Sangchul Lee
John L. Junkins
Texas A&M University
Aerospace Engineering Department
College Station, TX 77843-3141

Construction of Benchmark Problems for Solution of Ordinary Differential Equations

An inverse method is introduced to construct benchmark problems for the numerical solution of initial value problems. Benchmark problems constructed in this fashion have a known exact solution, even though analytical solutions are generally not obtainable. The process leading to the exact solution makes use of an initially available approximate numerical solution. A smooth interpolation of the approximate solution is forced to exactly satisfy the differential equation by analytically deriving a small forcing function to absorb all of the errors in the interpolated approximate solution. Using this special case exact solution, it is possible to directly investigate the relationship between global errors of a candidate numerical solution process and the associated tuning parameters for a given code and a given problem. Under the assumption that the original differential equation is well-posed with respect to the small perturbations, we thereby obtain valuable information about the optimal choice of the tuning parameters and the achievable accuracy of the numerical solution. Five illustrative examples are presented. © *1994 John Wiley & Sons, Inc.*

INTRODUCTION

We consider the initial value problem for linear or nonlinear ordinary differential equations. In general, we do not know the true solution and any numerical method gives us an approximate solution; the numerical solutions generally contain two sources of error, round-off and truncation (Gear, 1971). We must somehow evaluate the accuracy of a given approximate solution, typically without knowing the true solution. The most common way of assessing the true error of a numerical solution is to reduce some tolerance parameter, integrate again, and compare the results (Hairer et al., 1987; Shampine, 1987). Although more sophisticated error analyses can be conducted, there is no general way to absolutely guarantee the final accuracy of the solutions. This does not preclude obtaining practical solutions for most applications, but it remains very difficult to answer subtle questions.

Many numerical methods are available for solving initial value problems. Early numerical methods were merely fixed step size implementations and these methods were straightforward to implement, but the results were often inconclusive. In the 1960s, research on numerical methods for highly nonlinear initial value problems led to adaptive methods that could automatically vary the step size and/or the order of the method to match a user-specified local error tolerance at each step. This work led to the current generation of numerical methods. Due the presence of round-off error, it is common to find that accu-

Received October 10, 1993; Accepted April 25, 1994.
Shock and Vibration, Vol. 1, No. 5, pp. 403–414 (1994)

CCC 1070-9622/94/050403-12

racy improves until step sizes or tolerances are decreased below some critical value; the accuracy then degrades while solution costs increase (Gear, 1971; Shampine, 1974). Shampine (1974, 1980) pointed out that a typical adaptive code will not quit when impossible accuracies are specified. He also reported that the standard ways to assess true errors may lead to wrong conclusions even using the best codes available at that time. Shampine (1974) considered a machine dependent limit on the step size and one on the local error tolerance, and he suggested a way of automatically selecting an initial step size that appears to be reliable and reasonably efficient (Shampine, 1978). Enright (1989) pointed out that the relationship between the accuracy obtained and the specified tolerances is generally extremely sensitive to both the problem and the method. In particular, for Runge–Kutta methods with interpolants, he proposed an error and step size control mechanism based on monitoring and controlling the defect of a continuous approximation rather than the local error of the discrete approximation.

In view of the historical and recent developments, we observe that the theory of differential equation solvers is far from complete, so that the understanding of a given code's performance invariably requires a study of experimental results. Hull, et al. (1972) and Krogh (1973) provided two outstanding collections of test problems for this purpose. These test problems have been used in the development and testing of many codes and can be regarded as standard benchmark problems for initial value problem solvers. Whenever we know the true solutions of a test problem, however, we can investigate the relationship between the true, or global error and the tuning parameters of a given code (e.g., step size, local error tolerance, order, etc.). The relationship between the behavior of an algorithm on a benchmark problem and the behavior of the algorithm on a problem of interest is difficult to establish. Because the problem of interest is almost never exactly solvable, we need a means to establish a customized benchmark problem that is a close neighbor of any given problem of interest. We introduce here a broadly applicable inverse method that constructs a neighbor of a given numerical approximate solution; the neighboring problem does in fact exactly satisfy the original differential equations (with a known, small forcing function) and serves as an excellent benchmark problem. More specifically, we present a broadly useful approach to construct a benchmark problem near the problem of interest in a particular application. By virtue of the fact that the benchmark problem is a customized near neighbor of the problem of interest, we show that numerical convergence studies on the benchmark problem are directly useful in algorithm selection, tuning, and accuracy validation.

The difficulties mentioned earlier result from not knowing the true solution. What happens if we are able to construct a problem-dependent "exact" benchmark problem? First we can easily investigate the true error/parameter relationship and find the limiting precision and associated values of critical parameters of a given code. Second, the problem of how to assess global error vanishes automatically. Finally, we have an absolute standard to find which method is most suitable for an important member of our particular family of problems. The sensitivity of the accuracy/tolerance relation of a given method is primarily a result of the heuristics used to monitor the local error and control the step size. If we do not know the true solution, then it is very hard to assess which method is the best for a class of problems because of the high sensitivity of accuracy to variations in step size control logic. The remaining and most critical question is: How useful is the convergence and accuracy information obtained for the exactly solved benchmark problem, in regard to drawing conclusions for the (neighboring) original problem? It is important to recall that the benchmark problem includes a regular perturbation to the original problem. If the perturbation is small enough, it is to be expected that all derivatives will be close for the two problems and consequently, the behavior of standard discrete variable methods will be similar both with respect to accuracy and stability. It is certainly true that there are open questions on this issue needing further investigation; however, by constructing a family of neighboring benchmark problems, it is usually possible to judge the size of the neighborhood in which the convergence and accuracy properties are relatively invariant with respect to the perturbation. Several applications presented herein provide strong evidence supporting the practicality of this approach.

In this study we propose a method to construct a benchmark problem that is a close neighbor of a given approximate solution of the original problem. The benchmark problem is constructed so that it satisfies exactly the differ-

ential equation but with a known, usually small, time varying forcing function. We can investigate the global error/parameter relationship of the benchmark problem with the true solution in hand. Under the assumption that the original problem is well-posed with respect to small perturbations, we have valuable information about the optimal parameters and the accuracy of the numerical solution. Actually the stability assumption is not so severe because any numerical method needs it more or less to obtain reliable solutions. Also, by introducing several neighboring approximate solutions with initial condition and parameter variations, then repeating the entire process, it is possible to experimentally establish insight on the size of the region over which the convergence properties are invariant.

Lee and Junkins (1993) presented two computer codes for first order and second order systems of differential equations, when the classical Runge–Kutta fourth order method with a fixed step size was used. An illustrations, we show the utility of these codes for two simple nonstiff problems. When we use the IMSL (1989) subroutines DIVPRK and DIVPBS as solvers, we show the utility of this methodology for two celestial mechanics problems (Krogh, 1973) that have been used as test problems several times in the literature. Subroutine DIVPRK uses the Runge–Kutta formulas of order five and six developed by J. H. Verner. Subroutine DIVPBS uses the Bulirsh–Stoer extrapolation method and will terminate when impossible accuracies are specified. In the fifth example, we consider a typical stiff problem and discuss some limitations and restrictions of this methodology.

CONSTRUCTION OF EXACT BENCHMARK PROBLEMS

We want to construct new differential equations that are slightly perturbed versions of the original differential equations. For these new differential equations, we can establish the true analytical solution using an algebraic inverse idea. Then we can investigate the error/tolerance relationship with an absolute standard. Under local stability assumptions, we have valuable information about the optimal parameters and the accuracy of the particular numerical solution for the given original differential equations. The stability assumption is easily validated by constructing some neighboring benchmark problems.

Here we introduce one way for constructing exact benchmark problems. We take a global approach for the perturbation term instead of a piecewise polynomial perturbation to avoid the lack of smoothness at break points. First we consider the following two distinct initial value problems:

$$\dot{x} = f_1(x, t), \quad x(t_0) = x_0 \quad \text{over } t_0 \le t \le t_f \qquad f_1: R^N \times R - R^N \tag{1}$$

$$\ddot{x} = f_2(x, \dot{x}, t), \quad x(t_0) = x_0, \quad \dot{x}(t_0) = \dot{x}_0 \quad \text{over } t_0 \le t \le t_f \qquad f_2: R^N \times R^N \times R \to R^N. \tag{2}$$

A candidate discrete approximate solution can be obtained from the original first or second order differential Eqs. (1) and (2) using a numerical method. We distinguish between first and second order systems because there are certain drawbacks if one converts a naturally second order system into a first order system. To establish a continuous, differentiable motion near a given approximate solution, least square approximation using the discrete version of the Chebyshev polynomials can be invoked to obtain the solution from the the already discrete solution (Abramowitz and Stegun, 1972; Junkins, 1978). We first consider the least square approximation process. There are n data points denoted as

$$x_1 = g(t_1), \quad x_2 = g(t_2), \quad \ldots, \quad x_n = g(t_n)$$

where t_i are the values of the equally spaced independent variable ($h_t = (t_{i+1} - t_i) = \text{constant}$).

A linear transformation of independent variables should be made to use discrete orthogonality with weight function $w(t) = 1$,

$$\bar{t}(t) = \frac{t - t_1}{h_t}$$

where h_t is the constant increment of t,

$$x = g(t) = G(\bar{t}). \tag{3}$$

From n data points, the function G can be established as a linear combination of m basis functions that form the discrete version of the Chebyshev polynomials as follows:

$$G(\bar{t}) \equiv \sum_{i=1}^{m} a_i T_i(\bar{t})$$

where $m \leq n$ and $T_i(\bar{t})$ is the ith Chebyshev polynomial.

The Chebyshev polynomials are defined as follows: If $u_m = m (m = 0, 1, 2, \ldots, N)$ and $w(u) = 1$, then

$$T_n(u) = \sum_{m=0}^{n} (-1)^m \binom{n}{m} \binom{n+m}{m} \frac{u!(N-m)!}{(u-m)!N!}.$$

With the recurrence relations:

$$T_0(u) = 1$$

$$T_1(u) = 1 - \frac{2u}{N}$$

$$(n+1)(N-n)T_{n+1}(u) = (2n+1)(N-2u)T_n(u) - n(N+n+1)T_{n-1}(u).$$

Note that the recurrence relations make it easy to evaluate an expansion in Chebyshev polynomials, and a similar recurrence makes it easy to evaluate the derivative of the expansion.

Using discrete orthogonality of the Chebyshev polynomials, the typical coefficient a_j can be obtained as follows:

$$a_j = \frac{\sum_{i=1}^{n} x_i T_j(\bar{t}_i)}{\sum_{i=1}^{n} T_j(\bar{t}_i) T_j(\bar{t}_i)}$$

where $1 \leq j \leq m$.

We can find $g(t)$ from $G(\bar{t})$ because $g(t) = G(\bar{t}(t))$. Using the least square approximation, we can find the continuous, differentiable, analytical solution $x(t)$ of Eq. (3) that interpolates the n discrete numerical solutions obtained from Eqs. (1) and (2). Now this analytical expression $x(t)$ does not satisfy exactly the Eqs. (1) and (2). However, substituting $x(t)$, $\dot{x}(t)$ into Eq. (1) allows us to determine an analytical function for the perturbation term $e_1(t)$ that appears in the following differential equation:

$$\dot{x}(t) = f_1(x(t), t) + e_1(t) \equiv F_1(x, t). \quad (4)$$

Alternatively, if the system is second order, then substituting $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ into Eq. (2) allows us to determine the perturbation term $e_2(t)$ that appears in the following differential equation:

$$\ddot{x}(t) = f_2(x(t), \dot{x}(t), t) + e_2(t) \equiv F_2(x, \dot{x}, t). \quad (5)$$

Note that because $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ are available functions, $F_1(x, t)$, $F_2(x, \dot{x}, t)$ are also available functions that satisfy Eqs. (4) and (5) exactly, and $x(t)$ is a neighbor of the original numerical solution $\{x_1, x_2, \ldots, x_n\}$. By construction, the functions $e_1(t) = \dot{x}(t) - f_1(x(t), t)$ and $e_2(t) = \ddot{x}(t) - f_2(x(t), \dot{x}(t), t)$ are known analytically and therefore these small forcing functions can be computed exactly at all t. These functions are programmed and Eqs. (4) and (5) can be solved by numerical methods and the results can be compared to the exact $x(t)$, $\dot{x}(t)$. The above mathematical procedure can be performed in an automated fashion using computer symbol manipulation. The symbol manipulation can also automate the generation of C or FORTRAN Code to compute function e, (t) and/or $e_2(t)$.

Now Eq. (4) is a benchmark problem neighboring Eq. (1) and we have arranged that $x(t)$, $\dot{x}(t)$ satisfy Eq. (4) exactly; and Eq. (5) becomes the benchmark problem neighboring Eq. (2) and we have arranged that $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ satisfies Eq. (5) exactly. We obviously want the perturbation function $e(t)$ to be as small as possible, that is, the benchmark problem is not only a near neighbor of the original discrete solution, but it also very nearly satisfies the same differential equations. The previously discussed least square approximation method typically gives the poorest approximation near the ends of the interval. This may result in a relatively large $e(t)$ near the initial and final times. To avoid this problem we can integrate Eqs. (1) and (2) over the enlarged interval $t_{0-} \leq t \leq t_{f+}$ (where $t_{0-} < t_0$, $t_{f+} > t_f$) and use these numerical results as generators for analytical solutions over the original interval ($t_0 \leq t \leq t_f$). Experience indicates that a 20% "enlargement" $\{(t_{f+} - t_{0-}) \geq 1.2(t_f - t_0)\}$ is almost always sufficient to support good interpolation over the original interval ($t_0 \leq t \leq t_f$). If the measure of $e(t)$ is judged too large then we increase the number of Chebyshev polynomials m to reduce $e(t)$ over the whole interval, or "start over" by attempting to find a better approximate numerical solution to initiate the process. Figures 1 and 2 provide logical flow charts showing construction of a benchmark problem and an associated convergence study for second order systems.

ILLUSTRATIVE EXAMPLES

Now we demonstrate the previous ideas using five initial value problems for ordinary differential equations. First we show the utility of the computer codes (Lee and Junkins, 1993) for two simple nonstiff problems. Then, two celestial me-

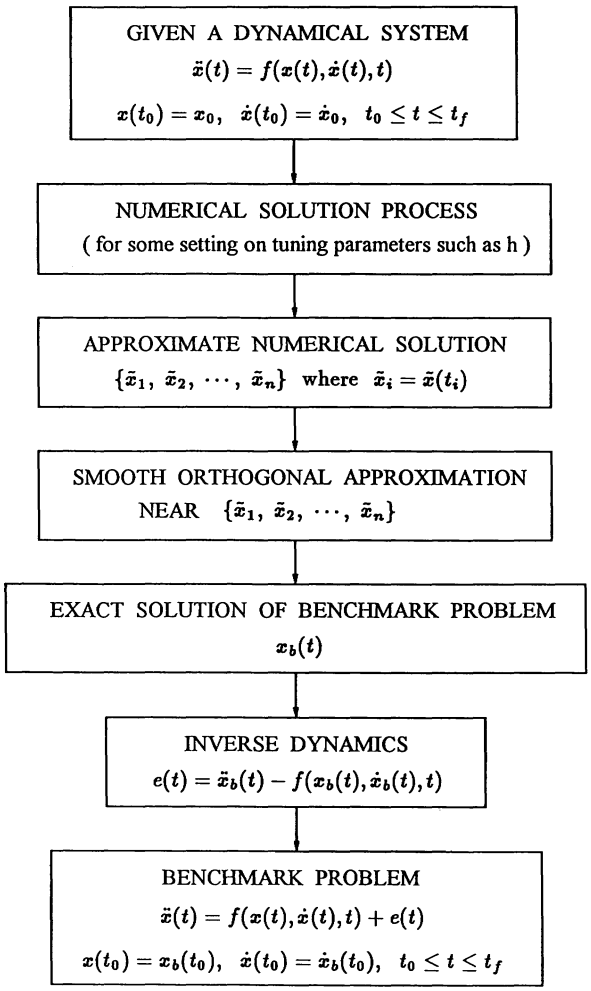

FIGURE 1 Flow chart for construction of a benchmark problem.

chanics problems are introduced to illustrate the utility of this methodology when we use the IMSL (1989) subroutines DIVPRK and DIVPBS. Finally, we consider a stiff problem in the fifth example.

First Order Systems

We consider the following pair of nonlinear differential equations.

$$\begin{aligned} \dot{x} &= 2x_1 - 2x_1x_2 \\ \dot{x} &= -x_2 + x_1x_2 \end{aligned} \tag{6}$$

where $x_1(0) = 1$ and $x_2(0) = 3$, and we seek the solution over the interval $0 \le t \le 10$.

First, we solve Eq. (6) using the Runge–Kutta fourth order method to evaluate the candidate discrete approximate solution. Here we use 121 data points over the 20% enlarged time interval $-1 \le t \le 11$. Second, we establish a continuous, differentiable, analytical expression for interpolating $x_1(t)$ and $x_2(t)$ from the discrete approximate solution. We use 51 Chebyshev polynomials for fitting. Finally we substitute $x_1(t)$, $x_2(t)$, $\dot{x}_1(t)$, $\dot{x}(t)$ into Eq. (6) and determine functions for $e_1(t)$ and $e_2(t)$ that satisfy the following equations exactly

$$\begin{aligned} \dot{x}_1 &= 2x_1 - 2x_1x_2 + e_1 \\ \dot{x}_2 &= -x_2 + x_1x_2 + e_2. \end{aligned} \tag{7}$$

Now, Eq. (7) provides a benchmark problem for Eq. (6), and $x_1(t)$, $x_2(t)$ are the solutions that satisfy Eq. (7) exactly. Upon solving Eq. (7) numerically with various values chosen for h, we establish the relationship between step size and global error. When we use the pointwise error in the root mean square sense, Fig. 3 shows the relationship in log/log scale. The critical value h is about 0.0005 and if h decreased below 0.0005, then the results begin to deteriorate. The rate of convergence is 4 in this problem and this coincides with the fact that an rth order method should have a global error of $O(h^r)$ in the absence of arithmetic errors (Gear, 1971). Figure 4 shows the perturbation terms over the time interval. For the benchmark problem, the numerical results are very reliable when we use 0.0005 as h because the error measures are about 10^{-13} while the solutions for $x_1(t)$, $x_2(t)$ vary from 10^{-2} to 10^0 order. Now we turn our attention to the original problem. Figure 5 shows the relationship between step size and error at $t = 10$ on a log/log scale for the original problem. Because we do not know the true solution, we could follow the common way of assessing the accuracy of a family of approximate solutions using the IMSL (1989) subroutines DIVPRK and DIVPBS. Comparing Figs. 3 and 5, we notice that the shape is roughly similar but, in Fig. 5, the critical value h is 0.0002 instead of 0.0005. The reason for this minor discrepancy is the relatively large perturbation terms in Fig. 4. If we decrease the perturbation terms $e_1(t)$ and $e_2(t)$ by finding a higher order, more accurate interpolation and thereby make the benchmark problem closer to the original Eq. (6), then we can reduce this discrepancy.

Second Order Systems

We consider the following nonlinear, nonautonomous second order differential equation.

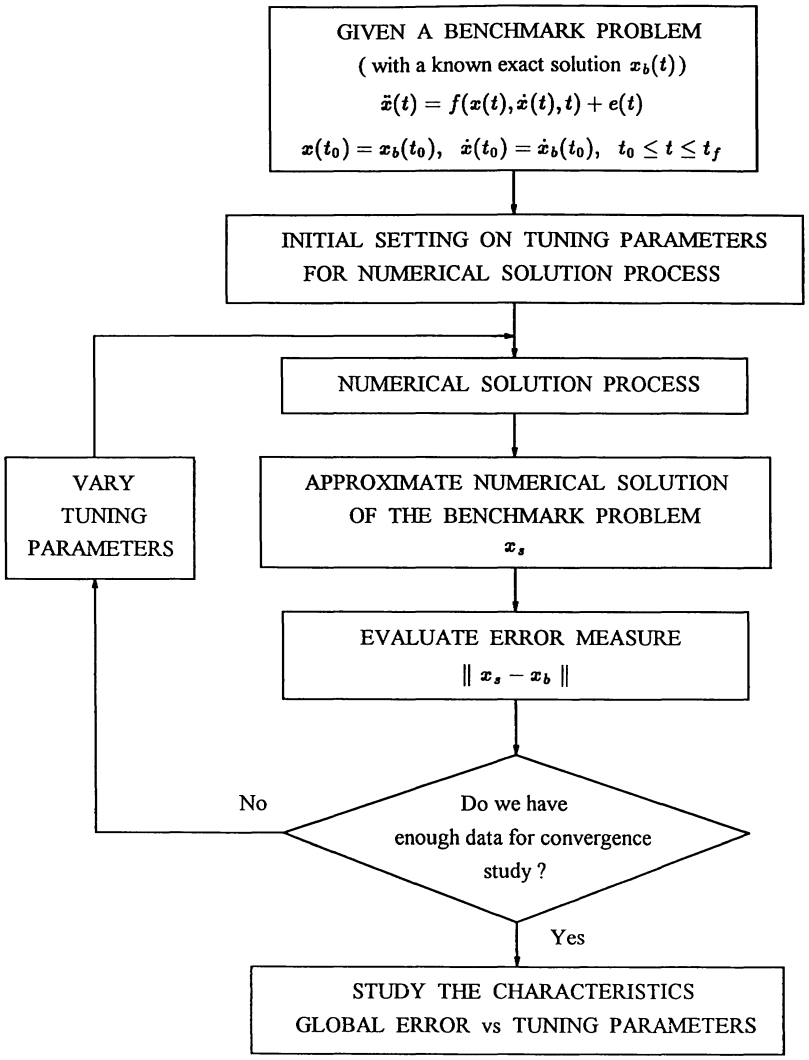

FIGURE 2 Flow chart for convergence study.

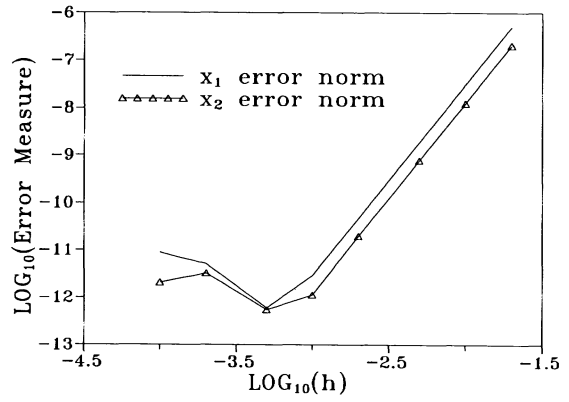

FIGURE 3 Global error vs. step size for the benchmark problem.

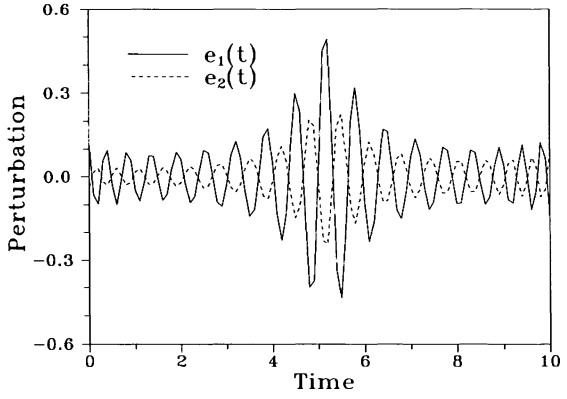

FIGURE 4 Perturbation terms of example 1.

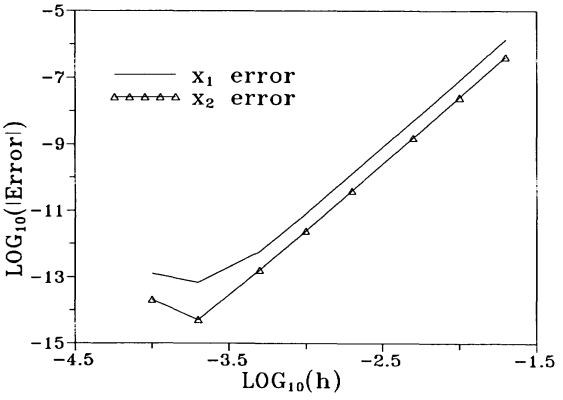

FIGURE 5 Error (at $t = 10$) vs. step size for the original problem.

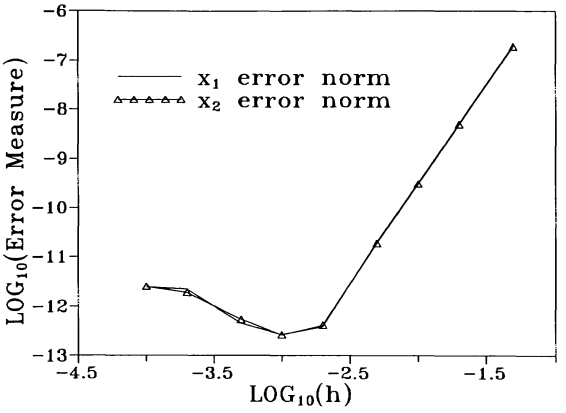

FIGURE 6 Global error vs. step size for the benchmark problem.

$$\ddot{x} = -x - 0.1(1 + x^2)\dot{x} + 0.1x^3 + \sin 3t \quad (8)$$

where $x(0) = 1$ and $\dot{x}(0) = 0$, and we seek the solution over the interval $0 \leq t \leq 10$. We convert Eq. (8) to a first order system as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - 0.1(1 + x_1^2)x_2 + 0.1x_1^3 + \sin 3t \end{aligned} \quad (9)$$

where $x_1(0) = 1$ and $x_2(0) = 0$.

We solve Eq. (9) using the Runge–Kutta fourth order method to evaluate the candidate discrete approximate solution. Here we construct the interpolated solution using 121 data points over the 20% enlarged time interval $-1 \leq t \leq 11$. An analytical expression for $x_1(t)$ is obtained from the discrete approximate solution. In this problem, a degree 30 Chebyshev polynomial is established by the least square approximation. Substituting $x_1(t)$, $\dot{x}_1(t)$, $\ddot{x}_1(t)$, into Eq. (8) we calculate the function $e(t)$ that satisfies the following equation exactly.

$$\ddot{x} = -x - 0.1(1 + x^2)\dot{x} + 0.1x^3 + \sin 3t + e. \quad (10)$$

To use the Runge–Kutta method, Eq. (10) can be converted to a first order system as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - 0.1(1 + x_1^2)x_2 + 0.1x_1^3 + \sin 3t + e. \end{aligned} \quad (11)$$

Now, Eq. (10) becomes a benchmark problem for Eq. (8), and $x(t)$ is an algebraic function that satisfies Eq. (10) exactly. When we use the pointwise error in the root mean square sense, Fig. 6 shows the relationship between global error and step size. The rate of convergence is 4 as expected. Figure 7 shows the perturbation term over the time interval. The critical value for step size is about 0.001. Now we consider the original problem. The relationship between step size and error at $t = 10$ is shown in Fig. 8 when we follow the common way assessing the true solution using the IMSL (1989) subroutines DIVPRK and DIVPBS. Comparing Figs. 6 and 8, we observe that the critical value h and the accuracy are almost the same.

We change the initial conditions slightly and the nonautonomous term in the differential equation as follows:

$$\ddot{x} = -x - 0.1(1 + x^2)\dot{x} + 0.1x^3 + 1.2 \sin 3t \quad (12)$$

where $x(0) = 1.2$ and $\dot{x}(0) = 0.2$ over the interval $0 \leq t \leq 10$.

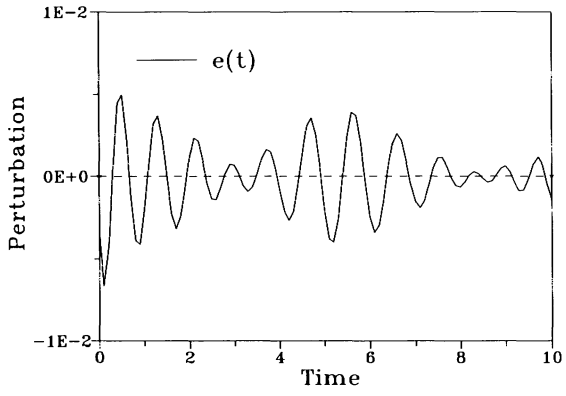

FIGURE 7 Perturbation term of example 2.

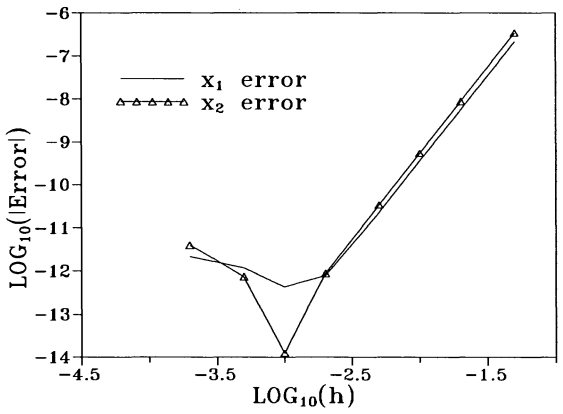

FIGURE 8 Error (at $t = 10$) vs. step size for the original problem.

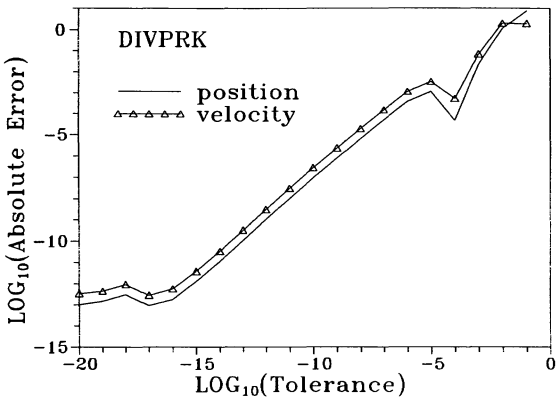

FIGURE 10 Absolute error vs. tolerance for the benchmark problem (DIVPRK).

After using the same procedure, we obtain the global error/step size relationship shown in Fig. 9. We notice that Figs. 6 and 9 are almost the same. In other words, the critical value for h and the accuracy are almost identical even though there are 20% perturbations in the initial condition and the forcing term in the differential equation, in this case.

Two Body Problem

We consider the simple two body problem. The exact solution is periodic with period 2π and the solution traces out an ellipse with eccentricity 0.6.

$$\begin{aligned} \ddot{x} &= -x/r^3, \quad x(0) = 0.4, \quad \dot{x}(0) = 0 \\ \ddot{y} &= -y/r^3, \quad y(0) = 0, \quad \dot{y}(0) = 2 \end{aligned} \tag{13}$$

where $r = (x^2 + y^2)^{1/2}$.

These equations can be solved exactly (Battin, 1987); the analytical solution is not included here because of space limitations. We reformulate Eq. (13) as a first order system as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1/(x_1^2 + x_3^2)^{3/2} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -x_3/(x_1^2 + x_3^2)^{3/2} \end{aligned} \tag{14}$$

where $x_1(0) = 0.4$, $x_2(0) = 0$, $x_3(0) = 0$, $x_4(0) = 2$.

We solve Eq. (14) using DIVPRK to evaluate the candidate discrete approximate solution. Here we use 121 data points over the 20% enlarged time interval and a degree 50 Chebyshev polynomial approximation is used for the least square fitting of $x_1(t)$ and $x_3(t)$. After constructing the benchmark problem, we do an absolute error test on $(0.\ 2\pi)$. Figures 10 and 11 show the

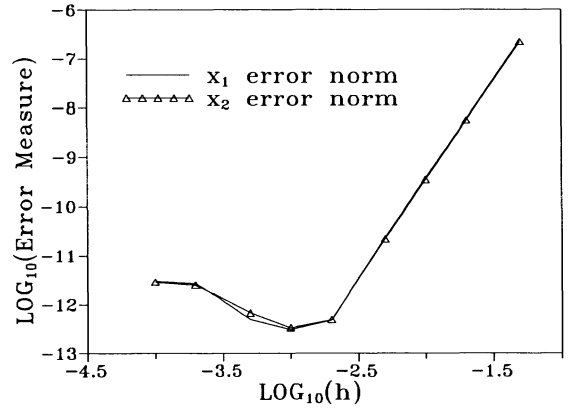

FIGURE 9 Global error vs. step size for the benchmark problem of 20% perturbations.

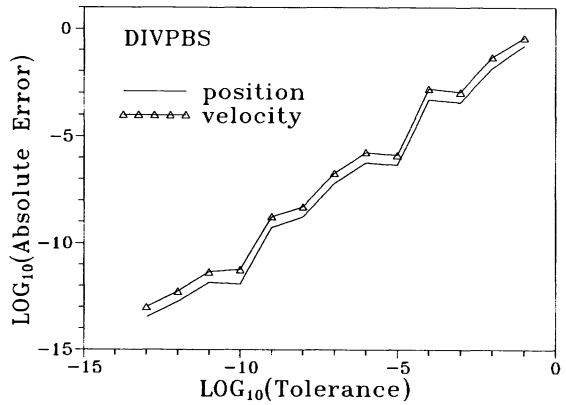

FIGURE 11 Absolute error vs. tolerance for the benchmark problem (DIVPBS).

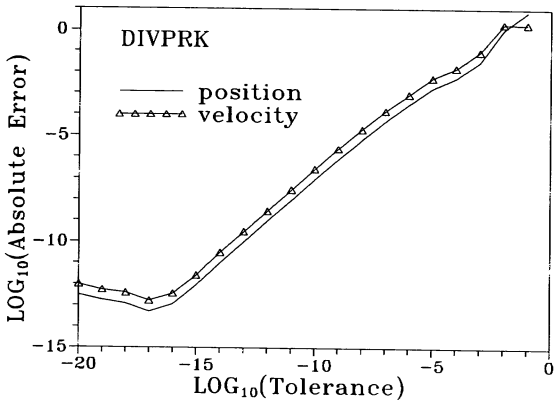

FIGURE 12 Absolute error vs. tolerance for the two body problem (DIVPRK).

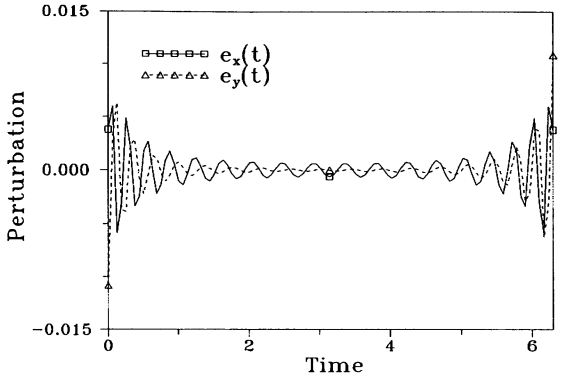

FIGURE 14 Perturbation terms of the two body problem.

relationship between absolute error and tolerance in log/log scale when we use DIVPRK and DIVPBS for the benchmark problem. Figures 12 and 13 show the relationship between absolute error and tolerance in log/log scale when we use DIVPRK and DIVPBS for the original two body problem. We notice that Figs. 10 and 11 are almost identical to Figs. 12 and 13, respectively. The perturbation terms are shown in Fig. 14. We plot the relationship between the number of function calls and the absolute error in Fig. 15. Thus the benchmark problem (constructed by the method of this study) essentially gives results that are identical to those obtained by using the exact solution of the original problem.

Euler Equations of Motion

We consider the Euler equation of motion for a rigid body without external forces,

$$\begin{aligned} \dot{x}_1 &= x_2 x_3 \\ \dot{x}_2 &= -0.51 x_3 x_1 \\ \dot{x}_3 &= -x_1 x_2 \end{aligned} \tag{15}$$

where $x_1(0) = 0$, $x_2(0) = 1$, $x_3(0) = 1$.

The classical exact solutions of Eq. (15) are the Jacobian elliptic functions (Abramowitz and Stegun, 1972) as follows:

$$x_1 = sn(t \mid 0.51), \quad x_2 = dn(t \mid 0.51),$$
$$x_3 = cn(t \mid 0.51).$$

They are periodic with a quarter period K where $K = 1.86264\ 08023\ 32738\ 55203\ \cdots$ in this case.

We solve Eq. (15) using DIVPRK to evaluate the candidate discrete approximate solution. To

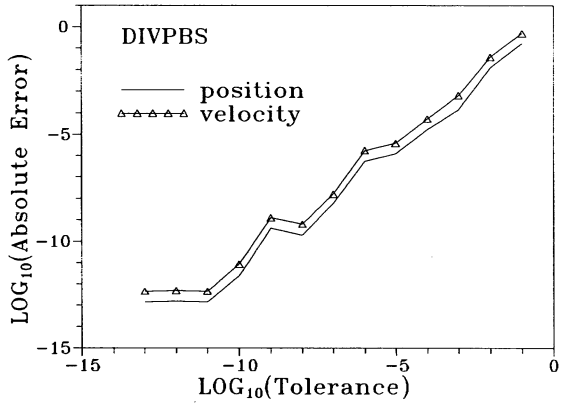

FIGURE 13 Absolute error vs. tolerance for the two body problem (DIVPBS).

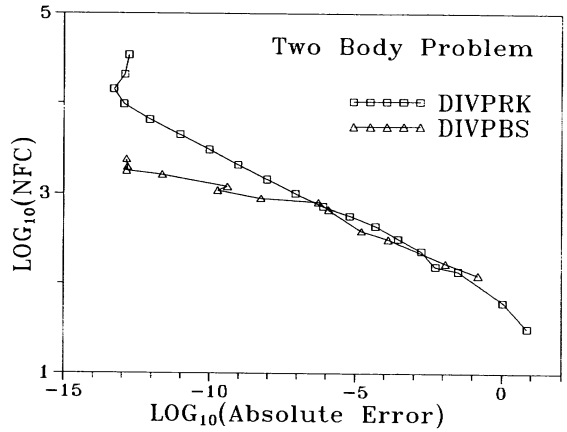

FIGURE 15 Number of function calls vs. absolute error.

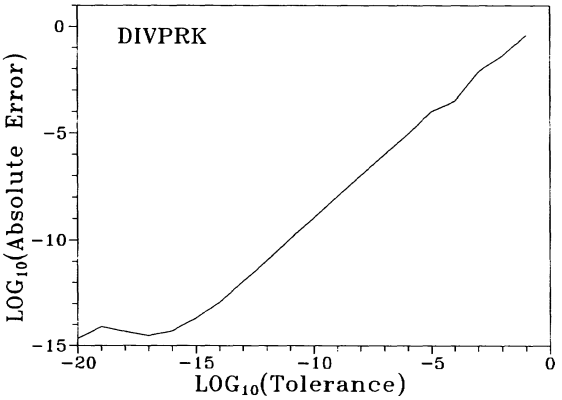

FIGURE 16 Absolute error vs. tolerance for the benchmark problem (DIVPRK).

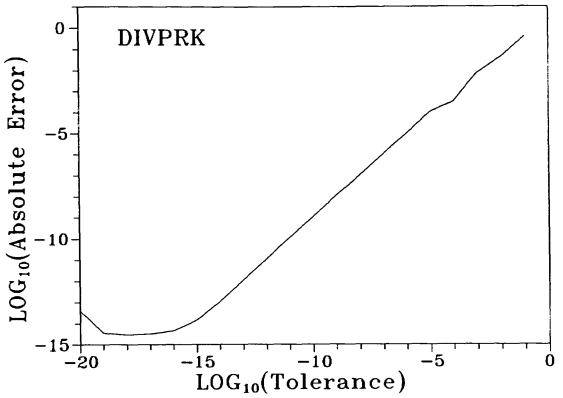

FIGURE 18 Absolute error vs. tolerance for the Euler equations (DIVPRK).

establish a benchmark using our method, we use 121 data points over the 20% enlarged time interval and determine a degree 50 Chebyshev least square polynomial approximation of $x_1(t)$, $x_2(t)$, and $x_3(t)$. After constructing the benchmark problem, we do an absolute error test on (0, 4 K). Figures 16 and 17 show the relationship between absolute error and tolerance in log/log scale when we use DIVPRK and DIVPBS for the benchmark problem. Figures 18 and 19 show the relationship between absolute error and tolerance in log/log scale when we use DIVPRK and DIVPBS to solve Eq. (15) and compare to the classical Jacobian elliptic function solution. We notice that Figs. 16 and 17 are almost identical to Figs. 18 and 19, respectively. The perturbation terms are shown in Fig. 20. We plot the relationship between the number of function calls and the absolute error in Fig. 21. Thus, again, this example indicates that our neighboring benchmark problem leads to essentially identical convergence properties to using the exact solution of the original problem.

A Stiff Problem

We consider the following problem (Shampine and Gordon, 1975) that represents a typical stiff problem.

$$\begin{aligned} \dot{x}_1 &= -29998x_1 - 39996x_2 \\ \dot{x}_2 &= 14998.5x_1 + 19997x_2 \end{aligned} \tag{16}$$

where $x_1(0) = 1$, $x_2(0) = 1$.

The exact solutions of Eq. (16) are as follows:

$$\begin{aligned} x_1(t) &= 7\exp(-10^4 t) - 6\exp(-t) \\ x_2(t) &= -3.5\exp(-10^4 t) + 4.5\exp(-t). \end{aligned} \tag{17}$$

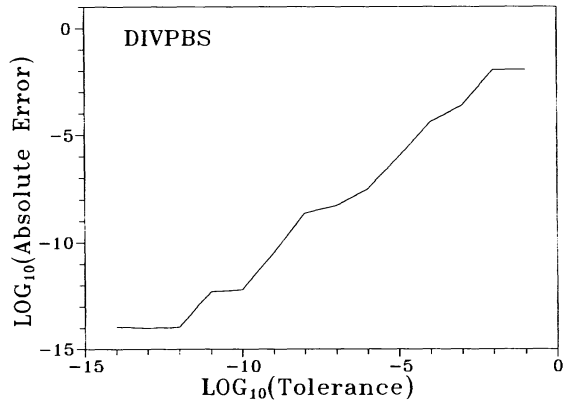

FIGURE 17 Absolute error vs. tolerance for the benchmark problem (DIVPBS).

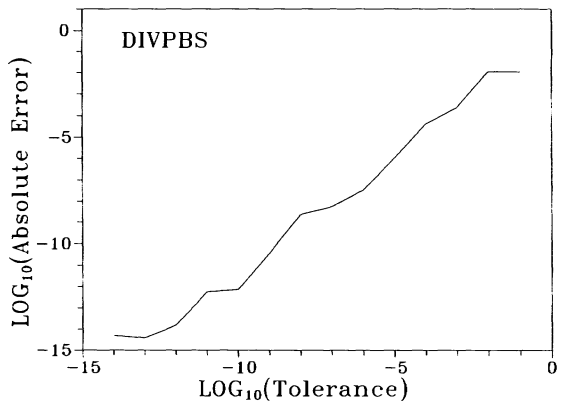

FIGURE 19 Absolute error vs. tolerance for the Euler equations (DIVPBS).

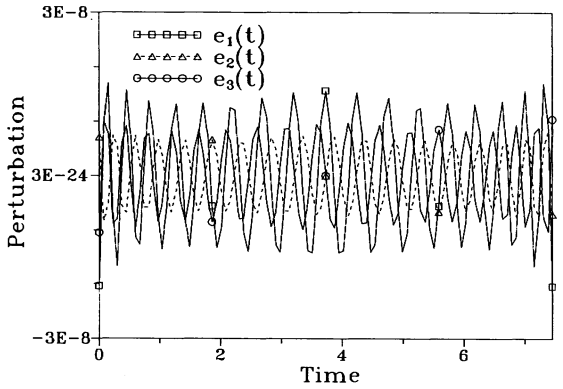

FIGURE 20 Perturbation terms of the Euler equations.

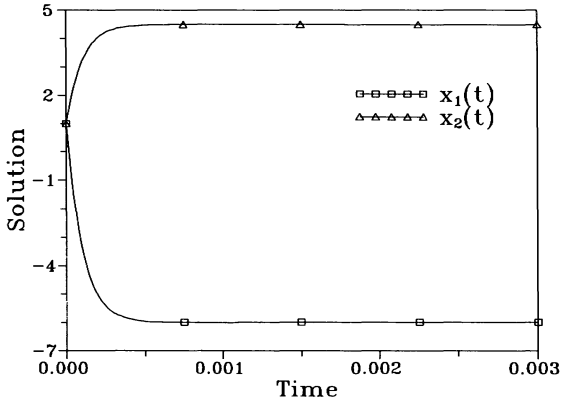

FIGURE 22 Solution of example 5 for the rapid change region.

The eigenvalues of the coefficient matrix are -1 and -10^4. Figures 22 and 23 show the solutions over two different intervals, a region of very rapid change followed by gradual asymptotic behavior. It is almost impossible to obtain a satisfactory orthogonal function benchmark problem that covers both regions with a reasonable number of terms. We conclude that the proposed methodology is not adequate for such stiff problems unless piecewise approximation methods, for example, the type introduced by Junkins et al. (1973) are used. Stiff problems are relatively expensive to solve and the expense depends strongly on the tolerance (Gear, 1971; Shampine and Gordon, 1975; Shampine and Gear, 1979). Enright et al. (1975) provide a good collection of stiff test problems.

SUMMARY AND CONCLUSION

The present article introduces an inverse method for constructing exact benchmark problems for initial value problems. This methodology gives valuable information about the optimal tuning parameters and the accuracy of the numerical solution for a class of ordinary differential equation problems and for a given solution code. Numerical examples indicate that a rigorous error analysis is usually obtained not merely for one nominal solution, but for a substantial neighborhood of the nominal solution. If one wants to use the classical Runge–Kutta method with a fixed step size, then the codes (Lee and Junkins, 1993) provide directly useful information about the optimal step size h and the associated accuracy.

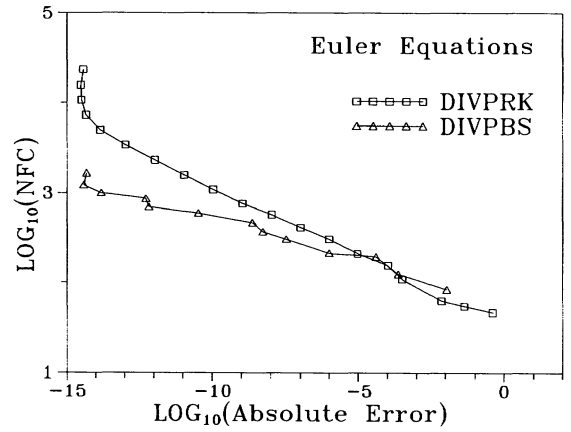

FIGURE 21 Number of function calls vs. absolute error.

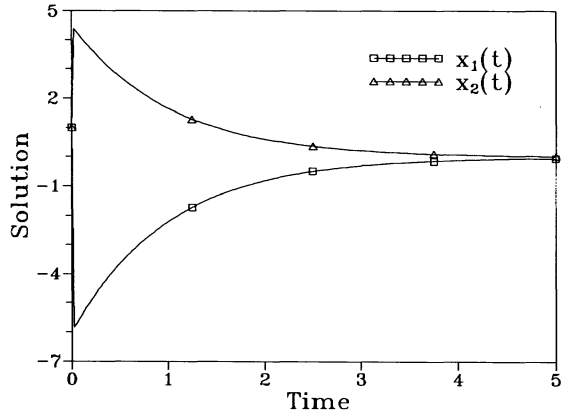

FIGURE 23 Solution of example 5 for the gradual change region.

More sophisticated users who are familiar with adaptive and robust codes can also construct similar benchmark problems; however, the Chebyshev approximation method may have to be replaced or modified to obtain a method not restricted to uniformly spaced data. For stiff systems, special purpose approximations may be required in lieu of the global Chebyshev approximations. The analytical expressions for the benchmark problem and its solution can be established using computer symbol manipulation [e.g., MACSYMA (1988), Mathematica, MAPLE, etc.]. Then the user investigates the global error/parameter relationship and compares various codes with special case absolute standards. In examples 3 and 4, we show the utility of this methodology using the IMSL (1989) subroutines DIVPRK and DIVPBS as solvers. And we investigate the absolute error/tolerance relationship and compare DIVPRK and DIVPBS. We have developed some basic methodologies, but there remains a need for additional numerical experiments to further evaluate the practical utility of this approach.

REFERENCES

Abramowitz, M., and Stegun, I. A., 1972, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards, Applied Mathematics Series 55, U.S. Department of Commerce.

Battin, R. H., 1987, *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, New York, New York.

Enright, W. H., 1989, "Analysis of Error Control Strategies for Continuous Runge–Kutta Methods," *SIAM Journal of Numerical Analysis*, Vol. 26, pp. 588–599.

Enright, W. H., Hull, T. E., and Lindberg, B., 1975, "Comparing Numerical Methods for Stiff Systems of ODEs," *BIT*, Vol. 15, pp. 10–48.

Gear, C. W., 1971, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice–Hall, Englewood Cliffs, NJ.

Hairer, E., Norsett, S. P., and Wanner, G., 1987, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer–Verlag, Berlin, pp. 236–241.

Hull, T. E., Enright, W. H., Fellen, B. M., and Sedgwick, A. E., 1972, "Comparing Numerical Methods for Ordinary Differential Equations," *SIAM Journal of Numerical Analysis*, Vol. 9, pp. 603–637.

IMSL Math/Library User's Manual Version 1.1, IMSL Inc., 1989.

Junkins, J. L., 1978, *An Introduction to Optimal Estimation of Dynamical Systems*, Sijhoff & Noordhoff, Alphen aan den Rijn, The Netherlands.

Junkins, J. L., Miller, G. W., and Jancaitis, J. R., 1973, "A Weighting Function Approach to Modeling of Irregular Surfaces," *Journal of Geophysical Research*, Vol. 78, pp. 1794–1803.

Krogh, F. T., 1973, "On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations," *Journal of the Association for Computing Machinery*, Vol. 20, pp. 545–562.

Lee, S., and Junkins, J. L., 1993, *Construction of Benchmark Problems for Solution of Ordinary Differential Equations*, Dept. of Aerospace Engineering, Texas A&M Univ., Technical Report, AERO 93-0801, College Station, TX.

MACSYMA Reference Manual Version 13, Symbolics Inc., 1988.

Shampine, L. F., 1974, "Limiting Precision in Differential Equation Solvers," *Mathematical Computation*, Vol. 28, pp. 141–144.

Shampine, L. F., 1978, "Limiting Precision in Differential Equation Solvers, II: Sources of Trouble and Starting a Code," *Mathematical Computation*, Vol. 32, pp. 1115–1122.

Shampine, L. F., 1980, "What Everyone Solving Differential Equations Numerically Should Know," in I. Gladwell and D. K. Sayers, *Computational Techniques for Ordinary Differential Equations*, Academic Press, London, pp. 1–18.

Shampine, L. F., 1987, "Tolerance Proportionality in ODE Codes," in A. Bellen, C. W. Gear, and E. Russo, *Numerical Methods for Ordinary Differential Equations*, Proceedings, L'Aquila, Springer–Verlag, New York, pp. 118–135.

Shampine, L. F., and Gear, C. W., 1979, "A User's View of Solving Stiff Ordinary Differential Equations," *SIAM Review*, Vol. 21, pp. 1–17.

Shampine, L. F., and Gordon, M. K., 1975, *Computer Solution of Ordinary Differential Equations*, W. H. Freeman, San Francisco.