{k=1}^{M} e^{-\rho(\alpha_k - \alpha_{\min})}$.

Line 3 of Algorithm 1 multiplies the term $\partial c/\partial \boldsymbol{\alpha}$ by the derivative of $\boldsymbol{\alpha}$ with respect to the eigenvalues $\boldsymbol{\lambda}$. The complex eigenvalue can be written in the polar form $\lambda = re^{j\theta}$, where the modulus is $r = (\lambda_r^2 + \lambda_c^2)^{1/2}$ with the real and complex parts of the eigenvalue, λ_r and λ_c, respectively. Because of the properties of the complex logarithm, $\ell_n(\lambda) = \ell_n(r) + j\theta$, and therefore

$$Re[\ell_n(\lambda)] = \frac{1}{2}\ell_n(\lambda_r^2 + \lambda_c^2)$$

With this expression, the damping (3), can now be expressed as

$$\alpha = -\frac{1}{2\Delta t}\ell_n(\lambda_r^2 + \lambda_c^2)$$

Algorithm 1 Derivative of matrix pencil-based method

1:	**Given**: Δt, ρ, $\boldsymbol{H}$, $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}^T$, $\hat{\boldsymbol{V}}_1^T$, $\hat{\boldsymbol{V}}_2^T$, $[\hat{\boldsymbol{V}}_1^T]^+$, $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}$
2:	Compute $\partial c/\partial \boldsymbol{\alpha}$ ▷Derivative of the KS aggregation of damping (6)
3:	$[\partial c/\partial \boldsymbol{\lambda}] \leftarrow [\partial c/\partial \boldsymbol{\alpha}][\partial \boldsymbol{\alpha}/\partial \boldsymbol{\lambda}]$ ▷Contribution from the damping extraction (7)
4:	$[\partial c/\partial \boldsymbol{M}] \leftarrow [\partial c/\partial \boldsymbol{\lambda}][\partial \boldsymbol{\lambda}/\partial \boldsymbol{M}]$ ▷Derivative of the eigenvalue decomposition of $\boldsymbol{M}$ (8)
5:	$[\partial c/\partial \hat{\boldsymbol{V}}_1^T] \leftarrow [\partial c/\partial \boldsymbol{M}][\partial \boldsymbol{M}/\partial \hat{\boldsymbol{V}}_1^T]$ ▷Contribution to $\boldsymbol{M}$ by $\hat{\boldsymbol{V}}_1$ (9)
6:	$[\partial c/\partial \hat{\boldsymbol{V}}_2^T] \leftarrow [\partial c/\partial \boldsymbol{M}][\partial \boldsymbol{M}/\partial \hat{\boldsymbol{V}}_2^T]$ ▷Contribution from the pseudoinverse (9) and (10)
7:	$[\partial c/\partial \hat{\boldsymbol{V}}^T] \leftarrow [\partial c/\partial \hat{\boldsymbol{V}}_1^T][\partial \hat{\boldsymbol{V}}_1^T/\partial \hat{\boldsymbol{V}}^T] + [\partial c/\partial \hat{\boldsymbol{V}}_2^T][\partial \hat{\boldsymbol{V}}_2^T/\partial \hat{\boldsymbol{V}}^T]$ ▷Combine derivatives of $\hat{\boldsymbol{V}}_1$ and $\hat{\boldsymbol{V}}_2$
8:	$[\partial c/\partial \boldsymbol{Y}] \leftarrow [\partial c/\partial \hat{\boldsymbol{V}}^T][\partial \hat{\boldsymbol{V}}^T/\partial \boldsymbol{Y}]$ ▷SVD derivative (11)
9:	$[\partial c/\partial \boldsymbol{y}] \leftarrow [\partial c/\partial \boldsymbol{Y}][\partial \boldsymbol{Y}/\partial \boldsymbol{y}]$ ▷Derivative of the Hankel matrix construction Algorithm 2
10:	$[\partial c/\partial \hat{\boldsymbol{y}}] \leftarrow [\partial c/\partial \boldsymbol{y}]\boldsymbol{H}$ ▷Derivative of the interpolation to uniform samples

Now, the derivative of real-valued α with respect to complex-valued λ can be taken by treating the derivatives with respect to the real part λ_r and the complex part λ_c separately such that

$$\frac{\partial \alpha}{\partial \lambda_r} = -\frac{1}{\Delta t}\left[\frac{\lambda_r}{\lambda_r^2 + \lambda_c^2}\right], \quad \frac{\partial \alpha}{\partial \lambda_c} = -\frac{1}{\Delta t}\left[\frac{\lambda_c}{\lambda_r^2 + \lambda_c^2}\right] \tag{7}$$

The next contribution in line 4 of Algorithm 1 comes from the eigenvalue decomposition of the matrix $\boldsymbol{M}$. Given the left and right eigenvectors, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \quad \boldsymbol{\psi}_2 \quad \dots \quad \boldsymbol{\psi}_M]$ and $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \quad \boldsymbol{\phi}_2 \quad \dots \quad \boldsymbol{\phi}_M]$, that are biorthogonal, the derivative for the kth eigenvalue is

$$\frac{\partial \lambda_k}{\partial \boldsymbol{M}} = \frac{\bar{\boldsymbol{\psi}}_k \boldsymbol{\phi}_k^T}{\bar{\boldsymbol{\psi}}_k^T \boldsymbol{\phi}_k} \tag{8}$$

where $\bar{\boldsymbol{\psi}}$ denotes the complex conjugate of the left eigenvector $\boldsymbol{\psi}$. This derivative is complex-valued. In this context, the real and complex contributions to the derivative are $\partial \lambda_r/\partial \boldsymbol{M}$ and $\partial \lambda_c/\partial \boldsymbol{M}$, respectively. These contributions are combined using Eq. (7).

Line 5 of Algorithm 1 arises from the dependence of the matrix $\boldsymbol{M}$ on the filtered singular vectors $\hat{\boldsymbol{V}}_1^T$. This derivative requires two steps. The first step involves the evaluation of the derivative with respect to the pseudoinverse matrix $[\boldsymbol{V}_1^T]^+$, from the expression for $\boldsymbol{M}$ (5) as follows:

$$\frac{\partial c}{\partial [\boldsymbol{V}_2^T]^+} = \boldsymbol{V}_2 \frac{\partial c}{\partial \boldsymbol{M}} \tag{9}$$

The second step comes from the derivative of the right pseudoinverse $[\hat{\boldsymbol{V}}_1^T]^+$ with respect to matrix $\hat{\boldsymbol{V}}_1^T$. In this case, using $\boldsymbol{A}$ and $\boldsymbol{A}^+$ to denote $\hat{\boldsymbol{V}}_1^T$ and its pseudoinverse, respectively, the derivative can be written as

$$\frac{\partial c}{\partial \boldsymbol{A}} = (\boldsymbol{A}\boldsymbol{A}^T)^{-1}\frac{\partial c}{\partial \boldsymbol{A}^+}^T (\boldsymbol{I} - \boldsymbol{A}^+\boldsymbol{A}) - [\boldsymbol{A}^+]^T \frac{\partial c}{\partial \boldsymbol{A}^+}[\boldsymbol{A}^+]^T \tag{10}$$

Details on the derivative of pseudoinverse matrices are given in Golub and Pereyra [20].

Next, line 6 of Algorithm 1 multiplies the term $\partial c/\partial \boldsymbol{M}$ by the derivative of the matrix $\boldsymbol{M}$ with respect to the matrix of filtered singular vectors $\hat{\boldsymbol{V}}_2^T$. This derivative is a simple application of matrix calculus to the formulation of $\boldsymbol{M}$. The derivative contribution in line 7 of Algorithm 1 combines $\partial c/\partial \hat{\boldsymbol{V}}_1^T$ and $\partial c/\partial \hat{\boldsymbol{V}}_2^T$ into the derivative $\partial c/\partial \hat{\boldsymbol{V}}^T$.

The derivative of the SVD computation, in line 8 of the Algorithm 1, adds the contribution from the term $\partial c/\partial \hat{\boldsymbol{V}}^T$ from the derivative of the matrix $\hat{\boldsymbol{V}}^T$ with respect to the matrix $\boldsymbol{Y}$. This derivative must be handled with care. Here, we use the reverse mode approach outlined by Giles [21]. Further details can be found in the work of Papadopoulo and Lourakis [22]. Assuming that the singular values are distinct, we first evaluate a skew symmetric matrix $\boldsymbol{F}$, whose components are $F_{ij} = (\sigma_j^2 - \sigma_i^2)^{-1}$ for $i \neq j$, and $F_{ii} = 0$ along the diagonal. In addition, we define the square diagonal matrix of singular values $\boldsymbol{S}_1 = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_{L+1}\}$, and the matrix $\boldsymbol{E}_1 = [\boldsymbol{I}_1 \quad 0] \in \mathbb{R}^{(L+1)\times(N-L)}$, and compute the derivative as

$$\frac{\partial c}{\partial \boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{E}_1^T\boldsymbol{S}_1\left(\boldsymbol{F} \circ \left(\boldsymbol{V}^T\frac{\partial c}{\partial \boldsymbol{V}} - \frac{\partial c}{\partial \boldsymbol{V}}^T\boldsymbol{V}\right)\right)\boldsymbol{V}^T \tag{11}$$

where $\circ$ denotes the Hadamard componentwise product.

Next, the construction of the matrix $\boldsymbol{Y}$ contributes an accumulation of the derivative $\partial c/\partial \boldsymbol{Y}$ into the derivative of the damping with respect to the samples, $\partial c/\partial \boldsymbol{y}$, shown in line 9 of Algorithm 1. Because of the structure of the matrix $\boldsymbol{Y}$, this contribution simply involves summing the antidiagonals of $\partial c/\partial \boldsymbol{Y}$ as shown in Algorithm 2.

Finally, the sampling interpolation (4) is accounted for by multiplying the $\partial c/\partial \boldsymbol{y}$ by the derivative of the samples, $\boldsymbol{y}$, with respect to the full sample set, $\hat{\boldsymbol{y}}$. This derivative is the interpolation matrix $\boldsymbol{H}$, as is shown in line 10 of Algorithm 1. This final step completes the computation of the desired derivative, $\partial c/\partial \hat{\boldsymbol{y}}$.

IV. Derivative Verification and Adjoint Demonstration

To demonstrate that the derivative described above is computed correctly, we take the function

$$z(t) = 0.5e^t \sin(8\pi t) + 10e^{-t}\sin(2\pi t)$$

and evaluate at the points $\hat{y}_i = z(\Delta t(i-1))$ for 201 initial points, with $\Delta t = 3/200$. We compute an interpolation to $N = 150$ sample points, use a KS aggregation parameter of $\rho = 100$, and apply the damping estimation procedures outlined above to obtain a damping aggregate value of 1.00702. Note that this value overestimates the minimum damping value of $\alpha_{\min} = 1$. Next, we compute the directional derivative of the minimum damping along a randomly generated direction, $\boldsymbol{p}_{\hat{y}}$. To evaluate this directional derivative by finite difference, we compute

$$\frac{\partial c}{\partial \hat{\boldsymbol{y}}}\ \boldsymbol{p}_{\hat{y}} \approx \frac{c(\hat{\boldsymbol{y}} + h\boldsymbol{p}_{\hat{y}}) - c(\hat{\boldsymbol{y}} - h\boldsymbol{p}_{\hat{y}})}{2h}$$

where $h = 1 \times 10^{-6}$. Table 1 shows a comparison between the values of the directional derivative obtained using the finite-difference method and the proposed analytic method. The derivatives match to six digits with a relative error of 7.9×10^{-8}. Note that we were not able to use complex step approximation to verify the derivative due to the use of the SVD, and eigenvalue problem within the forward algorithm.

Algorithm 2 Evaluation of derivative $\partial c/\partial \boldsymbol{y}$

```
∂c/∂y = 0
for i = 1 → N − L do
    for j = 1 → L + 1 do
        ∂c/∂y_{i+j} ← ∂c/∂y_{i+j} + ∂c/∂Y_{i,j}
    end for
end for.
```

Table 1 Comparison of directional derivative with finite differences

Evaluation method	$(\partial c/\partial \hat{y})\ \boldsymbol{p}_{\hat{y}}$
Finite difference	−0.0**327942**774
Analytic	−0.0**327942**800
Relative error	7.9×10^{-8}

Table 2 Damping estimate gradient evaluation for single-degree-of-freedom system

ζ	ω	$d\hat{\alpha}/d\zeta$
−0.125	16.0	15.996085797
0.125	16.0	15.996683413

As a second check, we compute the derivative of the damping in a spring-mass-damper system with respect to the damping factor, ζ. For this test, the governing equation takes the form

$$\ddot{u} + 2\zeta\omega\dot{u} + \omega^2 u = 0$$

and the damping is $\alpha = \omega\zeta$. The initial conditions are set as $u(0) = 1$ and $\dot{u}(0) = 0$. As a result, the derivative of the damping estimate should give

$$\frac{d\hat{\alpha}}{d\zeta} \approx \omega \tag{12}$$

We run the simulation with $\omega = 16$, and values of the damping factor of $\zeta = 0.125$ and $\zeta = -0.125$. We perform the simulation using an explicit Runge–Kutta 4–5 adaptive integrator with a target error or 10^{-8}, which takes 1502 time steps over a 1.5 s simulation interval. We interpolate the data to $N = 250$ sample points. The results of the computation of these gradients are shown in Table 2. Although an exact match is not expected due to the damping aggregation method, the gradient value is very close to the natural frequency, as predicted by Eq. (12). This raises our confidence in the implementation of the derivative and also gives us a positive indication of how it will perform in conjunction with the adjoint method.

V. Flutter Identification Demonstration

In this section, we demonstrate the matrix pencil method by finding a portion of the flutter boundary for a pitch-plunge aeroelastic model of the NACA 64A010 airfoil [23,24] for Mach numbers between 0.7 and 0.9. The parameters selected for this study correspond to the aeroelastic model first studied by Isogai [23], and subsequently examined numerically by several authors [25–28].

The flutter boundary is located by solving a series of constrained optimization problems, at fixed Mach number, where the objective is to minimize free stream velocity subject to an equality constraint that the aeroelastic damping predicted by the matrix pencil method must equal zero. In this work, we solve these optimization problems using the SLSQP [29] optimizer through the pyOpt optimization interface [30]. In this study, the aeroelastic simulation and adjoint-based gradient evaluation are performed using the FUNtoFEM aeroelastic framework [11,16], which uses FUN3D [31] for aerodynamic simulation and includes a time-dependent aeroelastic adjoint implementation [32].

The time domain aerodynamic simulation uses the optimized second-order backward difference formula [33] in FUN3D to integrate the Euler equations in time. A dual time stepping scheme, with 15 subiterations per time step, is employed for all simulations, which typically results in four to six orders of magnitude of convergence at each time step. The nondimensional aerodynamic time step

$$\Delta t^* = \frac{\Delta t a_\infty}{2b} \tag{13}$$

is held fixed at a value of $\Delta t^* = 0.1$, throughout all computations, where b is the aerodynamic semichord, and a_∞ the free stream speed of sound. Each aeroelastic simulation is run for 2000 time steps. Because the adjoint method in FUN3D is only implemented for three-dimensional simulations, the aerodynamic mesh of the NACA 64A010 airfoil shown in Fig. 1 contains three identical but offset planes, each with 25,119 nodes for a total of 75,357 nodes.

The structural model consists of a two-degree-of-freedom system, illustrated in Fig. 2, with a translational plunge spring and a rotational spring attached at the elastic axis. As formulated by Bisplinghoff et al. [34], the structure is governed by the following system of equations:

$$m\ddot{h} + S_\alpha\ddot{\alpha} + m\omega_h^2 h = -L, \quad S_\alpha\ddot{h} + I_\alpha\ddot{\alpha} + I_\alpha\omega_\alpha^2\alpha = M_y \tag{14}$$

where m is the mass per unit span, S_α is the static unbalance, I_α is the sectional moment of inertia about the elastic axis, h is the plunge, α is the angle of attack or pitch in radians, L is the sectional lift, and M_y is the sectional moment about the elastic axis. In this work, the structural governing equations are integrated forward in time using a Newmark-beta method ([35] chap. 9, p. 780).

The values of the structural model parameters shown in Table 3 are taken from the nondimensional parameters and natural frequencies used by Isogai [23]. The nondimensional mass ratio is computed as

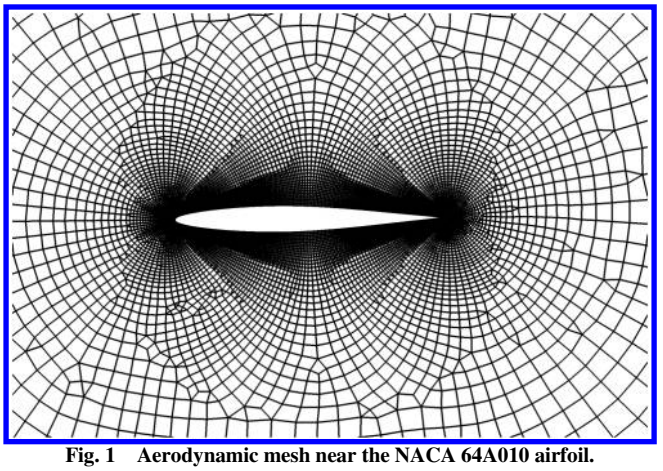


Fig. 1 Aerodynamic mesh near the NACA 64A010 airfoil.

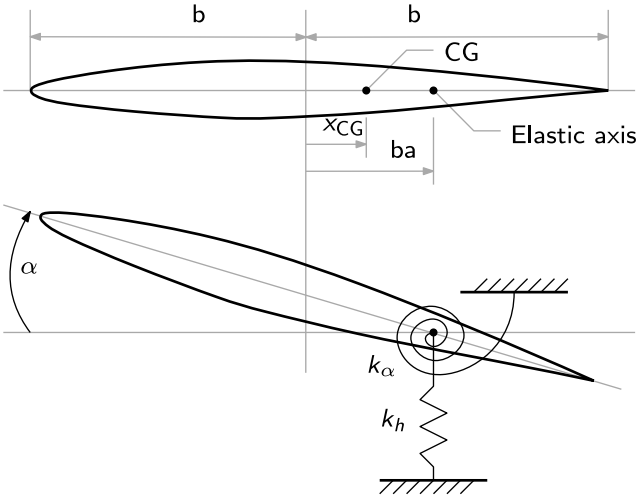


Fig. 2 Illustration of the pitch-plunge airfoil model.

Table 3 Selected NACA 64A010 structural properties

Property	Value
Mass ratio (μ)	60
Elastic axis location (a)	−2.0
Normalized static unbalance (S_α/mb)	1.8
Radius of gyration (r_α^2)	3.48
Plunge natural frequency [rad/s] (ω_h)	100
Pitch natural frequency [rad/s] (ω_α)	100

$$\mu = \frac{m}{\rho_\infty \pi b^2}$$

and the radius of gyration, r_α, is given as

$$r_\alpha^2 = \frac{I_\alpha}{mb^2}$$

which take values of $\mu = 60$, and $r_\alpha^2 = 3.48$, respectively. Additionally, the nondimensional flutter speed index is defined as

$$V_f = \frac{U_\infty}{b\omega_\alpha\sqrt{\mu}} \tag{15}$$

where U_∞ is the onset velocity.

Within the flutter identification optimization problem, the speed index (15) is treated as the design variable. All other properties of the structural model and the nondimensionalized properties of the aerodynamic flow are held fixed. Therefore, as the speed index is varied by the optimizer, the dynamic pressure and the physical time step must be adjusted to keep the nondimensionalized time step (13) fixed. Because the structural governing equations are integrated in physical units, the structural time step is treated as a design-dependent parameter within the simulation and differentiated within the adjoint method.

In each simulation, the aerodynamic initial conditions are set to uniform free stream. For the structural model, the plunge is initially held fixed and the pitch motion is forced at 100 rad/s with an amplitude of 1° for one period of forcing. After this initial forced motion, the structural degrees of freedom are released and driven by the governing Eqs. (14). To estimate the aeroelastic damping, the matrix pencil method is applied to the final 1000 steps of the time history of the pitch angle. The pitch time history is down sampled to $N = 250$ time samples using Eq. (4) within the matrix pencil method.

Each optimization problem converged in between 4 and 6 iterations, where each iteration required a forward time-dependent analysis and an adjoint-based gradient evaluation of the matrix pencil derivative. Each simulation ran on 72 processor cores and required approximately 25 min of wall time for a single forward analysis and 29 min of wall time for the adjoint evaluation. The total time taken within the matrix pencil method itself is less than 5 s of wall time per optimization iteration. Figure 3 shows the predicted flutter speed index (15) as a function of Mach number with comparisons against the work of Alonso and Jameson [26], Liu et al. [25], Biao et al. [28], and the recent results of Thomas et al. [27] using the CUPyDO framework. The results from this work using the matrix pencil method, labeled "FUNtoFEM" in Fig. 3, demonstrate good agreement with the results from other authors, capturing both the transonic flutter dip and a portion of the upper branch of the flutter boundary.

Figure 4 illustrates the convergence behavior of the optimization problems at $M = 0.7$ and $M = 0.875$, respectively. In both cases, the optimizer requires six iterations to converge. In the $M = 0.7$ case, the method converges from a stable solution with positive damping, whereas in the $M = 0.875$ case, the method converges from an unstable solution with negative damping. These results also demonstrate that relatively short time intervals can be used to obtain a good damping estimate.

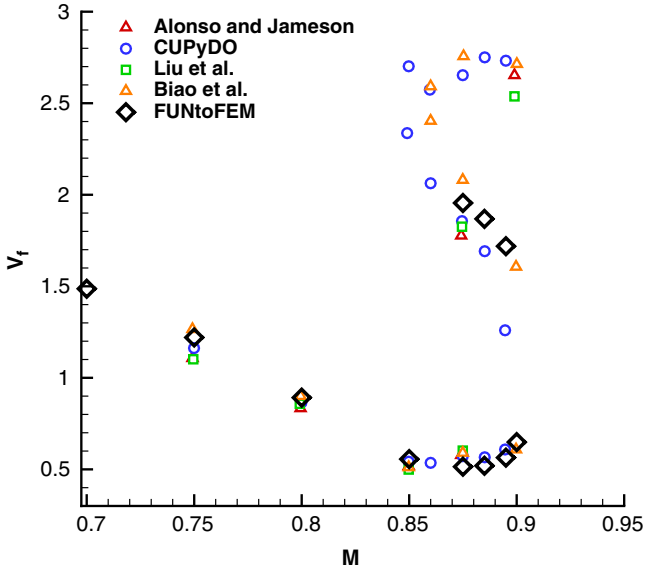
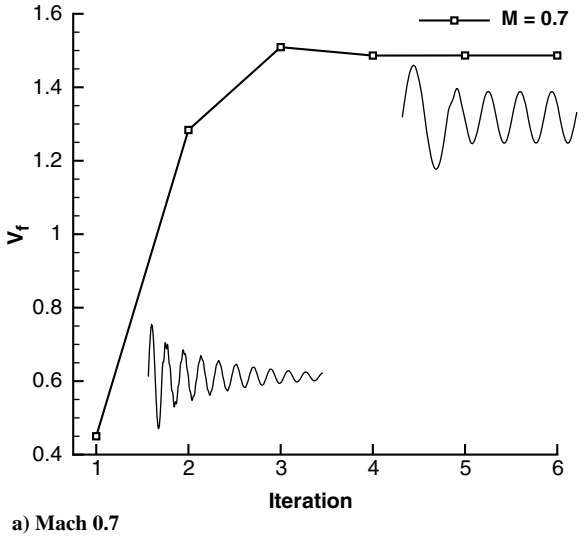
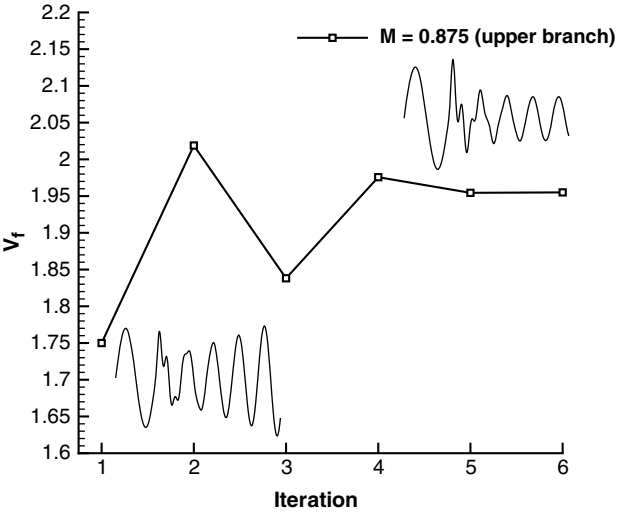


Fig. 3 NACA 64A010 flutter boundary.



a) Mach 0.7



b) Mach 0.875 (upper branch)

Fig. 4 Flutter point identification optimization convergence histories. The pitch response is illustrated at the initial and final iterations of the optimization.

VI. Conclusions

This study presented a matrix pencil method for evaluating the damping in unsteady aeroelastic simulations. In the proposed method, the matrix pencil method is used to extract the damping from a Prony series, and a critical damping estimate is obtained via Kreisselmeier–Steinhauser aggregation, providing a conservative bound on the minimum damping. Finite-difference comparisons were used to verify the derivative implementation, because the singular value decomposition and eigenproblem solution techniques within the method are incompatible with regular complex-step verification procedures. As a final demonstration, the matrix pencil method was used to identify the flutter boundary of the NACA 64A010 benchmark aeroelastic test case, giving good agreement with established results.

Acknowledgments

The authors gratefully acknowledge the funding provided by NASA through the Transformative Tools and Technologies program with grant number NNX15AU22A with Technical Monitor Steve Massey. The authors specifically thank Jordan Trout for his assistance in running the final aeroelastic benchmark cases. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

References

[1] Clark, W. S., and Hall, K. C., "A Time-Linearized Navier–Stokes Analysis of Stall Flutter," *Journal of Turbomachinery*, Vol. 122, No. 3, Feb. 1999, pp. 467–476. doi:10.1115/1.1303073

[2] Jonsson, E., Riso, C., Lupp, C. A., Cesnik, C. E. S., Martins, J. R. R. A., and Epureanu, B. I., "Flutter and Post-Flutter Constraints in Aircraft Design Optimization," *Progress in Aerospace Sciences* (in press). doi:10.1016/j.paerosci.2019.04.001

[3] Jacobson, K. E., Kiviaho, J. F., Kennedy, G. J., and Smith, M. J., "Evaluation of Time-Domain Damping Identification Methods for Flutter-Constrained Optimization," *Journal of Fluids and Structures*, Vol. 87, May 2019, pp. 174–188. doi:10.1016/j.jfluidstructs.2019.03.011

[4] Hua, Y., and Sarkar, T. K., "Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 5, 1990, pp. 814–824. doi:10.1109/29.56027

[5] Razavilar, J., Li, Y., and Liu, K. J. R., "Spectral Estimation Based on Structured Low Rank Matrix Pencil," *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 5, IEEE Publ., Piscataway, NJ, 1996, pp. 2503–2506. doi:10.1109/ICASSP.1996.547972

[6] Maute, K., Nikbay, M., and Farhat, C., "Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems," *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2051–2061. doi:10.2514/2.1227

[7] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62

[8] Brezillon, J., Ronzheimer, A., Haar, D., Abu-Zurayk, M., Lummer, M., Kruger, W., and Natterer, F. J., "Development and Application of Multi-Disciplinary Optimization Capabilities Based on High-Fidelity Methods," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA Paper 2012-1757, 2012. doi:10.2514/6.2012-1757

[9] Ghazlane, I., Carrier, G., Dumont, A., and Desideri, J. A., "Aerostructural Adjoint Method for Flexible Wing Optimization," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA Paper 2012-1924, 2012. doi:10.2514/6.2012-1924

[10] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014 pp. 935–951. doi:10.2514/1.J052255

[11] Kiviaho, J. F., Jacobson, K., Smith, M. J., and Kennedy, G., "A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2017-4144, 2017. doi:10.2514/6.2017-4144

[12] Mishra, A., Mavriplis, D., and Sitaraman, J., "Time-Dependent Aeroelastic Adjoint-Based Aerodynamic Shape Optimization of Helicopter Rotors in Forward Flight," *AIAA Journal*, Vol. 54, No. 12, 2016, pp. 3813–3827. doi:10.2514/1.J054962

[13] Mavriplis, D. J., Fabiano, E., and Anderson, E., "Recent Advances in High-Fidelity Multidisciplinary Adjoint-Based Optimization with the NSU3D Flow Solver Framework," *55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, AIAA Paper 2017-1669, 2017. doi:10.2514/6.2017-1669

[14] Fabiano, E., and Mavriplis, D., "Adjoint-Based Aeroacoustic Design-Optimization of Flexible Rotors in Forward Flight," *Journal of the American Helicopter Society*, Vol. 62, No. 4, 2017, pp. 1–17. doi:10.4050/JAHS.62.042005

[15] Wang, L., Diskin, B., Biedron, R., Nielsen, E. J., and Bauchau, O., "Sensitivity Analysis of Multidisciplinary Rotorcraft Simulations," *55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, AIAA Paper 2017-1670, 2017. doi:10.2514/6.2017-1670

[16] Jacobson, K., Kiviaho, J. F., Smith, M. J., and Kennedy, G., "An Aeroelastic Coupling Framework for Time-Accurate Analysis and Optimization," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, AIAA Paper 2018-0100, 2018. doi:10.2514/6.2018-0100

[17] Kiviaho, J. F., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least-Squares," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2237–2243. doi:10.2514/1.J057318

[18] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *IFAC Proceedings Volumes*, Vol. 12, No. 7, Sept. 1979, pp. 113–117. doi:10.1016/S1474-6670(17)65584-8

[19] Kennedy, G. J., and Hicken, J. E., "Improved Constraint-Aggregation Methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, June 2015, pp. 332–354. doi:10.1016/j.cma.2015.02.017

[20] Golub, G. H., and Pereyra, V., "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate," *SIAM Journal on Numerical Analysis*, Vol. 10, No. 2, 1973, pp. 413–432. doi:10.1137/0710036

[21] Giles, M., "An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Algorithmic Differentiation," TR 08/01, Oxford Univ. Computing Lab., Oxford, Jan. 2008.

[22] Papadopoulo, T., and Lourakis, M. I. A., "Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications," *Computer Vision—ECCV 2000*, Springer, Berlin, 2000, pp. 554–570, https://hal.inria.fr/inria-00072686.

[23] Isogai, K., "On the Transonic-Dip Mechanism of Flutter of a Sweptback Wing," *AIAA Journal*, Vol. 17, No. 7, July 1979, pp. 793–795. doi:10.2514/3.61226

[24] Davis, S. S., "NACA 64A010 (NASA Ames Model) Oscillatory Pitching," AGARD Rept. 702, 1982.

[25] Liu, F., Cai, J., Zhu, Y., Tsai, H. M., and Wong, A. S. F., "Calculation of Wing Flutter by a Coupled Fluid-Structure Method," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 334–342. doi:10.2514/2.2766

[26] Alonso, J., and Jameson, A., "Fully-Implicit Time-Marching Aeroelastic Solutions," *32nd Aerospace Sciences Meeting and Exhibit*, AIAA Paper 1994-0056, 1994. doi:10.2514/6.1994-56

[27] Thomas, D., Cerquaglia, M. L., Boman, R., Economon, T. D., Alonso, J. J., Dimitriadis, G., and Terrapon, V. E., "CUPyDO—An Integrated Python Environment for Coupled Fluid-Structure Simulations," *Advances in Engineering Software*, Vol. 128, Feb. 2019, pp. 69–85. doi:10.1016/j.advengsoft.2018.05.007

[28] Biao, Z., Zhide, Q., and Chao, G., "Transonic Flutter Analysis of an Airfoil with Approximate Boundary Method," *26th International Congress of the Aeronautical Sciences*, Vol. 7, No. 5, ICAS, Sept. 2008.

[29] Kraft, D., "A Software Package for Sequential Quadratic Programming," TR DFVLR-FB 88-28, DLR German Aerospace Center, Köln, 1988.

[30] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3

[31] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163. doi:10.2514/2.1765

[32] Nielsen, E. J., and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1355–1373. doi:10.2514/1.J051859

[33] Vatsa, V. N., Carpenter, M. H., and Lockard, D. P., "Re-Evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications," *Proceedings of the 48th AIAA Aerospace Sciences Meeting*, AIAA Paper 2010-0122, 2010. doi:10.2514/6.2010-122

[34] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity*, Dover Books on Aeronautical Engineering Series, Dover Publ., Mineola, NY, 1996.

[35] Bathe, K.-J., *Finite Element Procedures*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1996.

D. E. Raveh
Associate Editor