Error estimators for stiff differential equations *

Lawrence F. SHAMPINE and Lorraine S. BACA
Numerical Mathematics Division 1642, Sandia National Laboratories, Albuquerque, New Mexico 87185

Received 7 November 1983

Abstract: When solving ordinary differential equations numerically, the local error is estimated at each step. In the classical situation of 'small' step sizes, it is clear what is required of the error estimators. Stiff problems are solved with 'large' step sizes. The quality of error estimators is studied in this situation, and it is shown how to modify unsatisfactory estimators so as to improve them greatly. Several formulas from the literature are treated as examples.

Keywords: Stiff, local error, error estimator

1. Introduction

Discrete variable methods for the solution of
\[ y' = f(y), \quad a \leq x \leq b, \quad y(a) \text{ given,} \]
produce approximate solutions \( y_n \approx y(x_n) \) on a mesh
\[ a = x_0 < x_1 < \cdots, \]
by stepping successively from \( a = x_0 \) to \( b \). The step size \( h_{n+1} = x_{n+1} - x_n \). The classical theory of convergence and stability is developed in the limit that the maximum step size tends to zero. This means that in practice one supposes that the step size is 'small'.

In the case of stiff differential equations, one must work with step sizes which are 'large' compared to certain characteristics of the problem. This makes the analysis of, say, stability difficult. The (now classical) linear stability theory considers a very restricted class of problems which have a form sufficiently simple that the analysis of stability is feasible for 'large' step sizes \( h \). This class can be viewed as arising from the general problem by linearization. The idea is that it is necessary that a method behave well on this restricted class, and it is hoped that the behavior for more general problems will be similar.

Modern codes for the initial value problem estimate the error made at each step. This estimate is used to see if the approximation \( y_{n+1} \) meets the accuracy requirement of the user of the code and to adjust \( h_{n+1} \) subsequently so as to obtain the desired accuracy efficiently. In the limit
$h_{n+1} \to 0$, it is clear enough what is required of the estimator. What is not at all clear is how to assess the quality of the error estimate when $h_{n+1}$ is 'large', as when solving stiff problems. A reasonable approach is to proceed as with the stability analysis by considering a standard restricted class of problems.

Sacks-Davis [4] found that certain natural error estimators for second derivative methods, which are satisfactory for small step sizes, grossly over-estimate the error made when solving stiff problems with large step sizes. He proposed modified estimators with better behavior. He did not investigate this matter in general with the consequences that one of his estimators is still unsatisfactory and that extension of the idea to other kinds of methods by Chua and Dew [2] and by Zedan [9] was not done properly.

In this paper we discuss the quality of error estimators when solving stiff problems with large step size. Gross over-estimates of the local error lead to the use of unnecessarily small step sizes, hence comparatively inefficient integrations. Some of the recent approaches to the estimation of true (global) errors are based on local error estimates and so require reasonably good estimates for large step sizes. An investigation of one such approach [6] brought this issue of quality to our attention. Here we show how to modify unsatisfactory estimators so as to improve them greatly. A number of examples of quite different kinds of methods are taken from the literature and treated in detail.

2. Quality of error estimates

For simplicity we shall speak of one-step methods for the numerical solution of the system

$$y' = f(y), \quad a \leq x \leq b, \quad y(a) \text{ given},$$

although our ideas are applicable to methods with memory. Indeed, in the last section we consider a class of such methods. Proceeding from an approximation $y_n$ of $y(x_n)$, a one-step method forms $y_{n+1} = y(x_n + h)$. The local error of this step is defined in terms of the local solution $u$ of

$$u' = f(u), \quad u(x_n) = y_n.$$

Then

$$\text{local error} = u(x_n + h) - y_{n+1}.$$

The method is said to be of order $p$ if for any sufficiently smooth $f$, there is a constant $C$ such that for all sufficiently small step sizes $h$,

$$\|u(x_n + h) - y_{n+1}\| \leq C h^{p+1}.$$ 

Along with the method there should be an estimator EST of this local error which is used to decide if the step should be accepted and also to adjust $h$ subsequently. In this context it is clear enough what we want of the estimator, namely

$$\|u(x_n + h) - y_{n+1} - \text{EST}\| = o(h^{p+1}).$$

The typical error estimator can be viewed as arising from another one-step method which produces an approximation $y^*_{n+1}$ of order greater than $p$. Then if we let

$$\text{EST} = y^*_{n+1} - y_{n+1}, \quad (2.1)$$
we have
\[ \| u(x_n + h) - y_{n+1} - (y^*_n - y^*_{n+1}) \| = \| u(x_n + h) - y^*_n \| = O(h^{p+2}). \]

The correctness of an estimator EST as \( h \to 0 \) can be tested by Taylor series expansion or may be obvious from its construction as in (2.1). The question we consider here is the quality of the error estimate when \( h \) is, in a sense, quite 'large'. This situation is typical of stiff problems, and we shall exploit a technique to study it much used for stability analyses: We restrict our attention to a class of problems of the form
\[ y' = Jy + g \]
where \( g \) is a constant vector and \( J \) is a constant matrix which can be diagonalized by a similarity transformation
\[ M^{-1}JM = \text{diag}\{ \lambda_i \}. \]
It is supposed that the eigenvalues \( \lambda_i \) of \( J \) are such that either \( |\lambda_i| \) is 'small' but non-zero, or \( \text{Re}(\lambda_i) < 0 \). As in the usual linear stability arguments, we require that the numerical method perform adequately on this class of equations and hope that its behavior on the non-linear problems of practice will be similar. Because of the loose connection to practice, it is pointless to expend any great effort to secure minor advantages on the restricted class.

Our assumptions about (2.2) lead to the representation
\[ u(x_n + h) = q + \exp(hJ)(y_n - q) \]
where \( q = -J^{-1}g \). We shall assume that the numerical method integrates the constant solution \( q \) exactly, is linear for this class of problems, and has the form
\[ y' = q + R(hJ)(y_n - q) \]
for a rational function \( R(z) \). These assumptions are typically true. Further suppose that the error estimate has the form
\[ \text{EST} = R_{e}(hJ)(y_n - q) \]
for this class of problems where \( R_{e}(z) \) is another rational function. This is certainly true if EST arises as in (2.1) from a higher order one-step method which satisfies the same assumptions as the basic method.

Now
\[ u(x_n + h) - y_{n+1} = [\exp(hJ) - R(hJ)](y_n - q), \]
\[ u(x_n + h) - y_{n+1} = M \text{diag}\{ \exp(h\lambda_i) - R(h\lambda_i) \} M^{-1}(y_n - q) \]
(2.3)
exposes the behavior of the local error. The role of the method itself is confined to how well the rational function \( R(z) \) approximates \( \exp(z) \) at the points \( h\lambda_i \). The actual effect on the local error depends on the conditioning of the differential equation (the matrices \( M, M^{-1} \)), just where the integration is (how close \( y_n \) is to the limit solution \( q \)), and the norm in which the error is to be measured.

The order condition implies that
\[ R(z) = \exp(z) + O(|z|^{p+1}) \quad \text{as } |z| \to 0, \]
so that the approximation is good for 'small' \( |h\lambda_i| \). When solving stiff problems, one is also
interested in $|h\lambda_i| \gg 1$ with $\Re(\lambda_i) < 0$. Because the exponential term tends to zero very rapidly, to get even a ‘reasonable’ approximation in this range places stringent requirements on the method. A method for which $|R(z)| \leq 1$ for $\Re(z) \leq 0$ is called A-stable. If, in addition, $R(z) \to 0$ as $|z| \to \infty$, the method is called L-stable. As examples we cite the trapezoidal rule

$$y_{n+1}^* = y_n + \frac{1}{2}h \left[f\left(y_{n+1}^*ight) + f(y_n)\right],$$

for which

$$R^*(z) = \frac{1 + z/2}{1 - z/2},$$

which is of second order and A-stable, and the backward Euler method

$$y_{n+1} = y_n + hf(y_{n+1}),$$

for which

$$R(z) = \frac{1}{1 - z},$$

which is of first order and L-stable.

The estimate of the local error can be written in a form like (2.3), namely

$$\text{EST} = M \text{diag}\{R_e(h\lambda_i)\} M^{-1}(y_n - q).$$

Clearly, for this class of problems, an appropriate measure of the quality of the error estimator for ‘large’ $h$ is that $R_e(z)$ approximates $\exp(z) - R(z)$ well for $|z|$ large with $\Re(z) < 0$. If we take the backward Euler method as our one-step method and construct an error estimator with the trapezoidal rule by (2.1), we find that for $|z| \to \infty$, $\Re(z) < 0$,

$$\exp(z) - \frac{1}{1 - z} - \frac{1}{z},$$

but

$$R_e(z) = \frac{1 + z/2}{1 - z/2} - \frac{1}{1 - z} \sim 1.$$

Although the trapezoidal rule provides a perfectly satisfactory estimator for ‘small’ $h$ ($|h\lambda_i|$ small), it grossly over-estimates the error for ‘large’ $h$ ($|h\lambda_i| \gg 1$, $\Re(\lambda_i) < 0$). This causes a code to use a step size unnecessarily small. As we have observed, it is not possible to be precise about the effects due to an estimator poor for large $h$. A meaningful way to describe the situation with this example is that the stability of the method is that of the L-stable backward Euler formula, but the estimated accuracy is like that of a method which is only A-stable.

Suppose we have an error estimator $\text{EST}$ with the correct behavior as $h \to 0$, but that its behavior is unsatisfactory for large $h$ when the method is used to solve (2.2). Because we are most interested in a gross over-estimate of the error, as with the backward Euler-trapezoidal rule combination, we shall describe only this case. Let $R(z)$, $R_e(z)$ be the rational functions describing the behavior of the method and error estimate on the class (2.2) in the way just explained. Let

$$R_i(z) = \exp(z) - R(z)$$
describe the true local error. Suppose now that
\[
\frac{R_e(z)}{R_t(z)} \sim cz^m, \quad \Re(z) < 0, |z| \to \infty,
\]
with positive integer \(m\) so that the error is grossly over-estimated for ‘large’ step sizes.

Our idea for improving the behavior of EST is quite simple. It proceeds from the fact that methods for stiff problems form and factor matrices \(W = I - hA\) at each step. Here \(A\) is an approximate Jacobian and \(d\) is a constant characteristic of the method. There may even be several such matrices. For example, \(I - hA\) would be formed to evaluate the backward Euler method, and \(I - 0.5hJ\) to evaluate the trapezoidal rule. Because it is easy to solve linear systems with matrix \(W\), it is easy to form
\[
\text{EST'} = W^{-m} \text{EST}.
\]

First we note that
\[
\text{EST'} \sim \text{EST} \quad \text{as } h \to 0,
\]
so that this modified estimate is also asymptotically correct. It is obvious that the behavior of the modified estimate is described by the rational function
\[
R'_e(z) = \frac{1}{(1 - dz)^m} R_e(z).
\]
This implies that
\[
\frac{R'_e(z)}{R_t(z)} \sim \frac{c}{(-d)^m}, \quad |z| \to \infty, \quad \Re(z) < \infty,
\]
hence that the modified estimate corrects the order at infinity.

In the case of the backward Euler method with trapezoidal rule error estimate, either matrix \(W\) could be used. The matrix \(W = I - hJ\) results in a better constant. With it we have
\[
R'_e(z) = 1.
\]

Thus by simply solving one linear system with a factored matrix,
\[
\text{EST'} = (I - hJ)^{-1} \text{EST} = (I - hJ)^{-1}(y_{n+1}^* - y_{n+1}),
\]
we obtain an asymptotically correct error estimator which correctly reflects the damping of the basic Euler formula at infinity.

Our idea for modifying the error estimator does not alter the behavior for small \(h\) while correcting the behavior for large \(h\). In a sense, the modification has a uniform effect for all \(h\). This is seen from
\[
|R'_e(z)| = |(1 - dz)^{-m} R_e(z)| = \left|\frac{1}{1 - dz}\right|^m |R_e(z)| \leq |R_e(z)| \quad \text{for } \Re(z) \leq 0.
\]
That is, in a sense, our modification never increases the original estimate. This is a rather pleasant property.
3. Some applications

We have seen that the behavior of an error estimator needs to be considered for large, as well as small, step sizes when solving stiff problems. A way of correcting gross over-estimates of the local error was described and illustrated with a simple formula pair. In this section we present several examples from the literature which show how severe the problem can be and illustrate our technique for improving the estimate.

Steinhaug and Wolfbrandt [8] have given a second order, semi-implicit, one-step method which has attracted some attention with respect to its local error estimator. For equations in autonomous form the formula is

\[ W_{k_1} = f(y_n), \quad W_{k_2} = f(y_n + \frac{3}{2}hk_1) - \frac{3}{4}hdJk_1, \]
\[ y_{n+1} = y_n + \frac{1}{4}h(k_1 + 3k_2), \]

where
\[ W = I - hdJ, \quad d = 1 - \frac{1}{6}\sqrt{2}, \]
and \( J \) is an approximation to the Jacobian matrix. If \( J = f_y(y_n) \), the method is L-stable. The true error function

\[ R_1(z) = \exp(z)(1 - dz)^2 - z(1 - 2d) - 1 \]
\[ (1 - dz)^2, \]
so that
\[ R_1(z) \sim -\frac{(1 - 2d)}{d^2z} \quad \text{for } |z| \to \infty, \Re(z) < 0. \]

We caution the reader that here, and in similar expressions which follow, \( d \) is not a free parameter; it is just notation for the constant \( 1 - \frac{1}{6}\sqrt{2} \).

Steinhaug and Wolfbrandt proposed a local error estimator which continues the computations resulting in \( y_{n+1} \) by

\[ Wk_3 = f(y_n + h(\frac{1}{2}k_1 + \frac{3}{2}k_2)), \]
\[ Wk_4 = f(y_n + h(\frac{1}{2}k_1 + \frac{5}{4}k_2 + \frac{3}{4}k_3)) + hdJ(\frac{3}{8}k_1 + 6k_2), \]
\[ \text{EST}_1 = \frac{1}{4}h(k_1 - 5k_2 + 5k_3 - k_4). \quad (3.1) \]

If the step succeeds and if \( h \) is not changed, these computations represent the bulk of the work in calculating \( y_{n+2} \). However, this suggests a disadvantage made clearer if one were to use the non-autonomous form: this error estimate requires evaluation of \( f \) outside the interval \([x_n, x_n + h]\).

Another disadvantage of (3.1) appears when we ask about the quality of the error estimate for large \( h \). A little calculation shows that the error estimate function

\[ R_e(z) = \frac{(3d^2 + 2d - \frac{1}{2})z^3 + (-3d^3 - \frac{4}{3}d^2 + \frac{1}{3}d)z^4}{8(1 - dz)^4}, \quad (3.2) \]
so that

$$R_e(z) \sim \frac{(-3d^2 - \frac{1}{2}d + \frac{1}{2})}{8d^3} \quad \text{for } |z| \to \infty, \ Re(z) < 0.$$  

That is, the error estimate tends to a finite limit at infinity whereas the true error tends to zero. This is entirely analogous to our example of the backward Euler formula with trapezoidal rule estimator.

In an effort to retain the one-step nature of the formula itself, Scraton [5] made some additional assumptions about the matrix $J$ and derived the estimator

$$\text{EST}_2 = \frac{1}{b} h \left[ f_y + 3f(y, z + \frac{1}{2}h_k) - k_1 - 3k_2 \right].$$

This uses only information available from the step itself. Unfortunately

$$Re(z) = \frac{-1}{d^2} \left( \frac{1}{d^2} \right) \quad \text{for } |z| \to \infty, \ Re(z) < 0.$$  

The disparity between the true and estimated errors is now so strong that a code based on this estimate must be comparatively very inefficient.

Chua and Dew [2], and also Zedan [9], suggested a modification of Scraton’s estimator to correct the bad behavior at infinity. Following Sacks-Davis [4], they proposed

$$\text{EST}_3 = \frac{1}{W}(\text{EST}_2),$$  

for which

$$R_e(z) = \frac{-(d - 2d^2)z^3}{3(1 - dz)^3},$$

so that

$$R_e(z) \sim \frac{-(d - 2d^2)z^3}{3d^3} \quad \text{for } |z| \to \infty, \ Re(z) < 0.$$  

The disparity between the true and estimated errors is now so strong that a code based on this estimate must be comparatively very inefficient.

Alt [1] has presented some semi-implicit one-step methods with error estimates which are extremely poor for large step sizes. He, for example, derives an A-stable semi-implicit third order
method for the basic formula. He proposes to estimate its local error by also evaluating an explicit fourth order Runge-Kutta method. He observes that the estimate is asymptotically correct as the step size $h \to 0$, and the particular fourth order formula is chosen to provide starting approximations for the evaluation of the semi-implicit formula. Because the semi-implicit formula is A-stable, $R_i(z)$ is bounded for $\text{Re}(z) \leq 0$. The explicit 4-stage, fourth order Runge-Kutta formula has $R^*(z)$ a fourth order polynomial in $z$ so that $R_e(z)$ grows like $z^4$ at infinity. Although our approach could correct such a gross over-estimate of the local error, it hardly seems worthwhile. It would be necessary to solve four linear systems at each step, so it seems likely that it would be better to make a fresh start at deriving an error estimate. The situation is still worse with the higher order pair of formulas Alt derives.

The error estimate used by Alt is so poor that it is reasonable to ask why the gross inefficiency was not immediately obvious in his computational results. We looked into this a little bit. Only one example was solved with an error control, namely

$$y' = z - y^2 - (1 + x), \quad y(0) = 1,$$
$$z' = 1 - 20[z^2 - (1 + x)^2], \quad z(0) = 1$$
on $0 \leq x \leq 2$. The eigenvalues of the Jacobian along the true solution are stated to be $-2/(1 + x)$ and $-40(1 + x)$. Considering the interval length and the eigenvalues, this is at best a moderately stiff problem. It was solved with a pure relative error tolerance of $10^{-7}$ on the local error at each step. Since the problem was solved with a third order method, this stringent tolerance reduces the stiffness significantly. We solved the problem with a fourth order, explicit Runge-Kutta Code RKF [7] on a CBM SP9000 at various tolerances. At, e.g., a pure relative error tolerance of $10^{-4}$, the maximum error at the end point $x = 2$ was $7.5 \times 10^{-7}$. This integration required only 304 function evaluations so one could scarcely call the problem stiff. At $x = 2$ the code was using a step size 0.03 which is comparable to the step size 0.04337 reported by Alt. With a tolerance $10^{-6}$, the error at the end point was at the roundoff level for the machine used, the cost was 362 evaluations and the step size was 0.03. We conclude that this particular example simply does not test the behavior of the error estimate for stiff problems. Perhaps we should emphasize that we are criticizing the quality of the error estimate, which we believe is disastrously bad, not the formula he derived for the integration itself.

4. A refinement

So far we have only considered modification of an error estimate so that it will have the correct order at infinity. One might well ask about adjusting the constant as well. This is basically easy and fairly inexpensive. If we alter EST to

$$\text{EST}' = [\alpha I + (1 - \alpha)W^{-1}] \text{EST},$$
it is still true that

$$\text{EST}' \sim \text{EST}, \quad \text{as } h \to 0.$$Supposing that EST already has the correct order at infinity,

$$\frac{R_e(z)}{R_i(z)} - \gamma \neq 0, \quad |z| \to \infty, \quad \text{Re}(z) < 0,$$
we find that
\[
\frac{R'(z)}{R(z)} = \left[ \alpha + \frac{1 - \alpha}{1 - dz} \right] \frac{R(z)}{R(z)} \sim \alpha \gamma.
\]

By simply taking \( \alpha = 1/\gamma \) we adjust the behavior of EST so that it is exact at infinity for EST'.

A certain difficulty may arise. The resulting estimate of the local error vanishes at \( z_0 = 1/d\alpha \). If \( \alpha > 0 \), i.e., \( \gamma > 0 \), this is a positive real number and leads to no difficulty. If \( \gamma < 0 \), we introduce a point in the left half complex plane where the estimated error is zero. This kind of behavior may already be present in the original estimate, e.g., according to (3.2), the Steihaug–Wolfbrandt estimate vanishes for
\[
\left( \frac{3d^2 + 2d - \gamma}{-3d^3 - \frac{7}{6}d^2 + \frac{1}{2}d} \right) < 0.
\]

It is hardly to be expected that the true error will vanish at the same point, and in fact, \( R(z_0) \neq 0 \) for the Steihaug–Wolfbrandt formula. Conversely, the method may be exact at some \( z_0 \), \( R(z_0) = 0 \) (exponential fitting), and it is not to be expected that the estimated error will be zero at the same point. We simply have to cope with such difficulties in the error estimate. On the other hand, we do not want to introduce any more defects.

Adjusting the constant at infinity requires solution of another linear system. The adjustment at infinity may worsen the approximation of \( R(z) \) by \( R(z) \) elsewhere. In view of the loose connection between our assessment of quality and the quality of the estimator for nonlinear problems, we do not believe the effort required to adjust the constant to be usually worthwhile. If \( \gamma < 0 \), adjusting the constant is still harder to justify. Nevertheless, one should keep the option in mind for cases when the constant \( \gamma \) is very large. A family of Rosenbrock formulas derived by Kaps and Rentrop [3] makes the point. Their third order formula is such that
\[
R(z) - L_3(1/d), \quad |z| \rightarrow \infty, \ Re(z) < 0
\]
and the fourth order formula of the pair has
\[
R^*(z) - L_4(1/d).
\]

Here \( L_3(x) \), \( L_4(x) \) are the Laguerre polynomials of orders three and four respectively. The local error of the third order result is estimated by comparison to the fourth order result as in (2.1). Thus
\[
R_1(z) = \exp(z) - R(z) \sim -L_3(1/d),
\]
\[
R_\ast(z) = R^*(z) - R(z) \sim L_4(1/d) - L_3(1/d).
\]

They give a family of formula pairs with \( d \) as a parameter. By choosing \( d \) appropriately, it is possible to make \( R(\infty) = 0 \) or \( R^*(\infty) = 0 \), but not both. In either case the error estimate is of the wrong order, in the one, an over-estimate and in the other, an under-estimate. If \( d \) is ‘close’ to one of these values, the error estimate has the correct order, but the constant is so large that the estimate is as bad as if the order were wrong. This would require correction. The two pairs of formulas given by Kaps and Rentrop do not have \( d \) very near these points, so the error of the estimate at infinity is not large. As it happens, one pair has \( \gamma > 0 \) and the other \( \gamma < 0 \) (in our notation, not theirs) so that one is particularly unfavorable for adjustment.
5. Second derivative methods

In [4] Sacks-Davis considers the quality of error estimates for second derivative methods when applied to stiff differential equations. He points out that a usual error estimate $E_1$ based on a comparison of predicted and corrected values leads to gross over-estimates of the local errors for large step sizes $h$. Observing from $y' = f(y)$ that

$$y'' = f_y(y)f(y)$$

and recalling that the Jacobian $f_y$ will be needed in solving stiff problems, one is led to considering second derivative formulas of the form

$$y_{n+1} = y_n + h \sum_{j=0}^{k-1} \beta_j y_{n+1-j} + h^2 \gamma_0 y_{n+1}'',$$

where $\gamma_0$ and the $\beta_j$ characterize the formula. In evaluating an implicit formula of this kind, one needs to form and factor

$$W = I - h \beta_0 f_y - h^2 \gamma_0 f_y^2.$$

Sacks-Davis suggested that one consider the error estimate $E_2 = W^{-1}E_1$ and showed that it has a better behavior for large $h$.

In our notation, Sacks-Davis proves that for the error estimator $E_1$ as $|z| \to \infty$, $\text{Re}(z) < 0$,

$$\frac{R_2(z)}{R_1(z)} \sim \begin{cases} z^3 & \text{formula of order three}, \\ z^2 & \text{formula of order four}, \\ z^2 & \text{formula of order five} \end{cases}$$

On the other hand, the modified estimate $E_2$ has

$$\frac{R_2(z)}{R_1(z)} \sim \begin{cases} \frac{14}{13} z & \text{formula of order three}, \\ \frac{102}{133} z & \text{formula of order five} \end{cases}$$

The estimators were only one part of the paper, and Sacks-Davis did not go into this particular issue too deeply. The estimator for the formula of order three is much improved, but still not of the correct order. He presented his idea in the form of an example and did not indicate how to extend it to other kinds of formulas. Indeed, as we pointed out in Section 3, Chua and Dew, and also Zedan, did not properly extend the work of Sacks-Davis to a different kind of formula. Now that we have looked into the matter a little more deeply, it is easy to see how to correct his estimator for the formula of order three. The code retains the Jacobian $f_y$ so that we can form at a reasonable cost

$$E' = W^{-2}(I + \alpha h f_y)E_1 = W^{-1}(I + \alpha h f_y)E_2$$

for a suitable constant $\alpha$. The corresponding measure of quality for the formula of order three is

$$\frac{R'_2(z)}{R_1(z)} = \frac{(1 + az)}{(1 - \beta_0 z - \gamma_0 z^2)^2} \frac{R_2(z)}{R_1(z)} - \frac{\alpha}{\gamma_0^2}.$$
Thus, taking $\alpha = \gamma_0^2$ leads to the proper behavior, even the right constant, for large $h$, and obviously $E' \sim E_1$ as $h \to 0$.

6. Remarks

Sacks-Davis [4] and Zedan [9] have presented plots of $R_e(z)/R_i(z)$ along the negative real axis. We have made similar plots for all the examples of this paper but have chosen not to present them because the asymptotic behavior for $|z| \to 0$ and $|z| \to \infty$ with $\text{Re}(z) < 0$ gives a fair picture. It is worth remark that the plots show that $|z|$ on the order of 50 is already ‘large’. This represents only modest stiffness, so that the asymptotic results are quite accurate for realistically large step sizes.

References