JOURNAL OF COMPUTATIONAL PHYSICS 37, 388–398 (1980)

An Automatic Error-Control Technique for Computation of Eigenlengths

P. NELSON, JR.

Department of Mathematics, Texas Tech University, Lubbock, Texas 79409

AND

A. K. RAY*

Department of Physics, Texas Tech University, Lubbock, Texas 79409

Received April 27, 1979; revised December 10, 1979

An automatic error-control technique is presented for the calculation of eigenlengths of systems of ordinary differential equations. The method appears to be computationally very stable with significant savings in computing time and effort when compared with conventional initial-value codes. Numerical examples are presented which demonstrate the efficacy of the method.

1. INTRODUCTION

Consider a linear first-order ordinary differential system in $2n$ dependent variables, subject to two-point boundary conditions which specify n of the dependent variables at one endpoint and the complementary set of n variables at the other endpoint. Such a two-point boundary-value problem (TPBVP) of the second kind is well known to have a natural association with a certain Riccati initial-value problem for an $n \times n$ matrix. If the left endpoint is located at $x = 0$, the eigenlength can be defined as a value of the right endpoint for which the associated homogeneous differential system subject to homogeneous boundary conditions of the second kind has a nontrivial solution. The smallest eigenlength [1] is identical with the right endpoint of the maximal interval of existence for the associated Riccati initial-value problem. The term "integration-to-blowup" is popularly used to describe the use of this fact to determine the smallest eigenlength by numerically integrating the Riccati initial-value problem until the matrix of dependent variables becomes large in some appropriate sense. Boland and Nelson [2] have obtained results which go far toward explaining

* Present address: Quantum Theory Project, University of Florida, Gainesville, Fla. 32611.

0021-9991/80/120388-11$02.00/0

the frequently observed good accuracy of integration-to-blowup and which can be used to estimate this accuracy in any particular application, such as calculation of eigenlengths. The calculation of eigenlengths is one of the fundamental computational problems associated with nuclear reactors, and it has applications also in the areas of structural mechanics, quantum mechanics [3], and nuclear physics [4].

Most modern initial-value codes provide automatic selection of step-size based upon specified local error tolerances, usually some combination of absolute and relative error. When such a code is used for the numerical solution of the Riccati initial-value problem, the step-size tends to become intolerably small as the independent variable approaches the critical length. In [2] this problem was circumvented by using a minimum allowable step-size. However, this ad hoc device is ultimately unsatisfactory in that there is no clear-cut relation between the minimum step-size and the accuracy of the final answer. In the method we will describe in this paper, this problem is alleviated by using appropriate types of local error criteria. We will call this the square-error method and a tentative step in this method will be accepted or rejected depending on the estimated local truncation error per unit step in the dependent variable. This local truncation error will be required to be less than some multiple of (a measure of) the square of the dependent variable, hence the name "square error." In this paper, we will apply this method to the computation of eigenlengths. In Section 2, we introduce the basic ideas and the theorems, and in Section 3, actual numerical examples are presented and compared with the results obtained by conventional methods. The results presented here suggest the square-error method can lead to efficient and effective computation of eigenlengths within the range of a prespecified error tolerance. Section 4 contains a few concluding remarks and suggestions for further research.

2. Theory

Consider the linear two-point boundary-value problem (TPBVP) defined by the differential system

$$u'(z) = A(z)\,u(z) + B(z)\,v(z), \tag{2.1a}$$

$$-v'(z) = C(z)\,u(z) + D(z)\,v(z), \tag{2.1b}$$

subject to

$$u(0) = 0, \tag{2.2a}$$

$$v(x) = 0, \tag{2.2b}$$

where u and v are n-vectors and the matrices A, B, C, and D are piecewise continuous on $[0, \infty)$. If $x > 0$ is such that there exists nonzero u, v satisfying (2.1) and (2.2), then x is termed an eigenlength of the differential system (2.1). Let x_c denote the smallest eigenlength of this TPBVP ($x_c = +\infty$ if no finite critical length exists).

Then [1], x_c is the right endpoint of the maximal interval of existence of the solution of the Riccati initial-value problem

$$R'(z) = B(z) + A(z)\,R(z) + R(z)\,D(z) + R(z)\,C(z)\,R(z), \tag{2.3a}$$

$$R(0) = 0. \tag{2.3b}$$

In view of the above, it is natural to consider computing x_c by numerical integration of (2.3) and approximating x_c as the value of z at which $R(z)$ or $R'(z)$ exceeds some specified magnitude [5, 6]. This is the technique known as "integration-to-blowup" and several workers [7–9] have reported good computational success in applying this method to vector systems.

Boland and Nelson [2] have considered neutron transport problems where the coefficient matrices have the special properties

$$B(z), C(z) * \geqslant * 0, \tag{2.4a}$$

$$A(z), D(z) \cdot \geqslant \cdot 0, \tag{2.4b}$$

which means [12] that the matrices $B(z)$ and $C(z)$ have nonnegative entries and that the matrices $A(z)$ and $D(z)$ have nonnegative entries except possibly along their diagonal. Under these conditions, it is shown in [2] that the inequalities (2.4a) and (2.4b) imply certain lower bounds on the matrix R and these bounds provide information for the problem of estimating x_c from the results obtained by numerical integration of (2.3a) and (2.3b). Before we present actual computational examples, we now state some theorems [2].

THEOREM 1. *Suppose* $a_0(z) = \min\{\sum_{i=1}^{n} A_{ij}(z): 1 \leqslant j \leqslant n\}$

$$a_0 = \inf\{a_0(z): 0 \leqslant z < \infty\},$$

and similarly b_0, c_0, *and* d_0. *If* R *is the solution of the Riccati equation, we define*

$$r_j(z) = \sum_{i=1}^{n} R_{ij}(z), \qquad r(z) = \min\{r_j(z): 1 \leqslant j \leqslant n\}.$$

Then, if $x_0 \in [0, x_c)$ *and* $r_0 = r(x_0)$, *the inequality*

$$r(z) \geqslant r_0 + \int_{z_0}^{z} [b_0 + (a_0 + d_0)\,r(s) + c_0 r^2(s)]\,ds$$

is satisfied for $z \in [x_0, x_c)$.

THEOREM 2. *Suppose* $c_0 > 0$. *Let* y_0 *be the maximum of zero,* $-(a_0 + d_0)/2c_0$, *and the largest real root of the quadratic equation* $c_0x^2 + (a_0 + d_0)x + b_0 = 0$ (*provided such roots exist; note the correction of the definition of* y_0 *as given in* [2].)

Also, we define

$$\alpha = \frac{4b_0c_0 - (a_0 + d_0)^2}{4c_0^2}, \qquad \beta = \frac{a_0 + d_0}{2c_0}.$$

Suppose also that, in the notation of Theorem 1, *the inequality* $r_0 > y_0$ *holds. For* $z \in [x_0, x_c)$, $r(z)$ *is not less than* $f(z)$, *where* $f(z)$ *is defined by the following equalities in the indicated parameter ranges:*

$$\tan^{-1}\left[\frac{f(z)+\beta}{\alpha^{1/2}}\right] = \tan^{-1}\left[\frac{r_0+\beta}{\alpha^{1/2}}\right] + c_0\alpha^{1/2}(z - x_0), \qquad \alpha > 0$$

$$-\frac{1}{f(z)+\beta} = -\frac{1}{r_0+\beta} + c_0(z - x_0), \qquad \alpha = 0$$

$$\ln\left[\frac{f(z)+\beta-(-\alpha)^{1/2}}{f(z)+\beta+(-\alpha)^{1/2}}\right] = \ln\left[\frac{r_0+\beta-(-\alpha)^{1/2}}{r_0+\beta+(-\alpha)^{1/2}}\right] + 2c_0(-\alpha)^{1/2}(z - x_0), \qquad \alpha < 0.$$

This theorem gives a lower bound for R, from which upper bounds for x_c can be determined. Two important corollaries follow from the above theorem.

COROLLARY 1. *If* $c_0 > 0$ *and* $y_0 = -1$, *then* $x_c < \infty$.

COROLLARY 2. *Under the assumptions of Theorem* 2, *the following bounds hold in the indicated parameter ranges:*

$$x_c \leqslant x_0 + \frac{1}{c_0\alpha^{1/2}}\left\{\frac{\pi}{2} - \tan^{-1}\left[\frac{r_0+\beta}{\alpha^{1/2}}\right]\right\}, \qquad \alpha > 0,$$

$$x_c \leqslant x_0 + \{c_0[r_0+\beta]^{-1}\}, \qquad \alpha = 0,$$

$$x_c \leqslant x_0 - \frac{1}{2c_0(-\alpha)^{1/2}} \ln\left[\frac{r_0+\beta-(-\alpha)^{1/2}}{r_0+\beta+(-\alpha)^{1/2}}\right], \qquad \alpha < 0.$$

These bounds are the fundamental results which we propose to use in actual numerical examples to compute eigenlengths.

For scalar problems ($n = 1$), Scott [10, Exercise 1, p. 153] observes that a bilinear transformation of dependent variables of the form

$$\tilde{R} = (R - 1)(R + 1)^{-1}$$

in principle can be used to compute eigenlengths as zeros of $\tilde{R} - 1$. This basic idea is very appealing, but it is not clear how it can be extended to vector ($v > 1$) problems. The simplest version of such transformations is perhaps the inverse transformation $\tilde{R} = S = R^{-1}$, but the authors in joint work with Wing [11] have shown that this approach can fail rather dramatically for vector problems.

3. Numerical Examples

The basic computational procedure is the numerical integration of the initial-value problem (2.3a) and (2.3b), up to the point $z = x_0$ that the difference between x_0 and the upper bound for x_0 as obtained from Corollary 2 is within the desired accuracy. The initial-value code used is a modified double-precision version of the Runge–Kutta–Fehlberg (RKF) code of Shampine and Allen [13]. This is a fourth-order Runge–Kutta-type code in which two additional function evaluations provide an estimate of the local truncation error for each tentative step. Such a step is accepted if the condition

$$|\delta y_i| \leqslant \varepsilon_a + \varepsilon_r |y_i| \tag{3.1a}$$

is satisfied for each component y_i of the vector of dependent variables, where δy_i is the estimated local truncation error per unit step in y_i and ε_a and ε_r are respectively absolute and relative error tolerances. If (3.1a) does not hold, then the tentative step is rejected and the error estimate is used to compute a new step-size for which the error criterion should be satisfied. At the end of a successful step the local error estimate is used to increase the size of the next tentative step to near the maximum value for which it is estimated that (3.1a) will hold. The latter feature ensures the integration will proceed with approximately maximal efficiency compatible with the specified error criterion.

In our numerical examples, apart from a straightforward application of RKF, we used the square-eror criterion. This consists of replacing (3.1a) by the error criterion

$$|\delta R_{ij}| \leqslant \varepsilon_a + \varepsilon_2 c_0 r_0 R_{ij}, \tag{3.1b}$$

where the absolute-error tolerance ε_a and square-error tolerance ε_2 are provided. The basic idea suggesting this error criterion is as follows. Near the eigenlength the second term on the right-hand side of (3.1b) dominates, from which we conclude

$$\delta r_j \lesssim \varepsilon_2 c_0 r_0 r_j.$$

But near the eigenlength the quadratic term in (2.3a) dominates, and from this it is possible to conclude

$$\frac{dr_j}{dx} \gtrsim c_0 r_0 r_j.$$

If we replace the left-hand side of the latter by $\delta r_j / \delta x$, then we conclude that (3.1b) implies

$$\delta x \lesssim \varepsilon_2.$$

Our expectation is that the error criterion (3.1b) will permit control of the relative error in computed eigenlengths. As our examples will clearly indicate, this method

turns out to be consistently superior to the conventional RKF (in the context of integration-to-blowup). All calculations were carried out in double-precision arithmetic on an IBM 370, Model 145 at Texas Tech University.

EXAMPLE 1. Consider the scalar problem

$$u' = v, \tag{3.2a}$$

$$-v' = u, \tag{3.2b}$$

$$u(0) = v(x) = 0. \tag{3.2c}$$

The associated Riccati initial-value problem is

$$R' = 1 + R^2, \tag{3.3a}$$

$$R(0) = 0. \tag{3.3b}$$

This problem has the well-known solution $R(z) = \tan(z)$, with the corresponding value $x_c = \pi/2 \approx 1.57079633$. For scalar systems with constant coefficients, the estimates of Corollary 2 are exact. We present this problem to illustrate the claim that the technique of integration-to-blowup is capable of considerable accuracy, and also to indicate the superiority of the square-error method to the conventional RKF in the computation of eigenlengths.

Table I contains the computed values of x_c for different values of the accuracy ε required in the computation of the eigenlength. In all cases, the relative-error tolerance ε_r and the absolute-error tolerance ε_a (or square-error tolerance ε_2) were taken to be the same as ε. We note that for values of ε ranging from 10^{-3} to 10^{-8},

TABLE I

Relative Errors in x_c (=(Exact-Computed)/Exact) and Required Function Evaluations for Example 1 with Various Error Tolerances and Integration-to-Blowup Procedures

	RKF		Square Error	
ε	No. of function evaluations	Relative error in x_c	No. of function evaluations	Relative error in x_c
1.0(−3)	287	0.71(−3)	147	0.62(−3)
1.0(−4)	849	1.20(−4)	348	1.49(−4)
1.0(−5)	2868	1.19(−5)	427	1.24(−5)
1.0(−6)	9460	1.34(−6)	964	1.26(−6)
1.0(−7)	30641	0.85(−7)	2105	0.90(−7)
1.0(−8)	97624	0.69(−8)	4414	0.80(−8)
1.0(−9)	—	—	8969	0.76(−9)

the square-error method consistently requires significantly fewer function evaluations, while keeping the values of x_c well within the desired accuracy. Also, though not indicated in Table I, the actual computer time involved in the square-error method is significantly smaller than the conventional RKF method. For example, for $\varepsilon = \varepsilon_a = \varepsilon_r = 10^{-8}$, the conventional method took 2 min 40.68 sec, while the square-error method required only 19.96 sec. For $\varepsilon = 10^{-9}$, x_c was not even computed using the conveentional RKF method because of the excessive computer-time requirements. Overall, the results seem to support amply our contention about the superiority of the square-error method.

EXAMPLE 2. This is the problem of the form (2.3a) and (2.3b) with coefficient matrices defined by

$$B = C = A + I = D + I = \begin{pmatrix} 0 & 0.4 \\ 0.025 & 0.15 \end{pmatrix}, \qquad 0 \leqslant z < 5$$

$$B = C = A + I = D + I = \begin{pmatrix} 0 & 0.275 \\ 0.5 & 0.3 \end{pmatrix}, \qquad 5 < z.$$

The problem thus formed is a crude one-dimensional two-energy-group model of a fast nuclear reactor, with the regions $0 \leqslant z \leqslant 5$ and $z > 5$ representing respectively the blanket and the core. In [1], the eigenlength was computed analytically, up to a transcendental equation in x_c which was solved numerically, via the bisection method, on a Wang 2200 programmable calculator. The result was $x_c = 12.49498546$, which is to be compared with our results using the method of integration-to-blowup with different error criteria.

Table II contains the computed values of x_c within an accuracy ε requested in the computation of eigenlengths. As in Example 1, the square error ε_2, the relative error ε_r, and the absolute error ε_a were taken to be the same as ε. Guided by our results for Example 1, only three different values of ε, namely, 10^{-3}, 10^{-6}, and 10^{-8}, were tried. Again we note that consistently the number of function evaluations for the square-error method is significantly smaller than for the conventional RKF method. However, within the desired accuracy, the values of x_c obtained via the square-error

TABLE II

Relative Errors in x_c and Required Function Evaluations for Example 2 with Various Error Tolerances and Integration-to-Blowup Procedures

ε	Function evaluations for RKF	Relative error in x_c for RKF	Function evaluations for square error	Relative error in x_c for square error
1.0(−3)	381	0.91(−4)	215	0.85(−5)
1.0(−6)	9908	0.89(−6)	1263	1.47(−6)
1.0(−8)	100546	0.72(−8)	5308	0.64(−8)

method compare very favorably with the values of x_c obtained via RKF. In fact, for $\varepsilon = 10^{-8}$, the values of x_c for the two methods differ only in the eighth decimal place. Also, as in Example 1, the actual computer time involved is significantly smaller in the square-error method. For example, for $\varepsilon = 10^{-8}$, the square-error method took 58.95 sec while the convenional RKF required 11 min 43.19 sec. Overall, this example again provides a definite indication of the superiority of the square-error method.

EXAMPLE 3. Consider the equation

$$(\operatorname{sgn} s)\frac{\partial n}{\partial z}(z, s) + a(s)\, n(z, s) = \lambda k(s) \int_{-1}^{1} n(z, s')\, ds', \qquad 0 \leqslant z \leqslant x \tag{3.4a}$$

subject to the conditions

$$n(0, s) = 0, \qquad 0 < s < 1, \tag{3.4b}$$

$$n(x, s) = 0, \qquad -1 < s < 0, \tag{3.4c}$$

where $a(s)$ and $k(s)$ are real piecewise continuous functions on $|s| \leqslant 1$. For fixed λ, we want to compute the interval lengths x such that (3.4) has a nontrivial solution. This type of equation arises in the study of particle transport in a slab [14–16].

If we make the substitutions

$$u(z, s) = n(z, s), \qquad s > 0 \tag{3.5a}$$

$$v(z, s) = n(z, s), \qquad s < 0, \tag{3.5b}$$

then (3.4) can be written as

$$\frac{\partial u}{\partial x}(z, s) + a(s)\, u(z, s) = \lambda k(s) \left\{ \int_{0}^{1} u(z, s')\, ds' + \int_{-1}^{0} v(z, s')\, ds' \right\}, \tag{3.6a}$$

$$-\frac{\partial v}{\partial z}(z, s) + a(s)\, v(z, s) = \lambda k(s) \left\{ \int_{0}^{1} u(z, s')\, ds' + \int_{-1}^{0} v(z, s')\, ds' \right\}, \tag{3.6b}$$

$$u(0, s) = 0, \tag{3.6c}$$

$$v(x, s) = 0. \tag{3.6d}$$

The integrals can be replaced with some type of numerical quadrature scheme, typically a Gauss–Legendre quadrature. Then (3.6) becomes a system of ordinary differential equations of the form

$$u' = Au + Bv, \tag{3.7a}$$

$$-v' = Cu + Dv, \tag{3.7b}$$

$$u(0) = 0, \tag{3.7c}$$

$$v(x) = 0. \tag{3.7d}$$

TABLE III

Results for x_c for Different Values of λ

λ	Scott [18] x_c	Wing [19] x_c	Allen [17] x_c
2	2.9617	2.97266	NR[a]
5	1.0744	1.08398	1.1292
10	0.52000	0.52930	0.5452
20	0.25581	0.26562	0.2680
30	0.16962	NR[a]	0.1766
40	0.12687	0.13672	0.1318

[a] NR indicates that no results were given for this case.

We shall consider one example of this class of problem. This has been studied by Allen [17], Scott [18], and Wing [19]. Let

$$a(s) = |s|,$$
$$k(s) = e^{-5|s|}.$$

Apart from investigating the application of the square-error method to this type of pseudo-transport problem, the results for this problem should be of special interest because of somewhat conflicting reports in the literature. Table III presents the quoted results in the literature; we note that the results for eigenlengths in [17] tend to be slightly higher than those in Ref. [19], while the results in Ref. [19] are higher than the corresponding results in Ref. [18]. One possible source of discrepancy could be due to the quadrature scheme and also the number of quadrature points used in replacing the integrals in (3.6). Only in Ref. [17], it is mentioned that a eight-point Gauss formula was used to approximate the quadratures.

To investigate the effect of the number of quadrature points, we used a Gauss–Legendre quadrature scheme and varied the number of quadrature points from 2 to 16 for each value of λ. As the number of quadrature points was increased from 2 to 4, the value of x_c dropped considerably for all values of λ and for both the conventional RKF method and the square-error method. However, both methods gave essentially the same value of x_c, the primary difference being that the square-error method required considerably fewer function evaluations and less computing time. For example, for $\lambda = 2$ with two quadrature points, the conventional RKF method required 29 min 9.87 sec with $\varepsilon = 10^{-8}$, whereas the square-error method required 1 min 42.31 sec. As in Examples 1 and 2, this again indicates the superiority of the square-error method. As the number of quadrature points strongly influenced computed values of x_c, we then increased the number of quadrature points until convergence in the values of x_c was obtained within the accuracy requested. This was done only for the square-error method because of excessive time requirements in using RKF and also because the computed values of x_c were essentially identical for the two methods.

TABLE IV

Computed Values of x_c Using the Square-Error Method with $\varepsilon = \varepsilon_a = \varepsilon_r = 10^{-3}$

λ	No. of quadrature points	x_c
2	8	2.9615
	16	2.9612
5	8	1.0743
	16	1.0743
10	8	0.5200
	16	0.5200
20	8	0.2558
	16	0.2558
30	8	0.1696
	16	0.1696
40	8	0.1269
	16	0.1268

The results are indicated in Table IV which contains the computed values of x_c for the square-error method using 8 and 16 quadrature points for each values of λ. Note that the values of x_c are identical, within the requested accuracy, with those reported in Ref. [18].

4. Concluding Remarks

In this paper we have presented a computational technique for approximating the critical or the eigenlength for ordinary differential equations within a specified tolerance. As our examples clearly indicate, this automatic error-control technique is quite superior to the conventional method and seems capable of handling both scalar and vector problems. There is significant savings in computing effort and time when this new technique is used in determining critical lengths.

There are a number of features of the technique which warrant further study. One possible source of further research would be to compare the square-error method with the arclength method [20, Especially Chap. 12]. This method involves a preliminary transformation of the Riccati equation so that the arclength becomes the independent variable. One of the authors has conducted some preliminary experiments and the results seem to indicate that the arclength procedures are superior to square error for fairly large error tolerances, but that the square-error procedure becomes preferable as the specified error-tolerance decreases. It is hoped that this can be pursued further and reported elsewhere. (We note parenthetically that implementation of the arclength method will require somewhat more effort than is obvious at first thought. The difficulty occurs in problems having discontinuities in the coefficient matrices (e.g.,

Example 2). The arclength transformation changes these from discontinuities in the independent variable to discontinuities in the dependent variables, and the latter are substantially more difficult to integrate numerically. In fact, Shampine, Watts, and Davenport [21] indicate that the type of code used in our work cannot successfully integrate such problems).

In our work, we have basically used a Runge–Kutta-type code. Other basic integration algorithms should also be considered for possible extension of this study, primarily because recent studies [21] have indicated that Runge–Kutta-type codes probably should not be used when extremely high accuracy is sought. The eventual aim of all these would be to produce an efficient, effective, and robust user-oriented subroutine for computing critical lengths, with the user required to supply only an error tolerance, aside from the quantities necessary to define the problem. Further research in this direction could be very rewarding.

REFERENCES

1. P. NELSON AND I. T. ELDER, *SIAM J. Numer. Anal.* **14** (1977), 124.
2. W. R. BOLAND AND P. NELSON, *Appl. Math. Comput.* **1** (1975), 67.
3. J. M. CALVERT AND W. D. DAVISON, *J. Phys. A* **2** (1969), 278.
4. F. V. ATKINSON, "Discrete and Continuous Boundary Problems," Academic Press, New York, 1964.
5. R. BELLMAN, H. KAGIWADA, AND R. KALABA, *Comm. ACM.* **9** (1966), 522.
6. J. CASTI, R. KALABA, AND M. R. SCOTT, *J. Math. Anal. Apl.* **41** (1973), 1.
7. R. BELLMAN, H. KAGIWADA, R. KALABA, AND S. UENO, *J. Mathematical Phys.* **9** (1968), 906.
8. H. H. KAGIWADA, R. KALABA, AND B. J. VEREEKE, *J. Approximation Theory* **1** (1968), 355.
9. H. H. KAGIWADA AND R. KALABA, *J. ACM* **17** (1970), 412.
10. M. R. SCOTT, "Invariant Imbeddings and its Applications to Ordinary Differential Equations," Addison–Wesley, Reading, Mass., 1973.
11. P. NELSON, A. K. RAY, AND G. M. WING, *J. Math. Anal. Appl.* **65** (1978), 201
12. W. T. REID, "Riccati Differential Equations," Academic Press, New York, 1972.
13. L. F. SHAMPINE AND R. C. ALLEN, "Numerical Computing," Saunders, Philadelphia, 1973.
14. B. G. CARLSON AND G. I. BELL, *Proc. Intern. Conf. Peaceful Uses of Atomic Energy, Geneva* **16** (1958), 535.
15. P. NELSON AND M. R. SCOTT, *J. Math. Anal. Appl.* **34** (1971), 628.
16. M. R. SCOTT, *J. Math. Anal. Appl.* **28** (1969), 112.
17. R. C. ALLEN, *J. Math. Anal. Appl.* **39** (1971), 48.
18. M. R. SCOTT, *J. Comput. Phys.* **12** (1973), 334.
19. G. M. WING, "Transport Theory," SIAM–AMS Proceedings (R. Bellman *et al.*, Eds.), Vol. 1, p. 159, Amer. Math. Soc., Providence, R.I., 1969.
20. L. F. SHAMPIRE AND M. K. GORDON, "Computer Solution of Ordinary Differential Equations: The Initial Value Problem," Freeman, San Francisco, 1975.
21. L. F. SHAMPINE, H. A. WATTS, AND S. M. DAVENPORT, *SIAM Rev.* **18** (1976), 376.