AIAA Journal
Vol. 40, No. 11, November 2002

QR-Based Algorithm for Eigenvalue Derivatives

Bradley T. Burchett*
Rose–Hulman Institute of Technology, Terre Haute, Indiana 47803
and
Mark Costello†
Oregon State University, Corvallis, Oregon 97331

Many engineering optimization problems require the calculation of eigenvalue sensitivities. Several straightforward methods have been developed for calculating partial derivatives of distinct complex eigenvalues. Recent work has expanded these methods to include systems with repeated real eigenvalues. A new method is presented that embeds eigenvalue derivative computation into an established numerical eigenvalue algorithm, namely, QR decomposition. The technique is shown to have computational advantages for high-order systems.

Nomenclature

$\boldsymbol{A}$ = square real matrix
$\boldsymbol{e}_1$ = unit vector parallel with Cartesian axis
m = dimension of $\boldsymbol{A}$
$\boldsymbol{p}$ = vector of parameters on which $\boldsymbol{A}$ depends
s = number of first varying parameter
t = number of second varying parameter
$\boldsymbol{v}$ = householder reflector vector
$\boldsymbol{x}$ = subcolumn of the $\boldsymbol{A}$ matrix

Introduction

THE algebraic eigenvalue problem has been studied extensively for many years because of its wide array of applications in engineering. Some examples are finding the natural frequencies of multi-degree-of-freedom vibrational systems, finding the principle stresses in three-dimensional deformable bodies, finding the principle moments of inertia for three-dimensional rigid bodies, and determining stability and performance of feedback control systems. The sensitivities of eigenvalues are, thus, important in designing systems from the preceding examples. First and second derivatives provide a means of determining system robustness and directly optimizing a design. In many cases, the system model has a large order, resulting in large dimension matrices. This motivates the need for algorithms that are both numerically efficient and stable.

Considerable attention has been given to the problem of computation of the derivatives of the eigenvalues and eigenvectors of a linear system with respect to system parameters. Murthy and Haftka[1] surveyed the different numerical methods for performing these operations and provide an extensive reference list of work performed in this area. As discussed by Murthy and Haftka,[1] three classes of methods have emerged: adjoint, direct, and iterative. Adjoint methods generally require both left and right eigenvectors for derivative calculations. The derivative of a particular eigenvector is expressed as a weighted sum of the system eigenvectors. One example of an adjoint method is that by Rogers.[2] Direct methods typically involve the solution to a system of linear equations. Unlike adjoint methods, eigenvector derivatives only involve a single eigenvector and not the entire eigensystem. Garg,[3] as well as Rudisill and Chu,[4] provides examples of direct methods. More recent work by Jankovic[5] presents a method for computing eigensystem derivatives applicable to linear and nonlinear problems. Nelson[6] derived a computationally efficient direct method. Lim et al.[7] developed a method to calculate derivatives of repeated eigenvalues and their eigenvectors using singular value decomposition. Zhang and Wei[8] presented a method using a complete modal space. Song et al.[9] invented a simplified method based on Nelson's efficient approach.[6] Chen[10] demonstrated a method for doubly repeated eigenvalues. Wei and Zhang[11] also presented a method using the generalized inverse technique. More recently, Friswell[12] and Prells and Friswell[13] developed a method based on Nelson's method.[6] The bulk of the methods mentioned are tailored to structural dynamics and, thus, take advantage of the distinct properties of the generalized eigenvalue problem for real symmetric matrices. Also, in structural dynamics applications, usually only a few eigenvalue derivatives are required. The method presented in this paper is developed specifically for use in designing feedback control systems where the matrix of interest is not symmetric and, thus, has both real and complex conjugate eigenvalues. Moreover, in most feedback control applications, eigenvalue derivatives for all eigenvalues are required. Efficient and reliable numerical tools exist for computing the eigensystem, such as the QR algorithm. In this paper, computation of eigenvalue derivatives is imbedded inside the basic QR algorithm so that eigenvalue derivatives are computed while eigenvalues are generated.

Received 11 June 2001; revision received 5 March 2002; accepted for publication 29 June 2002.

*Assistant Professor, Department of Mechanical Engineering.
†Assistant Professor, Department of Mechanical Engineering. Member AIAA.

Numerical Calculation of Eigenvalues

Any general real-valued matrix $\boldsymbol{A}$ can be factored into a unitary matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$ using the following algorithm[14–18] for $k = 1$–m:

$$\boldsymbol{x} = \boldsymbol{A}_{k:m,k} \tag{1}$$

$$v_k = \mathrm{sign}(\boldsymbol{x}_1)\|\boldsymbol{x}\|_2 \boldsymbol{e}_1 + \boldsymbol{x} \tag{2}$$

$$v_k = v_k / \|v_k\|_2 \tag{3}$$

$$\boldsymbol{A}_{k:m,k:m} = \boldsymbol{A}_{k:m,k:m} - 2v_k\left(v_k^H \boldsymbol{A}_{k:m,k:m}\right) \tag{4}$$

End

The QR algorithm is well known and documented in many sources.[14–18] Note that as the factorization proceeds, the length of the vectors decreases, thus decreasing the computational effort. The QR factorization can be used to find the matrix's real or complex Schur form by iterating. The QR algorithm is typically sped up by two important steps, first, by finding the upper Hessenberg form and second, by applying shifts. The upper Hessenberg form is found by executing the QR factorization with k taking the values from 1 to $m-2$. The subscripts in Eq. (1) are amended to $k+1{:}m$ and k. Equation (4) is evaluated twice for each k with subscript $k+1{:}m$ and $k{:}m$ for the first evaluation and $1{:}m$ and $k+1{:}m$ for the second. After the upper Hessenberg form is obtained, the QR factorization is iterated using a shift. The algorithm is, for $k = 1, 2, \ldots, m$, as follows:

1) Pick a shift $\boldsymbol{\sigma}_k$.
2) Determine the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices from the shifted $\boldsymbol{A} = \boldsymbol{A} - \boldsymbol{\sigma}_k \boldsymbol{I}$ matrix.
3) Solve $\boldsymbol{A} = \boldsymbol{R}\boldsymbol{Q} + \boldsymbol{\sigma}_k \boldsymbol{I}$.
4) Repeat these three steps until convergence.

End

The single-shift algorithm is preferred in this work because it results in complex Schur form, where all eigenvalues, both real and complex, are exposed along the matrix diagonal. The following sections describe how eigenvalue derivatives can be embedded in the basic QR algorithm.

Derivatives of a Single Householder QR Factorization

The components of this factorization can be differentiated as follows. First, assign the $\boldsymbol{x}$ vector derivatives equal to elements k–m of the kth column of the derivative matrices:

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_s} = \left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_s}\right]_{k:m,k} \tag{5}$$

$$\frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} = \left[\frac{\partial^2 \boldsymbol{A}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\right]_{k:m,k} \tag{6}$$

Next, define the derivative of the two-norm of a vector as

$$\frac{\partial \|\boldsymbol{x}\|_2}{\partial \boldsymbol{p}_s} = \frac{1}{2}(\boldsymbol{x}^H \boldsymbol{x})^{-\frac{1}{2}} \left(\frac{\partial \boldsymbol{x}^H}{\partial \boldsymbol{p}_s}\boldsymbol{x} + \boldsymbol{x}^H \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_s}\right) \tag{7}$$

Then, the first derivatives of the $\boldsymbol{v}_k$ vector are

$$\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s} = \operatorname{sign}(\boldsymbol{x}_1)\frac{\partial \|\boldsymbol{x}\|_2}{\partial \boldsymbol{p}_s}\boldsymbol{e}_1 + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_s} \tag{8}$$

Define the second derivative of the two-norm of a vector as

$$\begin{aligned}\frac{\partial \|\boldsymbol{x}\|_2}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} = \Bigg[&-\frac{1}{4}(\boldsymbol{x}^H \boldsymbol{x})^{-\frac{3}{2}}\left(\frac{\partial \boldsymbol{x}^H}{\partial \boldsymbol{p}_t}\boldsymbol{x} + \boldsymbol{x}^H \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_t}\right)\left(\frac{\partial \boldsymbol{x}^H}{\partial \boldsymbol{p}_s}\boldsymbol{x} + \boldsymbol{x}^H \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_s}\right) \\ &+ \frac{1}{2}(\boldsymbol{x}^H \boldsymbol{x})^{-\frac{1}{2}}\left(\frac{\partial^2 \boldsymbol{x}^H}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\boldsymbol{x} + \frac{\partial \boldsymbol{x}^H}{\partial \boldsymbol{p}_s}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_t} + \frac{\partial \boldsymbol{x}^H}{\partial \boldsymbol{p}_t}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}_s} + \boldsymbol{x}^H \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\right)\Bigg]\end{aligned} \tag{9}$$

The second derivative is

$$\frac{\partial^2 \boldsymbol{v}_k}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} = \operatorname{sign}(\boldsymbol{x}_1)\frac{\partial \|\boldsymbol{x}\|_2}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\boldsymbol{e}_1 + \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} \tag{10}$$

The first derivative scaling equations are then

$$\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s} = \frac{1}{\|\boldsymbol{v}_k\|_2^2}\left(\|\boldsymbol{v}_k\|_2 \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s} - \boldsymbol{v}_k \frac{\partial \|\boldsymbol{v}_k\|_2}{\partial \boldsymbol{p}_s}\right) \tag{11}$$

The second derivative scaling equation is

$$\begin{aligned}\frac{\partial^2 \boldsymbol{v}_k}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} = \frac{1}{\|\boldsymbol{v}_k\|_2^4}\Bigg[&\|\boldsymbol{v}_k\|_2^2\Bigg(\frac{\partial \|\boldsymbol{v}_k\|_2}{\partial \boldsymbol{p}_t}\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s} + \|\boldsymbol{v}_k\|_2 \frac{\partial^2 \boldsymbol{v}_k}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} \\ &- \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\frac{\partial \|\boldsymbol{v}_k\|_2}{\partial \boldsymbol{p}_s} - \boldsymbol{v}_k \frac{\partial \|\boldsymbol{v}_k\|_2}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\Bigg) - \left(\|\boldsymbol{v}_k\|_2 \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s} - \boldsymbol{v}_k \frac{\partial \|\boldsymbol{v}_k\|_2}{\partial \boldsymbol{p}_s}\right) \\ &\times \left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_t}\boldsymbol{v}_k + \boldsymbol{v}_k^H \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\right)\Bigg]\end{aligned} \tag{12}$$

The system derivative matrices are then updated as follows:

$$\begin{aligned}\left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_t}\right]_{k:m,k:m} = \left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_t}\right]_{k:m,k:m} &- 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\left(\boldsymbol{v}_k^H \boldsymbol{A}_{k:m,k:n}\right) \\ &- 2\boldsymbol{v}_k\left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_t}\boldsymbol{A}_{k:m,k:n}\right) - 2\boldsymbol{v}_k\left(\boldsymbol{v}_k^H \left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_t}\right]_{k:m,k:n}\right)\end{aligned} \tag{13}$$

$$\begin{aligned}\left[\frac{\partial^2 \boldsymbol{A}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\right]_{k:m,k:m} = \left[\frac{\partial^2 \boldsymbol{A}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\right]_{k:m,k:m} &- 2\frac{\partial^2 \boldsymbol{v}_k}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\left(\boldsymbol{v}_k^H \boldsymbol{A}_{k:m,k:n}\right) \\ &- 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s}\left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_t}\boldsymbol{A}_{k:m,k:n}\right) - 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s}\left(\boldsymbol{v}_k^H \left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_t}\right]_{k:m,k:m}\right) \\ &- 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_s}\boldsymbol{A}_{k:m,k:n}\right) - 2\boldsymbol{v}_k\left(\frac{\partial^2 \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\boldsymbol{A}_{k:m,k:n}\right) \\ &- 2\boldsymbol{v}_k\left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_s}\left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_t}\right]_{k:m,k:m}\right) - 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\left(\boldsymbol{v}_k^H \left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_s}\right]_{k:m,k:m}\right) \\ &- 2\boldsymbol{v}_k\left(\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_t}\left[\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{p}_s}\right]_{k:m,k:m}\right) - 2\boldsymbol{v}_k\left(\boldsymbol{v}_k^H \left[\frac{\partial^2 \boldsymbol{A}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\right]_{k:m,k:m}\right)\end{aligned} \tag{14}$$

When these derivatives are combined with the preceding algorithm, the following algorithm for computing first and second derivatives of a single Householder QR factorization emerges for $k = 1{:}\ m - 1$:

1) Solve $\boldsymbol{x} = \boldsymbol{A}_{k:m,k}$.
2) Compute $\partial \boldsymbol{x}/\partial \boldsymbol{p}_s$ using Eq. (5).
3) Compute $\partial^2 \boldsymbol{x}/\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t$ using Eq. (6).
4) Solve $\boldsymbol{v}_k = \operatorname{sign}(\boldsymbol{x}_1)\|\boldsymbol{x}\|_2 \boldsymbol{e}_1 + \boldsymbol{x}$.
5) Compute $\partial \|\boldsymbol{x}\|_2/\partial \boldsymbol{p}_s$ using Eq. (7).
6) Compute $\partial \boldsymbol{v}_k/\partial \boldsymbol{p}_s$ using Eq. (8).
7) Compute $\partial^2 \|\boldsymbol{x}\|_2/\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t$ using Eq. (9).
8) Compute $\partial^2 \boldsymbol{v}_k/\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t$ using Eq. (10).
9) Scale $\partial^2 \boldsymbol{v}_k/\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t$ using Eq. (12).
10) Scale $\partial \boldsymbol{v}_k/\partial \boldsymbol{p}_s$ using Eq. (11).
11) Solve $\boldsymbol{v}_k = \boldsymbol{v}_k/\|\boldsymbol{v}_k\|_2$.
12) Update $\partial^2 \boldsymbol{A}/\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t$ using Eq. (14).
13) Update $\boldsymbol{v}_k = \partial \boldsymbol{A}/\partial \boldsymbol{p}_s$ using Eq. (13).

End

The preceding steps generate the upper triangular matrix $\boldsymbol{R}$ and its first and second derivatives.

Derivatives of the Unitary Transformation Matrix

The following steps are required to calculate the unitary matrix $\boldsymbol{Q}$ and its first and second derivatives. Given the formula for a single Householder unitary reflector matrix $\boldsymbol{F}$,

$$\boldsymbol{F} = \boldsymbol{I} - 2(\boldsymbol{v}\boldsymbol{v}^H/\boldsymbol{v}^H\boldsymbol{v})$$

The first and second derivatives of this matrix are given as

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}_s} = -2\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{p}_s}\boldsymbol{v}^H - 2\boldsymbol{v}\frac{\partial \boldsymbol{v}^H}{\partial \boldsymbol{p}_s}$$

$$\frac{\partial^2 \boldsymbol{F}}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t} = -2\frac{\partial^2 \boldsymbol{v}_k}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}\boldsymbol{v}_k^H - 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_s}\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_t} - 2\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{p}_t}\frac{\partial \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_s} - 2\boldsymbol{v}_k\frac{\partial^2 \boldsymbol{v}_k^H}{\partial \boldsymbol{p}_s \partial \boldsymbol{p}_t}$$

The unitary matrix $\boldsymbol{Q}$ is given as the product of m reflector matrices:

$$\boldsymbol{Q} = \begin{bmatrix}\boldsymbol{I} & 0 \\ 0 & \boldsymbol{F}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & 0 \\ 0 & \boldsymbol{F}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_3\end{bmatrix}\cdots\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_m\end{bmatrix}$$

$$\boldsymbol{Q} = \boldsymbol{Q}_1 \boldsymbol{Q}_2 \cdots \boldsymbol{Q}_m$$

The first derivative of $\boldsymbol{Q}$ is, thus, given as a sum of products of $\boldsymbol{Q}_i$ and $\partial \boldsymbol{Q}_i/\partial \boldsymbol{p}_s$:

$$\begin{aligned}\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{p}_s} = \frac{\partial \boldsymbol{Q}_1}{\partial \boldsymbol{p}_s}\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_m + \boldsymbol{Q}_1 \frac{\partial \boldsymbol{Q}_2}{\partial \boldsymbol{p}_s}\boldsymbol{Q}_3 \cdots \boldsymbol{Q}_m \\ + \boldsymbol{Q}_1 \boldsymbol{Q}_2 \cdots \frac{\partial \boldsymbol{Q}_{m-1}}{\partial \boldsymbol{p}_s}\boldsymbol{Q}_m + \boldsymbol{Q}_1 \boldsymbol{Q}_2 \cdots \boldsymbol{Q}_{m-1}\frac{\partial \boldsymbol{Q}_m}{\partial \boldsymbol{p}_s}\end{aligned}$$

The second derivative of $\boldsymbol{Q}$ then follows as the sum of products of $\boldsymbol{Q}_i, \partial\boldsymbol{Q}_i/\partial\boldsymbol{p}_s$, $\partial\boldsymbol{Q}_i/\partial\boldsymbol{p}_t$, and $\partial^2\boldsymbol{Q}_i/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$:

$$\begin{aligned}\frac{\partial^2\boldsymbol{Q}}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t} &= \frac{\partial^2\boldsymbol{Q}_1}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}\boldsymbol{Q}_1\cdots\boldsymbol{Q}_m + \frac{\partial\boldsymbol{Q}_1}{\partial\boldsymbol{p}_s}\frac{\partial\boldsymbol{Q}_2}{\partial\boldsymbol{p}_t}\boldsymbol{Q}_3\cdots\boldsymbol{Q}_m \\ &+ \frac{\partial\boldsymbol{Q}_1}{\partial\boldsymbol{p}_s}\boldsymbol{Q}_2\cdots\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_t} + \frac{\partial\boldsymbol{Q}_1}{\partial\boldsymbol{p}_t}\frac{\partial\boldsymbol{Q}_2}{\partial\boldsymbol{p}_s}\boldsymbol{Q}_3\cdots\boldsymbol{Q}_m \\ &+ \boldsymbol{Q}_1\frac{\partial^2\boldsymbol{Q}_2}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}\boldsymbol{Q}_3\cdots\boldsymbol{Q}_m + \boldsymbol{Q}_1\frac{\partial\boldsymbol{Q}_2}{\partial\boldsymbol{p}_s}\boldsymbol{Q}_3\cdots\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_t} + \cdots \\ &+ \frac{\partial\boldsymbol{Q}_1}{\partial\boldsymbol{p}_t}\boldsymbol{Q}_2\cdots\frac{\partial\boldsymbol{Q}_{m-1}}{\partial\boldsymbol{p}_s}\boldsymbol{Q}_m + \boldsymbol{Q}_1\frac{\partial\boldsymbol{Q}_2}{\partial\boldsymbol{p}_t}\cdots\frac{\partial\boldsymbol{Q}_{m-1}}{\partial\boldsymbol{p}_s}\boldsymbol{Q}_m \\ &+ \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\frac{\partial^2\boldsymbol{Q}_{m-1}}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}\boldsymbol{Q}_m + \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\frac{\partial\boldsymbol{Q}_{m-1}}{\partial\boldsymbol{p}_s}\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_t} \\ &+ \frac{\partial\boldsymbol{Q}_1}{\partial\boldsymbol{p}_t}\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-1}\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_s} + \boldsymbol{Q}_1\frac{\partial\boldsymbol{Q}_2}{\partial\boldsymbol{p}_t}\cdots\boldsymbol{Q}_{m-1}\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_s} \\ &+ \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\frac{\partial\boldsymbol{Q}_{m-1}}{\partial\boldsymbol{p}_t}\frac{\partial\boldsymbol{Q}_m}{\partial\boldsymbol{p}_s} + \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-1}\frac{\partial^2\boldsymbol{Q}_m}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}\end{aligned}$$

QR Algorithm for Eigenvalue Derivatives

By the use of the preceding definitions, the following algorithm yeilds eigenvalue derivatives while eigenvalues derivatives are computed. First, find the upper Hessenberg form of the system and system derivative matrices for $k = 1{:}m - 2$; then, perform the following steps:

1) Assign $\boldsymbol{x}$, $\partial\boldsymbol{x}/\partial\boldsymbol{p}_s$, and $\partial^2\boldsymbol{x}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ equal to the $k + 1{:}m$ rows and kth column of the system dynamics matrix and appropriate derivatives, respectively.

2) Compute $\boldsymbol{v}_k$ using Eq. (2).

3) Compute $\partial\|\boldsymbol{x}\|_2/\partial\boldsymbol{p}_s$ using Eq. (7).

4) Compute $\partial\boldsymbol{v}_k/\partial\boldsymbol{p}_s$ using Eq. (8).

5) Compute $\partial^2\|\boldsymbol{x}\|_2/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using Eq. (9).

6) Compute $\partial^2\boldsymbol{v}_k/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using Eq. (10).

7) Compute $\partial\|\boldsymbol{v}_k\|_2/\partial\boldsymbol{p}_s$ using Eq. (7).

8) Compute $\partial^2\|\boldsymbol{v}_k\|_2/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using Eq. (9).

9) Scale $\partial^2\boldsymbol{v}_k/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using Eq. (12).

10) Scale $\partial\boldsymbol{v}_k/\partial\boldsymbol{p}_s$ using Eq. (11).

11) Scale $\boldsymbol{v}_k$ using Eq. (3).

12) Update $\partial^2\boldsymbol{A}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using Eq. (14) with subscripts $k + 1{:}m$ and $k{:}m$.

13) Update $\partial^2\boldsymbol{A}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ using transposed version of Eq. (14) with subscripts $1{:}m$ and $k + 1{:}m$.

14) Update $\partial\boldsymbol{A}/\partial\boldsymbol{p}_s$ using Eq. (13) with subscripts $k + 1{:}m$ and $k{:}m$.

15) Update $\partial\boldsymbol{A}/\partial\boldsymbol{p}_s$ using transposed version of Eq. (13) with subscripts $1{:}m$ and $k + 1{:}m$.

16) Update $\boldsymbol{A}$ using Eq. (4) with subscripts $k + 1{:}m$ and $k{:}m$.

17) Update $\boldsymbol{A}$ using transposed version of Eq. (4) with subscripts $1{:}m$ and $k + 1{:}m$.

End

The Hessenberg permutation matrix $\boldsymbol{P}$ and its first and second derivatives are found just as in the single QR step, except that the dimensions of the reflector matrices $\boldsymbol{F}$ are amended.

After the Hessenberg form is obtained, a single-shift QR algorithm with complex shifts admissible is employed. At each step, the lower right-hand 2×2 portion of the matrix is tested for complex eigenvalues. If the eigenvalues are complex, a complex shift equal to an eigenvalue is selected until convergence of the last row. For consistency, a shift equal to the complex conjugate of the resulting eigenvalue for the next to last row is chosen. If the eigenvalues are real, we choose a real shift equal to the bottom right matrix element. The single-shift approach is preferred instead of the double-shift approach because in the single-shift approach, the complex eigenvalues and their derivatives are exposed along the $\boldsymbol{R}$ matrix diagonal. In the double-shift case, there is no intuitively obvious relationship between the resulting real $\partial\boldsymbol{R}/\partial\boldsymbol{p}_s$, $\partial\boldsymbol{R}/\partial\boldsymbol{p}_t$, and $\partial^2\boldsymbol{R}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ matrices and the eigenvalue derivatives that occur in complex-conjugate pairs. The resulting algorithm is as follows for $k = 1, 2, \ldots, m$:

1) Pick a shift $\boldsymbol{\sigma}_k$

2) Determine the $\boldsymbol{Q}$, $\boldsymbol{R}$, $\partial\boldsymbol{Q}/\partial\boldsymbol{p}_s$, $\partial\boldsymbol{Q}/\partial\boldsymbol{p}_t$, $\partial^2\boldsymbol{Q}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$, $\partial\boldsymbol{R}/\partial\boldsymbol{p}_s$, $\partial\boldsymbol{R}/\partial\boldsymbol{p}_t$, and $\partial^2\boldsymbol{R}/\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t$ matrices from the shifted $\boldsymbol{A} = \boldsymbol{A} - \boldsymbol{\sigma}_k\boldsymbol{I}$ matrix.

3) Solve the following:

$$\boldsymbol{A} = \boldsymbol{RQ} + \sigma_k\boldsymbol{I}$$

$$\frac{\partial\boldsymbol{A}}{\partial\boldsymbol{p}_s} = \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{p}_s}\boldsymbol{Q} + \boldsymbol{R}\frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{p}_s}, \qquad \frac{\partial\boldsymbol{A}}{\partial\boldsymbol{p}_t} = \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{p}_t}\boldsymbol{Q} + \boldsymbol{R}\frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{p}_t}$$

$$\frac{\partial^2\boldsymbol{A}}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t} = \frac{\partial^2\boldsymbol{R}}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}\boldsymbol{Q} + \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{p}_s}\frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{p}_t} + \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{p}_t}\frac{\partial\boldsymbol{Q}}{\partial\boldsymbol{p}_s} + \boldsymbol{R}\frac{\partial^2\boldsymbol{Q}}{\partial\boldsymbol{p}_s\partial\boldsymbol{p}_t}$$

Repeat steps 1–3 until convergence.

End

Comparison of Computational Efficiency

The efficiency of the proposed method is compared with that of Rudisill and Chu.[4] Rudisill and Chu's method is well known and is a practical, direct way of calculating eigenvalue and eigenvector derivatives. The eigenvalue first derivatives are computed by the inner product of left and right eigenvectors with the $\boldsymbol{A}$ matrix first derivative. Right eigenvector derivatives are found by appending a scaling equation to the singular linear system formed from differentiating the algebraic eigenvalue problem. The left and right eigenvector first derivatives are required to compute the eigenvalue second derivatives. Here, we are only comparing the ability to compute eigenvalue first and second derivatives efficiently. The eigenvalue first derivatives are computed easily as the inner product of left eigenvector, $\boldsymbol{A}$ matrix derivative, and right eigenvector. Computing the second derivative using Rudisill and Chu's method requires the computation of a complete set of right and left eigenvector first derivatives. The right eigenvector derivatives each require the solution of $m + 1$ linear equations after multiplying the respective right eigenvector by an $m + 1$ by m matrix, consisting of the $\boldsymbol{A}$ matrix derivative. Counting m inversions of $m + 1$ by $m + 1$ matrices along with the $3m$ inner products, as well as m products of the augmented matrix and right eigenvector along with determining the complete set of eigenvalues and vectors, the total operations count is approximately

$$(11/3)m^4 + 8m^3 + 15m^2 + 5m \tag{15}$$

In contrast, the proposed method computes the eigenvalues and derivatives simultaneously. There is no need to compute system eigenvectors. Essentially, this method requires the standard QR operations count multiplied by an integer. Essentially, the Hessenberg reduction takes six times the number of operations required for first derivatives as for eigenvalues alone and nine times as many for second derivatives. Because the typical Hessenberg reduction requires $(10/3)m^3$ operations (see Ref. 14), this portion of the algorithm for derivatives takes $(160/3)m^3$ operations. Each QR factorization requires three times the number of operations for first derivatives as for eigenvalues alone and nine times as many for second derivaties. For a Hessenberg matrix, the base number is m^2, and so each QR factoriztion, including derivatives, costs a total of $26m^2$ operations. Typically, the matrix will reduce to Schur form in $2m$ QR iterations or less (see Ref. 17), so that the total computational cost is $(238/3)m^3$ operations. The two methods are compared in Fig. 1, where the proposed method has an obvious advantage for systems of order 20 or greater.

Numerical Example

To demonstrate the relevance of this algorithm to the discipline of control, we exercise it in the following control example. The following matrices are a linear time-invariant description of the longitudinal dynamics of an aircraft:

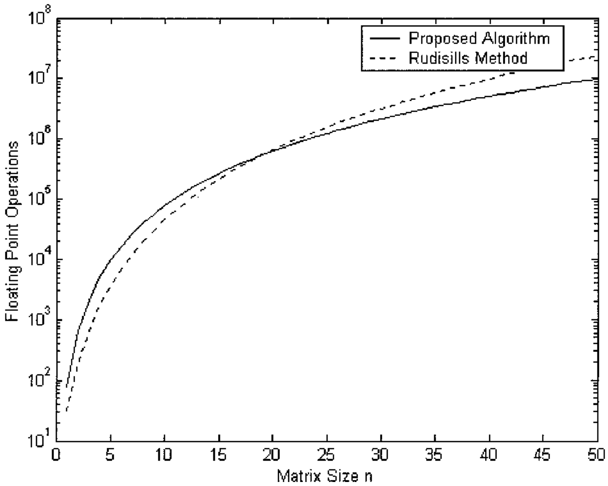

Fig. 1 Operations count comparison.

$$\tilde{A} = \begin{bmatrix} -0.7 & -0.046 & -12.2 & 0 \\ 0 & -0.014 & -0.2904 & -0.582 \\ 1 & -0.0057 & -1.4 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} -19.1 & -3.1 \\ -0.012 & -0.0096 \\ -0.14 & -0.72 \\ 0 & 0 \end{bmatrix}, \qquad \tilde{C} = [0 \quad 0 \quad -1 \quad 1]$$

$$\tilde{D} = [0 \quad 0]$$

Consider the case of static output feedback applied to the preceding system. The closed-loop system dynamics matrix is $A - BKC$. The eigenvalue derivatives are computed with a gain matrix of $K = [0.8; 0.2]$ using the proposed method and compared with finite difference approximations. The results shown hereafter agree favorably:

$$\frac{\partial A}{\partial p_1} = \begin{bmatrix} 0 & 0 & 19.1 & -19.1 \\ 0 & 0 & 0.012 & -0.012 \\ 0 & 0 & 0.14 & -0.14 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial A}{\partial p_2} = \begin{bmatrix} 0 & 0 & 3.1 & -3.1 \\ 0 & 0 & 0.0096 & -0.0096 \\ 0 & 0 & 0.72 & -0.72 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The second derivative of the closed-loop system matrix is zero for $D = 0$. The proposed method computes

$$\frac{\partial \lambda}{\partial p_1} = \begin{bmatrix} 0.93485 + 0.3547i \\ 0.93485 - 0.3547i \\ -1.7299 \\ 0.00016501 \end{bmatrix}$$

$$\frac{\partial \lambda}{\partial p_2} = \begin{bmatrix} 0.17368 + 0.028124i \\ 0.17368 - 0.028124i \\ 0.37256 \\ 0.000089266 \end{bmatrix}$$

$$\frac{\partial^2 \lambda}{\partial p_1 \partial p_2} = \begin{bmatrix} 0.024984 + 0.052727i \\ 0.024984 - 0.052727i \\ -0.050242 \\ -0.000079065 \end{bmatrix}$$

The finite difference approximations are

$$\frac{\partial \lambda}{\partial p_1} = \begin{bmatrix} 0.93485 + 0.3547i \\ 0.93485 - 0.3547i \\ -1.7299 \\ 0.00016501 \end{bmatrix}$$

$$\frac{\partial \lambda}{\partial p_2} = \begin{bmatrix} 0.17368 + 0.028124i \\ 0.17368 - 0.028124i \\ 0.37256 \\ 0.000089266 \end{bmatrix}$$

$$\frac{\partial^2 \lambda}{\partial p_1 \partial p_2} = \begin{bmatrix} 0.025108 + 0.052601i \\ 0.025108 - 0.052601i \\ -0.050138 \\ -0.000079061 \end{bmatrix}$$

Conclusions

By embedding eigenvalue derivative computation inside a well-established numerical eigenvalue routine, a method to compute eigenvalue derivatives reliably and efficiently has been developed. The method uses a single shift QR algorithm, resulting in the matrix complex Schur form and its first and second derivatives. This method is valid for systems with eigenvalues of distinct absolute value. For high-order systems, the QR-based approach is competitive, if not superior, in efficiency to established methods when the first and second eigenvalue derivatives are required.

References

[1]Murthy, D. V., and Haftka, R. T., "Derivatives of Eigenvalues and Eigenvectors of a General Complex Matrix," *International Journal for Numerical Methods in Engineering*, Vol. 26, 1988, pp. 293–311.

[2]Rogers, L. C., "Derivatives of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 8, No. 5, 1970, pp. 943, 944.

[3]Garg, S., "Derivatives of Eigensolutions for a General Matrix," *AIAA Journal*, Vol. 11, No. 8, 1973, pp. 1191–1194.

[4]Rudisill, C. S., and Chu, Y., "Numerical Methods for Evaluating the Derivatives of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 13, No. 6, 1975, pp. 834–837.

[5]Jankovic, M. S., "Exact *n*th Derivatives of Eigenvalues and Eigenvectors," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994, pp. 136–144.

[6]Nelson, R. B., "Simplified Calculation of Eigenvector Derivatives," *AIAA Journal*, Vol. 14, No. 9, 1976, pp. 1201–1205.

[7]Lim, K. B., Juang, J., and Ghaemmaghami, P., "Eigenvector Derivatives of Repeated Eigenvalues Using Singular Value Decomposition," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 1988, pp. 282, 283.

[8]Zhang, D., and Wei, F., "Computation of Eigenvector Derivatives with Repeated Eigenvalues Using a Complete Modal Space," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1749–1753.

[9]Song, D., Han, W., Chen, S., and Qiu, Z., "Simplified Calculation of Eigenvector Derivatives with Repeated Eigenvalues," *AIAA Journal*, Vol. 34, No. 4, 1995, pp. 859–862.

[10]Chen, T., "Eigenvector Derivatives for Doubly Repeated Eigenvalues," *AIAA Journal*, Vol. 34, No. 7, 1996, pp. 1531–1533.

[11]Wei, F., and Zhang, D., "Eigenvector Derivatives with Repeated Eigenvalues Using Generealized Inverse Technique," *AIAA Journal*, Vol. 34, No. 10, 1996, pp. 2206–2209.

[12]Friswell, M. I., "The Derivatives of Repeated Eigenvalues and Their Associated Eigenvectors," *Journal of Vibration and Acoustics*, Vol. 118, 1996, pp. 390–397.

[13]Prells, U., and Friswell, M. I., "Partial Derivatives of Repeated Eigenvalues and Their Eigenvalues," *AIAA Journal*, Vol. 35, No. 8, 1997, pp. 1363–1368.

[14]Trefethen, L. N., and Bau, D., *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[15]Demmel, J. W., *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[16]Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, England, U.K., 1965.

[17]Parlett, B. N., "The QR Algorithm," *Computing in Science and Engineering*, No. 1, Jan./Feb. 2000, pp. 38–42.

[18]Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes*, Cambridge Univ. Press, New York, 1988.

E. R. Johnson
Associate Editor