the stretched variable τ, instead of evaluating it from the explicit outer solution as is done generally.[2]

Example

Consider a simple second-order system so that we can get explicit expressions for the solutions.

$$\frac{dx}{dt} = z, \qquad x(t=0) = a \tag{16a}$$

$$\epsilon \frac{dz}{dt} = -x - z, \quad x(t=0) = b \tag{16b}$$

Applying the MAE method described earlier, we summarize the results as follows. The outer solutions corresponding to Eqs. (3) and (4) are

$$x^{(0)}(t) = ae^{-t}, \quad z^{(0)}(t) = -ae^{-t} \tag{17a}$$

$$x^{(1)}(t) = [x^{(1)}(0) - at]e^{-t} \tag{17b}$$

$$z^{(1)}(t) = [-x^{(1)}(0) + at - a]e^{-t} \tag{17c}$$

The inner solutions corresponding to Eqs. (8) and (9) are

$$\bar{x}^{(0)}(\tau) = a, \quad \bar{z}^{(0)}(\tau) = -a + (a+b)e^{-\tau} \tag{18a}$$

$$\bar{x}^{(1)}(\tau) = (a+b) - a\tau - (a+b)e^{-\tau} \tag{18b}$$

$$\bar{z}^{(1)}(\tau) = -(2a+b) + a\tau + [2a + b + (a+b)\tau]e^{-\tau} \tag{18c}$$

Considering the two-term expansions only, the common solution (CS) for x is obtained as

$$(\mathrm{CS})_x = (x^i)^o = (x^o)^i \tag{19}$$

From Eq. (18), we obtain $(x^i)^o$, the outer expansion of the inner solution by first expressing the inner solution in the outer variable $t = \epsilon\tau$ and then expanding it around $\epsilon = 0$. Thus,

$$(x^i)^o = a(1-t) + \epsilon(a+b) \tag{20}$$

Next, from Eq. (17), we obtain $(x^o)^i$, the inner expansion of the outer solution, as

$$(x^o)^i = a(1-t) + \epsilon x^{(1)}(0) \tag{21}$$

Alternatively, in the present approach, we formulate $(x^o)^i$ as

$$(x^o)^i = x^{(0)}(t=0) + \epsilon[x^{(1)}(t=0) + \tau\dot{x}^{(0)}(t=0)]$$

$$= a(1-t) + \epsilon x^{(1)}(0) \tag{22}$$

Equating Eqs. (20) and (21), we get the value of the undetermined coefficient $x^{(1)}(0) = (a+b)$. Similarly, for z, we have

$$(\mathrm{CS})_z = (z^i)^o = (z^i)^o \tag{23}$$

From Eq. (18), we obtain $(z^i)^o$, the outer expansion of the inner solution, as

$$(z^i)^o = -a(1-t) + \epsilon[-(2a+b)] \tag{24}$$

Next, we obtain $(z^o)^i$, the inner expansion of the outer solution, as

$$(z^o)^i = -a(1-t) + \epsilon z^{(1)}(0) \tag{25}$$

Alternatively, in the improved method, we formulate $(z^o)^i$ as

$$(z^o)^i = z^{(0)}(t=0) + \epsilon\,[z^{(1)}(t=0) + \tau\dot{z}^{(0)}(t=0)]$$

$$= -a(1-t) + \epsilon z^{(1)}(0) \tag{26}$$

Using Eqs. (23-25), we get the value of the undetermined coefficient, $z^{(1)}(0)$, as

$$z^{(1)}(0) = -(2a+b) \tag{27}$$

The composite solution corresponding to Eq. (12) is

$$x_c(t,\epsilon) = ae^{-t} + \epsilon\,[(a+b)(e^{-t} - e^{-t/\epsilon}) - ate^{-t}] \tag{28a}$$

$$z_c(t,\epsilon) = -ae^{-t} + (a+b)(1+t)e^{-t/\epsilon}$$

$$+ \epsilon\,[(2a+b)(e^{-t/\epsilon} - e^{-t}) + ate^{-t}] \tag{28b}$$

Conclusion

In this Note, a critical examination of the method of matched asymptotic expansion has revealed that the terms of the common solution could be generated as polynomials in a stretched variable without actually solving for them, as is done presently. We have also seen that the common solution of the method of matched asymptotic expansion is the same as the intermediate solution of the singular perturbation method and, hence, these two methods give identical results. Two examples have been given for illustration.

Acknowledgment

This work was performed under Grant NAG-1-736.

References

[1]Vasielva, A.B., "Asymptotic Behavior of Solutions to Certain Problems Involving Nonlinear Differential Equations Containing a Small Parameter Multiplying the Highest Derivatives," *Russian Mathematical Surveys,* Vol. 18, 1963, pp. 13-81.

[2]Van Dyke, M., *Perturbation Methods in Fluid Mechanics,* Academic, New York, 1964.

[3]Cole, J. D., *Perturbation Methods in Applied Mathematics,* Blaisdell, Waltham, MA, 1968.

[4]O'Malley, R. E., Jr., *Introduction to Singular Perturbations,* Academic, New York, 1974.

[5]Naidu, D. S., *Singular Perturbation Methodology in Control Systems,* IEE Control Engineering Series, Peter Perigrinus Ltd., Stevenage, Herts, England, 1988.

Simple Scheme for the Integration of Stiff Differential Equations

M. B. Subrahmanyam*
University of Missouri, Columbia, Missouri

I. Introduction

STIFF differential equations arise out of modeling systems with widely differing time constants. In the case of nonlinear equations, the Jacobian has eigenvalues that are several orders of magnitude apart. In the case of an advanced

Received June 1, 1987; revision received Dec. 28, 1987.

*Associate Professor, Department of Electrical and Computer Engineering; currently, Electronics Engineer, Naval Air Development Center, Warminster, PA. Member AIAA.

aircraft, such systems arise, for example, when the actuator dynamics and the flexible body dynamics need to be incorporated into the flight control equations.

In this Note, we present an A-stable single-step scheme for the solution of initial-value problems associated with stiff nonlinear vector differential equations. The method is easy to program on a digital computer. It can be observed from the examples in Sec. IV that significant increments in the computation interval can be obtained by using this method. It is shown that the local truncation error of the method is $O(h^3)$, where h is the interval of computation.

We consider the following initial-value problem:

$$\dot{x}(t) = f[x(t),t] \tag{1}$$

$$x(t_0) = x_0 \tag{2}$$

We rewrite Eq. (1) as

$$\dot{x}(t) = A[x(t),t]\, x(t) + B[x(t),t] \tag{3}$$

where

$$A = \frac{\partial f}{\partial x} \text{ and } B = f - Ax$$

Thus, A is the Jacobian matrix associated with Eq. (1).

II. Numerical Procedure

Consider the interval $[t_i, t_{i+1}]$, where $t_i = t_0 + ih$, h being the interval of computation. We assume that A and B are twice continuously differentiable in x and t. We make two passes, and, on each pass, we assume that A and B are constant on $[t_i, t_{i+1}]$.

Let $\tau_i = t_i + (h/2)$ and $x^c(t_i)$ be the value of x at t_i, obtained from our numerical scheme.

First Pass

On $[t_i, t_i + 1]$, approximate $A[x(t),t]$ and $B[x(t),t]$ by

$$A = A[x^c(t_i), \tau_i], \qquad B = B[x^c(t_i), \tau_i] \tag{4}$$

Equation (3) is approximated by

$$\dot{y} = Ay + B, \qquad y(t_i) = \dot{x}^c(t_i) \tag{5}$$

Incorporating Eq. (5) into the Taylor series for $y(t_{i+1})$, we get

$$y(t_{i+1}) = \sum_{k=0}^{\infty} A^k x^c(t_i) \frac{h^k}{k!} + \sum_{k=0}^{\infty} A^k B \frac{h^{k+1}}{(k+1)!} \tag{6}$$

Disregarding a difference of magnitude $O(h^3)$, we have

$$y(t_{i+1}) = \left(I - \frac{Ah}{2}\right)^{-1} \left[\left(I + \frac{Ah}{2}\right) x^c(t_i) + Bh\right] \tag{7}$$

Second Pass

Let

$$\bar{x}_i = [x^c(t_i) + y(t_{i+1})]/2 \tag{8}$$

$$A = A(\bar{x}_i, \tau_i) \tag{9}$$

and

$$B = B(\bar{x}_i, \tau_i) \tag{10}$$

Again approximate Eq. (3) by

$$\dot{x}^c = Ax^c + B, \qquad x^c(t_i) \text{ given} \tag{11}$$

Using the same procedure as in the first pass, we get

$$x^c(t_{i+1}) = \sum_{k=0}^{\infty} A^k x^c(t_i) \frac{h^k}{k!} + \sum_{k=0}^{\infty} A^k B \frac{h^{k+1}}{(k+1)!} \tag{12}$$

Up to a difference of $O(h^3)$, we have

$$x^c(t_{i+1}) = \left(I - \frac{Ah}{2}\right)^{-1} \left[\left(I + \frac{Ah}{2}\right) x^c(t_i) + Bh\right] \tag{13}$$

Thus, the scheme is given by Eqs. (7), (8), and (13). We note that, in Eq. (4), τ_i can be replaced by t_i without affecting the accuracy of the algorithm. This will be demonstrated in the next section. However, when τ_i is used, only one pass is needed to get the desired accuracy in the case of time-varying linear systems.

It can be observed that Eqs. (7) and (13) contain a Padé approximation for the matrix exponential $\exp(Ah)$. Application of the scheme to the time-invariant equation $\dot{x} = Ax$ yields

$$x^c(t_{i+1}) = [I - (Ah/2)]^{-1} [I + (Ah/2)] x^c(t_i) \tag{14}$$

indicating that the scheme is A-stable.

III. Error Analysis

We now show that the numerical procedure given previously has a local truncation error of $O(h^3)$.

Theorem 3.1: Consider Eqs. (7), (8), and (13). Let $x(t_{i+1})$ be the true value at t_{i+1} and $x^c(t_{i+1})$ be the computed value at t_{i+1} with $x(t_i) = x^c(t_i)$. Then $\|x(t_{i+1}) - x^c(t_{i+1})\| = O(h^3)$.

Proof: We have

$$\begin{aligned} x(t_{i+1}) &= x(t_i) + \dot{x}(t_i)\, h + \ddot{x}(t_i)(h^2/2) + O(h^3) \\ &= x(\tau_i) + \dot{x}(\tau_i)\, h + O(h^3) \\ &= x(t_i) + \{A[x(\tau_i),\tau_i]\, x(\tau_i) + B[x(\tau_i),\tau_i]\}\, h + O(h^3) \end{aligned}$$

Since $x(\tau_i) = x(t_i) + \dot{x}(t_i)(h/2) + O(h^2)$,

$$\begin{aligned} x(\tau_i) &= x(t_i) + A[x(\tau_i),\tau_i]\, x(t_i)h + B[x(\tau_i),\tau_i]h \\ &\quad + A[x(\tau_i),\tau_i]\, A[x(t_i),t_i]\, x(t_i)(h^2/2) \\ &\quad + A[x(\tau_i),\tau_i]\, B[x(t_i),t_i](h^2/2) + O(h^3) \\ &= x(t_i) + A[x(\tau_i),\tau_i]\, x(t_i)\, h + B[x(\tau_i),\tau_i]h \\ &\quad + A^2[x(t_i),t_i]x(t_i)(h^2/2) \\ &\quad + A[x(t_i),t_i]\, B[x(t_i),t_i](h^2/2) + O(h^3) \end{aligned} \tag{15}$$

From our algorithm, after the first pass, we have

$$y(t_{i+1}) = x(t_i) + A[x(t_i),\tau_i]x(t_i)h + B[x(t_i),\tau_i]h + O(h^2)$$

So the intermediate value is given by

$$\begin{aligned} \bar{x}_i &= [x(t_i) + y(t_{i+1})]/2 = x(t_i) + A[x(t_i),\tau_i]x(t_i)(h/2) \\ &\quad + B[x(t_i),\tau_i](h/2) + O(h^2) \end{aligned} \tag{16}$$

After the second pass,

$$\begin{aligned} x^c(t_{i+1}) &= x(t_i) + A(\bar{x}_i,\tau_i)x(t_i)h + A^2(\bar{x}_i,\tau_i) + x(t_i)(h^2/2) \\ &\quad + B(\bar{x}_i,\tau_i)h + A(\bar{x}_i,\tau_i)\, B(\bar{x}_i,\tau_i)(h^2/2) + O(h^3) \end{aligned} \tag{17}$$

From Eqs. (15) and (17), we have

$$x(t_{i+1})-x^c(t_{i+1})=\{A[x(\tau_i),\tau_i]-A(\bar{x}_i,\tau_i)\}\,x(t_i)h + \{A^2[x(t_i),t_i]-A^2(\bar{x}_i,\tau_i)\}\,x(t_i)(h^2/2) + \{B[x(\tau_i),\tau_i]-B(\bar{x}_i,\tau_i)\}\,h + \{AB[x(t_i),t_i]-AB(\bar{x}_i,\tau_i)\}\,(h^2/2)+O(h^3) \quad (18)$$

We have

$$\|A[x(\tau_i),\tau_i]-A(\bar{x}_i,\tau_i)\|\le L_1\|x(\tau_i)-\bar{x}_i\| \quad (19)$$

Since

$$x(\tau_i)=x(t_i)+A[x(t_i),t_i]x(t_i)(h/2)+B[x(t_i),t_i](h/2)+O(h^2)$$

we have from Eq. (16),

$$x(\tau_i)-\bar{x}_i=\{A[x(t_i),t_i]-A[x(t_i),\tau_i]\}x(t_i)(h/2) + \{B[x(t_i),t_i]-B[x(t_i),\tau_i]\}(h/2)+O(h^2) = O(h^2)$$

Thus, the right-hand side of Eq. (19) is $O(h^2)$. Also,

$$\|A^2[x(t_i),t_i]-A^2(\bar{x}_i,\tau_i)\|\le L_2\,[\|x(t_i)-\bar{x}_i\|+h]=O(h) \quad (20)$$

Similarly,

$$B[x(\tau_i),\tau_i]-B(\bar{x}_i,\tau_i)=O(h^2) \quad (21)$$

and

$$AB[x(t_i),t_i]-AB(\bar{x}_i,\tau_i)=O(h) \quad (22)$$

Incorporating Eqs. (19–22) into Eq. (18), we get

$$\|x(t_{i+1})-x^c(t_{i+1})\|=O(h^3)$$

IV. Numerical Examples

All of the computations for the examples were performed on a VAX 11/730 computer in double precision. Our method will be compared to the Runge-Kutta methods in terms of stability, accuracy, and computation time. The relative error associated with x_p is denoted by R_p. For comparison purposes, we denote our algorithm by SA.

Example 1

An example from chemical reaction kinetics is given by[1]

$$\frac{dx}{dt}=f(x) \quad (23)$$

where

$$f_1=-0.013x_1+10^3x_1x_3$$

$$f_2=2500x_2x_3$$

$$f_3=0.013x_1-10^3x_1x_3-2500x_2x_3$$

The initial condition is given by $x(0)=[1\ 1\ 0]^T$. The range of integration for this problem is [0, 50].

This is a moderately stiff problem with real eigenvalues of the Jacobian, the eigenvalue with the largest absolute value being -3800 around $t=41$. We assumed that the reference solution is obtained using the second-order Runge-Kutta (RK2) method with $h=0.0001$.

Table 1 gives the execution times and relative errors for different methods. The results of the fourth-order Runge-Kutta (RK4) method with $h=0.0006$ were satisfactory. However, RK4 became unstable with $h=0.0008$.

Using an RK2 scheme, the results were satisfactory with $h=0.0004$. The method became unstable for $t\ge 41$ with $h=0.0005$.

It can be observed from Table 1 that, with our method, improvement in stability is obtained. The explicit methods RK4 and RK2 are inefficient in this case.

Example 2

This example is due to Robertson[2] and is taken from Chap. 6 of Ref. 3. The equations are given by

$$\dot{x}_1=-0.04x_1+10^4x_2x_3$$

$$\dot{x}_2=0.04x_1-10^4x_2x_3-3\times 10^7x_2^2$$

$$\dot{x}_3=3\times 10^7x_2^2$$

$$x_1(0)=1,\quad x_2(0)=0,\quad x_3(0)=0 \quad (24)$$

The interval of interest is [0, 40]. Numerical experiments on this example are reported in Chap. 6 of Ref. 3. This example represents the severest test for the methods, since, for large values of t, the eigenvalues approach 0, 0, and -10^4. Thus, RK4 would require $h<2.8\times 10^{-4}$ for stability.

With $h=0.001$, RK4 becomes unstable for $t\ge 16$. All of the explicit methods eventually become unstable with $h=0.001$ (see Table 6.5, p. 290, of Ref. 3). The results with our algorithm are presented in Table 2. Referring to Table 2 and Table 6.5 of Ref. 3, our method has the best performance of any explicit method.

From the examples, we make the following observations. Both RK4 and RK2 are highly accurate, however, the small step size required makes them inefficient for the solution of stiff systems.

Our method has the best stability behavior of any explicit method considered. It also has adequate accuracy and the execution times are not excessive. The ease of implementation of our method makes it attractive for practical use.

Table 1 Computational results for example 1

Method	h	$t=50$[a] R_1	R_2	R_3	Execution time to compute to $t=50$,s
SA	0.05	1.1×10^{-6}	4.9×10^{-7}	1.3×10^{-6}	25.75
SA	0.1	1.1×10^{-6}	4.7×10^{-7}	1.0×10^{-2}	16.29
SA	0.5	7.3×10^{-7}	1.8×10^{-6}	1.6×10^{-1}	8.85
RK4	0.0008	Unstable			
RK4	0.0006	2.5×10^{-8}	1.4×10^{-8}	5.3×10^{-8}	723.0
RK2	0.0005	Unstable			
RK2	0.0004	1.0×10^{-8}	2.1×10^{-8}	3.7×10^{-8}	126.0

[a]Reference values at $t=50$: $x_1=0.597654040E+00$; $x_2=0.140234411E+01$; $x_3=0.189338404E-05$.

Table 2 Computational results for example 2

Method	h	$t=40$[a] R_1	R_2	R_3	Execution time to compute to $t=40$,s
SA	0.05	9.9×10^{-6}	2.9×10^{-5}	2.4×10^{-5}	22.03
SA	0.0667	5.9×10^{-4}	1.8×10^{-3}	1.5×10^{-3}	18.47
RK4	0.001	Unstable			

[a]Reference values at $t=40$: $x_1=0.715827294E+00$; $x_2=0.918553530E-05$; $x_3=0.284163831E+00$.

References

[1]Gear, C. W., "The Automatic Integration of Stiff Ordinary Differential Equations," *Proceedings of IFIP Congress—1968,* North Holland, Amsterdam, 1969, pp. 187-193.

[2]Robertson, H. H., "The Solution of a Set of Reaction Rate Equations," *Numerical Analysis, An Introduction,* edited by J. Walsh, Academic, London, 1967, pp. 178-182.

[3]Lapidus, L. and Seinfeld, J. H., *Numerical Solution of Ordinary Differential Equations,* Academic, New York, 1971, pp. 267-293.

Eigenvector Derivatives of Repeated Eigenvalues Using Singular Value Decomposition

Kyong B. Lim,* Jer-Nan Juang,† and Peiman Ghaemmaghami‡
NASA Langley Research Center, Hampton, Virginia

Introduction

THE computational problem of eigenvector derivatives corresponding to repeated eigenvalues has not been treated adequately in the past, whereas several techniques are available for distinct eigenvalues.[1] The main difficulty appears to be the nonuniqueness of eigenvectors for repeated eigenvalues. For symmetric eigenvalue problems, a method[2] has been proposed for computing the eigenvector derivatives of repeated eigenvalues. A computational method has been proposed[3] recently for general nondefective matrices, although the results are inconclusive.

In this Note, explicit formula for the first-order eigenvector derivative corresponding to the eigenvector of a repeated eigenvalue is derived for the non-self-adjoint eigenvalue problem. The method applies to the class of nondefective problems whose first eigenvalue derivatives of the repeated eigenvalues are distinct. A singular value decomposition (SVD)[4] approach is used to compute four *required* bases for eigenspaces, and to keep track of the dimensions of state variables and the conditioning of the state equations.

Computation of Bases for Eigenspaces

Consider the nth order eigenvalue problem

$$(A - \lambda_i I)x_i = 0 \tag{1}$$

where A is assumed nondefective. Define the following notations

$$A_i \triangleq A - \lambda_i I, \quad r_i \triangleq \text{rank}(A_i), \quad \nu_i \triangleq n - r_i; \quad i = 1,\ldots,N$$

where N is the number of distinct eigenvalues and ν_i the dimension of the null space of A_i or, equivalently, the number of linearly independent eigenvectors associated with λ_i. Thus, for a simple eigenvalue, $\nu_i = 1$.

Although any linear combination of the multiple eigenvectors is an eigenvector, let $\{x_i^{(1)}, x_i^{(2)},\ldots, x_i^{(\nu_i)}\}$ be some choice of right eigenvectors corresponding to λ_i. Consider the SVD of A_i

$$A_i = [\tilde{U}_i U_i]\begin{bmatrix}\Sigma i \\ 0\end{bmatrix}\begin{bmatrix}\tilde{V}_i^H \\ V_i^H\end{bmatrix} = \tilde{U}_i \Sigma_i \tilde{V}_i^H \tag{2}$$

where $\Sigma_i = \text{diag}[\sigma_1,\ldots,\sigma_{r_i}]$, $[\tilde{U}_i U_i]$, and $[\tilde{V}_i V_i]$ denote the nonzero singular values, left and right singular vectors, respectively. The superscript H represents Hermitian transpose. The following relations hold:

$$U_i^H \tilde{U}_i = V_i^H \tilde{V}_i = 0, \quad \tilde{U}_i^H \tilde{U}_i = \tilde{V}_i^H \tilde{V}_i = I_{r_i}, \quad U_i^H U_i = V_i^H V_i = I_{\nu_i}$$
$$1 \le \nu_i \le n \tag{3}$$

The $(n \times \nu_i)$ matrix V^i is an orthonormal basis for the right eigenvectors so that the kth eigenvector corresponding to the ith eigenvalue can be expressed as

$$x_i^{(k)} = V_i \alpha_i^{(k)}; \quad k = 1,\ldots,\nu_i \tag{4}$$

$\alpha_i^{(k)}$ denotes the Kth eigenvector coefficients (or coordinates in V_i bases) corresponding to the ith eigenvalue. The SVD in Eq. (2) also provides a basis U_i^* for left eigenvectors. The superscript * means the complex conjugate. Furthermore, the SVD provides the bases for the right and left complimentary spaces that are required in the sequel.

Computation of Eigenvalue Sensitivity

To compute eigenvalue derivatives (needed to compute eigenvector derivatives), let us take the derivative of the eigenvalue problem

$$A_i x_i^{(k)} = 0 \tag{5}$$

to obtain

$$\dot{A}_i^{(k)} x_i^{(k)} = -A_i \dot{x}_i^{(k)} = -\tilde{U}_i \Sigma_i \tilde{V}_i^H \dot{x}_i^{(k)} \tag{6}$$

where

$$\dot{A}_i^{(k)} = \dot{A} - \dot{\lambda}_i^{(k)} I \tag{7}$$

Premultiplying Eq. (6) by the hermitian transpose of left singular matrix, with the aid of Eq. (3) produces

$$\begin{bmatrix}\tilde{u}_i^H \\ u_i^H\end{bmatrix}\dot{A}_i^{(k)} x_i^{(k)} = \begin{bmatrix}\Sigma_i \tilde{V}_i^H \\ O\end{bmatrix}\dot{x}_i^{(k)} \tag{8}$$

Using the expansion of Eq. (4), the last V_i equations in Eq. (8) can be rewritten as

$$[U_i^H \dot{A} V_i]\,\alpha_i^{(k)} = \dot{\lambda}_i^{(k)} U_i^H V_i \alpha_i^{(k)} \tag{9}$$

The original form of Eq. (9) has been derived in Ref. 5. In Eq. (9), $\dot{\lambda}_i^{(k)}$ and $\alpha_i^{(k)}$ are the unknowns, and hence a subeigenvalue problem of order ν_i is defined. The eigenvalue derivative $\dot{\lambda}_i^{(k)}$ is associated with a specific eigenvector.

For the case of *distinct eigenvalue derivatives*, there exists a unique set of ν_i linearly independent vectors $\{\alpha_i^{(1)},\ldots,\alpha_i^{(\nu_i)}\}$ that are determined to a scaling factor, depending on the normalization of corresponding eigenvectors in Eq. (4). Henceforth, the case of repeated eigenvalues with repeated first eigenvalue derivatives is excluded. It is important to note that Eq. (9) always produces ν_i eigenvalue derivatives corresponding to the ν_i repeated eigenvalues, independent of the choice of eigenvectors. For simple eigenvalues, i.e., $\nu_i = 1$, Eq. (9) reduces to the familiar scalar equation[1]

$$\dot{\lambda}_i = U_i^H \dot{A} V_i \tag{10}$$

Received Sept. 11, 1987; revision received Feb. 18, 1988.

*Engineering Specialist, PRC Kentron, Inc., Hampton, VA.
†Senior Research Scientist.
‡Assistant Professor, Old Dominion University, Norfolk, VA.