Numer. Math. 41, 165-175 (1983)

Numerische Mathematik

Smoothing the Extrapolated Midpoint Rule*

Lawrence F. Shampine and Lorraine S. Baca
Numerical Mathematics Division, Sandia National Laboratories, Albuquerque, NM 87185, USA

Summary. The extrapolated midpoint rule is a popular way to solve the initial value problem for a system of ordinary differential equations. As originally formulated by Gragg, the results are smoothed to remove the weak instability of the midpoint rule. It is shown that this smoothing is not necessary. A cheaper smoothing scheme is proposed. A way to exploit smoothing to increase the robustness of extrapolation codes is formulated.

Subject Classifications: AMS(MOS): 65LO5; CR: 5.17.

1. Introduction

In his dissertation [4] and later in [5], W.B. Gragg proposed that the explicit midpoint rule with special starting procedure serve as the fundamental formula for the solution by h^2-extrapolation of the initial value problem for a system of ordinary differential equations (ODEs). The idea was implemented by Bulirsch and Stoer [2] who showed the approach to be an effective one for non-stiff problems.

It seems to have been considered obvious that smoothing is necessary, or at least highly desirable, when the midpoint rule is used for extrapolation. In the extrapolation methods one integrates repeatedly from x_n to x_n+H with successively smaller (fixed) step sizes h. The results at x_n+H of the various subintegrations are combined in a linear (polynomial extrapolation) or nonlinear (rational extrapolation) way to yield a high order result at x_n+H. The midpoint rule is well known to have no region of absolute stability. These successive integrations with a weakly stable formula appear to be dangerous. Milne and Reynolds [7] successfully smoothed the results of integrations with Milne's formula to eliminate a similar instability. In view of this, Gragg

* This work performed at Sandia National Laboratories supported by the US Department of Energy under contract number DE-AC04-76DP00789

proposed a smoothing scheme for his extrapolation procedure and justified it with the asymptotic expansion he developed for the error.

For nearly 20 years Gragg's approach to solving ODEs has been used without questioning if smoothing is really necessary. In a preliminary study [11], we showed it is not. Here we report our findings along with important new insight. First we shall observe that the analogy with the work of Milne and Reynolds is false. Gragg's scheme is intended to estimate and then eliminate a weakly stable component of the error expansion. It is easiest to understand what is happening if one considers an "ideal" smoothing which removes exactly this term. The remarkable result then is that polynomial extrapolation gives *precisely* the same numerical results with and without ideal smoothing. There is a rather tricky point about the effect of smoothing on the algorithms for accepting a step which we shall explain.

Recently Bader and Deuflhard [1] have been studying an extrapolated semi-implicit midpoint rule for the solution of stiff problems. The method generalizes the explicit midpoint rule in a natural way. Bader invented a smoothing scheme for the semi-implicit midpoint rule which resembles Gragg's scheme. We shall point out that in contrast to the situation for non-stiff problems, Bader's scheme is quite important for stiff problems. The schemes of Gragg and Bader suggest a family of schemes for non-stiff problems which we explore. Shampine invented a cheap smoothing scheme as an alternative to Gragg's which will be explained.

We propose a way to exploit smoothing which we believe will significantly increase the robustness of extrapolation codes.

Finally we consider the merits of the various smoothing schemes as compared to each other and to not smoothing at all.

2. Preliminaries

First it will be useful to explain why one might not need to smooth the midpoint rule as it is used to solve non-stiff ODEs. There are two distinct ways one might use the idea of extrapolation - local and global. In global extrapolation one begins an integration with step size h at the initial point of the interval of integration a and advances all the way to the final point b. One then reduces h and integrates again from a to b. (In practice the integrations might be done simultaneously.) Extrapolation is done at mesh points of interest, common to both integrations, so as to obtain accurate results there. A virtue of this approach is that one obtains global, or true, error estimates. For this reason we have been studying the possibility of writing a code of this kind. This approach *is* analogous to that of Milne and Reynolds. Smoothing must be done occasionally in each integration to remove the weakly stable component of the error, else meaningless results or overflow may result.

As implemented by Bulirsch and Stoer first and in all succeeding codes for non-stiff problems, extrapolation is done locally. The integrations with the midpoint rule are done from a solution at a mesh point x_n to the next mesh point $x_{n+1}=x_n+H$. The results of the various subintegrations are smoothed at

x_n+H and then extrapolated to yield an accurate result there. This accurate result is used to proceed to the next step. The resulting procedure is therefore a one-step method. The smoothing does not affect the subintegrations with the midpoint rule at all. This is entirely different from the situation of Milne and Reynolds. Smoothing can at most affect the accuracy of the extrapolated result at x_{n+1}.

The notation for the extrapolation schemes is a bit clumsy. We shall follow that of the text [14, p. 453ff]. Let $\eta(x;h)$ denote an approximation at the point $x=x_0+nh$ to the solution $y(x)$ of

$$y'=f(x,y), \qquad y(x_0)=y_0. \tag{2.1}$$

The midpoint rule with special starting procedure is

$$\begin{aligned}\eta(x_0;h)&=y_0,\\ \eta(x_0+h;h)&=y_0+hf(x_0,\eta(x_0;h)),\\ \eta(x+h;h)&=\eta(x-h;h)+2hf(x,\eta(x;h)).\end{aligned}$$

If f is sufficiently smooth, $\eta(x;h)$ has an asymptotic expansion of the form

$$\eta(x;h)=y(x)+\sum_{k=1}^{N} h^{2k}[u_k(x)+(-1)^{(x-x_0)/h}v_k(x)]+O(h^{2N+2}). \tag{2.2}$$

The first terms in this expansion satisfy equations of the form

$$\begin{aligned}u_1'&=f_y(x,y(x))u_1+\text{function of } x,\\ v_1'&=-f_y(x,y(x))v_1+\text{function of } x.\end{aligned}$$

Here $f_y(x,y(x))$ is the Jacobian matrix of f in (2.1) evaluated along the solution $y(x)$. The equation for the weakly stable component $v_1(x)$ has a character opposite that for y and u_1.

Gragg proposed smoothing at a fixed point x by

$$S(x;h)=1/4\eta(x-h;h)+1/2\eta(x;h)+1/4\eta(x+h;h).$$

In the form

$$S(x;h)=1/2[\eta(x;h)+\eta(x-h;h)+hf(x,\eta(x;h))],$$

it is clearer that this smoothing procedure requires an extra function evaluation in each integration to x. It is easy to see that

$$\begin{aligned}S(x;h)=y(x)&+h^2[u_1(x)+1/4y''(x)]\\ &+\sum_{k=2}^{N} h^{2k}[\tilde{u}_k(x)+(-1)^{(x-x_0)/h}\tilde{v}_k(x)]+O(h^{2N+2}).\end{aligned}$$

Thus the smoothing scheme replaces the weakly stable component $v_1(x)$ by $1/4$ $y''(x)$, which should have a behavior similar to $y(x)$, and alters the higher order terms in the error expansion (2.2).

For completeness we also state the semi-implicit midpoint rule investigated by Bader and Deuflhard. It uses an approximate Jacobian $J \doteq f_y(x_0, y_0)$. With the notation $\bar{f}(x,y) = f(x,y) - Jy$ it forms

$$\begin{aligned} \eta(x_0;h) &= y_0 \\ \eta(x_0+h;h) &= (I-hJ)^{-1}[\eta(x_0;h) + h\bar{f}(x_0, \eta(x_0;h))] \\ \eta(x+h;h) &= (I-hJ)^{-1}[(I+hJ)\eta(x-h;h) + 2h\bar{f}(x, \eta(x;h))]. \end{aligned} \tag{2.3}$$

Notice that if one takes $J=0$, this results in the explicit midpoint rule. An asymptotic h^2-expansion of the error is demonstrated in [1].

3. Ideal Smoothing

The object of Gragg's smoothing scheme is to remove the term

$$h^2(-1)^{(x-x_0)/h} v_1(x) = h^2 Q(x)$$

from the asymptotic expansion (2.2). Thus an ideal smoothing results in

$$\eta(x;h) - h^2 Q(x) = N(x;h).$$

There are two ways of extrapolation commonly seen - polynomial and rational. Rational extrapolation is not a linear process so that the usual absolute stability analysis does not apply. It is believed that the behavior is similar to that of polynomial extrapolation for which the analysis is possible. Stability is the whole point of smoothing, so we shall confine our attention to polynomial extrapolation. It proceeds as follows. The extrapolation is to be done at $x = x_0 + H$. A (fixed) sequence of integers $\{n_i\}$ is specified and integrations are done from x_0 to x with step sizes $h_i = H/n_i$. Let

$$T_{i,1} = \eta(x;h_i) \qquad i = 1, 2, \ldots.$$

A triangular array is generated by

$$T_{i,k} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left(\dfrac{n_i}{n_{i-k+1}}\right)^2 - 1}. \tag{3.1}$$

Each successive column k represents approximations to $y(x)$ of order $2k$.

It is now trivial to see the effect of the ideal smoothing. Let $T^{\eta}_{i,k}$ represent extrapolation of the values $\eta(x;h_i)$, i.e., no smoothing at all. Let $T^{N}_{i,k}$ represent extrapolation of $N(x;h_i)$, ideal smoothing, and let $T^{Q}_{i,k}$ represent extrapolation of the values $h_i^2 Q(x)$. Because of the obvious linearity of polynomial extrapolation, we have

$$T^{N}_{i,k} = T^{\eta}_{i,k} - T^{Q}_{i,k}.$$

However, it is obvious that

$$T^{Q}_{i,k} = 0 \qquad \text{for } k \geq 2,\ i \geq 2,$$

hence that

$$T_{i,k}^{N} = T_{i,k}^{\eta} \quad \text{for } k \geqq 2,\ i \geqq 2.$$

In words, the results of no smoothing and ideal smoothing are *exactly* the same after at least one extrapolation is done.

This result originates in the fact that the extrapolation process successively removes terms in the error expansion (2.2). This is true for rational as well as polynomial extrapolation. Thus there is no need to remove the weakly stable component by a special computation; extrapolation will do it automatically. As far as the numerical results go, Gragg's smoothing and/or rational extrapolation have different terms in the error expansion, but qualitatively the behavior is the same as for ideal smoothing with polynomial extrapolation.

In retrospect one should not find this result surprising. Smoothing only affects the accuracy when extrapolation is being used as a local method, and it removes exactly the kind of term extrapolation is supposed to remove.

There is a rather tricky aspect to comparing smoothing and not smoothing which arises in the way the codes decide if the numerical results are acceptable. The codes of Bulirsch and Stoer [2] and Hussels [6] use a diagonal convergence criterion. They estimate the error in the approximation of order $2k$, $T_{k,k}$, by a norm of the difference

$$T_{k,k} - T_{k+1,k+1}. \tag{3.2}$$

If this local error is smaller than a specified tolerance, the value accepted is $y(x_0+H) \doteq T_{k+1,k+1}$. Deuflhard [3] proposed the sub-diagonal criterion which estimates the error in $T_{k+1,k}$ by a norm of the difference

$$T_{k+1,k} - T_{k+1,k+1}. \tag{3.3}$$

If the error test is passed, he also accepts $T_{k+1,k+1}$. Stoer [13] has a procedure which can be regarded as an approximation to that of Deuflhard.

There are also differences among the various codes as to *when* they will first check for convergence. Deuflhard's code can check as early as possible, namely with $k=1$, whereas the others must do several extrapolations. We have been studying an algorithm for which the minimum k is 2, but the behavior of $T_{1,1}$ and $T_{2,1}$ is monitored for purposes other than error estimation. In all cases the result *accepted* comes from at least one extrapolation, hence is the same whether or not ideal smoothing is done (and essentially the same whether or not Gragg's smoothing is done). However, if $k=1$, the quantities $T_{1,1}$ and $T_{2,1}$ used in the various error tests are different, depending on smoothing. It is certainly easy to imagine that the weakly stable component might be so large that the error test is passed if it is smoothed out and failed if it is not. Thus, the code without smoothing might have to do one more subintegration at some cost to *recognize* that it has an acceptable result.

This difference in the acceptance tests arises only at the lowest possible order, so does not appear in most codes. Unfortunately this is a very important circumstance. The absolute stability regions of extrapolation, scaled for

equal work, shrink as the order is raised, at least for the standard choices of the sequence $\{n_i\}$. If stability is causing trouble, it is advantageous to use a low order. Of course ideal smoothing does not alter the stability of the methods, hence does not justify its cost. Gragg's scheme is not ideal so that it is possible that the stability "accidentally" obtained would justify its cost. We shall take this up below after considering some alternative schemes.

4. Other Smoothing Schemes

In this section we shall consider several other smoothing schemes. Bader invented a smoothing scheme for the semi-implicit midpoint rule. It resembles Gragg's smoothing scheme, but the resemblance is only superficial. Its goal is quite different, and the scheme *is* effective when solving stiff problems. The schemes of Gragg and Bader suggest a family of schemes for non-stiff problems. We have investigated the stability of members of this family. Gragg's scheme turns out to be as good as any. Shampine invented a smoothing scheme which is very cheap.

4.1 Bader's Scheme

Bader invented the smoothing

$$1/2[\eta(x+h;h)+\eta(x-h;h)]$$

to go with the semi-implicit midpoint rule (2.3). Although it resembles Gragg's scheme, it has a completely different purpose. The semi-implicit midpoint rule already has good stability properties; in particular, there is no weakly stable term to eliminate. Bader's goal was to improve the stability at infinity. The analysis of [1, §2.2] shows that the scheme is quite effective, with the consequence that the resulting numerical method has exceptionally good damping at infinity.

It should be clear that the resemblance of the smoothing procedures for the explicit and semi-implicit midpoint rule is purely formal. The objectives are entirely different so that it should not be surprising that smoothing might be important for the solution of stiff problems and not for the solution of non-stiff problems.

4.2 A Family of Smoothing Schemes

Bader's scheme could be applied to the explicit midpoint rule. With it and Gragg's scheme as examples, one is led to considering schemes based on a fixed linear combination of $\eta(x-h;h)$, $\eta(x;h)$, and $\eta(x+h;h)$. In order for such a combination to have an h^2-expansion it must have the form

$$\beta\eta(x-h;h)+(1-2\beta)\eta(x;h)+\beta\eta(x+h;h).$$

Obviously Bader's scheme has $\beta = 1/2$ and Gragg's scheme has $\beta = 1/4$. All the members of this family with $\beta \neq 0$ involve the same number of function evaluations.

The scaled stability regions for the formulas of orders $4, 6, \ldots, 20$, based on the double harmonic step size sequence, were computed for various β. First we considered $\beta = \pm 1/4, \pm 1/2, \pm 3/4, \pm 1$. Because the best stability was observed for $\beta = 1/4$, we then considered $\beta = 1/5$ and $1/3$. Of all the schemes considered, Gragg's scheme with $\beta = 1/4$ appeared the best. The derivation of Gragg's scheme suggests it should be effective, but not necessarily the best of this family. It was worth looking at the family even though nothing new turned up.

4.3 Shampine's Scheme

The idea of Shampine's scheme is very simple. The object of Gragg's scheme is to approximate the weakly stable term

$$h^2(-1)^{(x-x_0)/h} v_1(x)$$

in the asymptotic expansion (2.2). A function evaluation is made in each subintegration solely for this purpose. But $v_1(x)$ is the same for all the subintegrations - why approximate it every time? Shampine uses Gragg's scheme for the first subintegration with h_1 to approximate the weakly stable term and then simply adjusts the factor of h^2 for subsequent integrations. Now

$$\begin{aligned} q(x) &= \frac{\eta(x; h_1) - S(x; h_1)}{h_1^2} \\ &= v_1(x) - 1/4\, y''(x) + \sum_{k=2}^{N} h_1^{2k-2}[u_k(x) + (-1)^{(x-x_0)/h_1} v_k(x)] + O(h_1^{2N}) \\ &= v_1(x) - 1/4\, y''(x) - R(x; h_1). \end{aligned} \tag{4.1}$$

For other h the smoothing is simply

$$\begin{aligned} \eta(x; h) - h^2 q(x) = y(x) &+ h^2[u_1(x) + 1/4\, y''(x) + R(x; h_1)] \\ &+ \sum_{k=2}^{N} h^{2k}[u_k(x) + (-1)^{(x-x_0)/h} v_k(x)] + O(h^{2N+2}). \end{aligned}$$

Notice that this smoothing costs exactly one function evaluation regardless of the number of extrapolations. It does cost an extra vector of storage for the approximation to the weakly stable term. It resembles the ideal smoothing of Sect. 3 much more than Gragg's scheme does. Because it is a quadratic term in h^2, the observations of Sect. 3 made for ideal smoothing apply directly to it.

5. A Way to Exploit Smoothing

Modern ODE codes based on Adams methods enjoy a well-deserved reputation for robustness. They achieve this by a careful monitoring of basic

assumptions and by verification of anticipated behavior. For example, the step size and the number of terms used in the underlying approximate Taylor series expansion of the solution are manipulated so that the terms decrease in a regular manner. Perhaps because of our experience with writing Adams codes, we believe that the robustness of extrapolation codes could be significantly enhanced by a more critical evaluation of the assumptions underlying the current algorithms. Here we propose a natural test based on smoothing.

It is, of course, a fundamental hypothesis that the asymptotic expansion (2.2) exist. In practical computation this not always true. The expansion might not exist on the interval $[x_0, x]$ because f is not smooth on the whole interval, indeed discontinuous f are by no means rare. (More common is that the number of terms N is limited by the smoothness of the problem.) If the step size H is too small or if f is not evaluated accurately, the numerical value actually computed for $\eta(x;h)$ might be so contaminated by roundoff errors that the expected asymptotic behavior is not present. If the step size H is too large, the midpoint rule might be unstable. There is an important distinction here. If the step size h is small enough that the asymptotic expression (2.2) is valid, the weakly stable term reveals the poor stability of the midpoint rule, but the situation is more or less under control. If the step size h is too large, overflow is quite possible. Years ago in a comparison of ODE solvers [12] we pointed out that extrapolation codes can overflow for this reason, and recently [10] we have had experience with this difficulty in developing a type-insensitive code.

Even if the asymptotic expansion exists, we also need to have a step size small enough that the anticipated behavior is actually present. On another occasion we shall propose suitable monitoring devices. Here we just need the observation that we intend to extract $y(x)$ from the numerical values $\eta(x;h)$, hence h must be small enough that $y(x)$ be represented in these values.

Our proposal is very simple. Let us monitor the difference between the smoothed and unsmoothed values in the first subintegration. (For example, $h_1^2 q(x)$ from (4.1) is this difference for both Gragg's and Shampine's smoothings.) If this difference is large compared to the smoothed value, the step is to be rejected and attempted again with a smaller H. It is a fundamental practical hypothesis that the quantity we propose monitoring be of modest size compared to an approximate solution. A large value is a sure sign of trouble. Reducing the step size is an appropriate response for all the difficulties we have mentioned, with the exception of H too small. This last case is easily handled by a test along the lines of [8].

The test proposed is not a delicate one and in our opinion should be supplemented with other tests. Its great virtue is that it is done very early in the extrapolation process, so that serious trouble can be recognized before much effort is wasted. Because it is not delicate and because it is activated in circumstances when we do not know how the solution behaves, we do not know how much to reduce the step size. In other kinds of ODE solvers, the traditional response in such a situation is to reduce the step size by a fixed amount, e.g., halve it.

6. Which is the Best?

It is not clear how one should proceed in the local use of extrapolation of the explicit midpoint rule. In this section we shall consider several issues which affect the decision.

In [9] one of us argued vigorously that the robustness of a method is helped by evaluating the function at arguments spanning the whole step from x_n to x_n+H. The various smoothing procedures do this, but that based on not smoothing never evaluates at the endpoint x_n+H. The matter is most important for the stiff problems considered in [9], and the extrapolated midpoint rule should not be used for such problems. Extrapolation is an adaptive process which does cause the neighborhood of the endpoint to be probed if there is any hint of difficulty in the rest of the step. For these reasons, the advantage that smoothing has in this respect is not very impressive.

Other things being equal, the lower the cost per step, the better. No smoothing is obviously best in this regard with Shampine's smoothing a close second. Gragg's smoothing does not add terribly to the cost of a step, but the extra cost is a disadvantage in severe circumstances causing frequent step failures or restricting the step size drastically.

An issue difficult to quantify is the effect smoothing has on the process of extrapolation in finite precision arithmetic. As is well known, the extrapolation procedure (3.1) amplifies the arithmetic errors present in the results of the subintegrations $T_{i,1}$. Extracting a substantial term by smoothing, in effect, augments the precision available for the extrapolation. As we have observed, this is little different from not smoothing because the first extrapolation extracts virtually the same term.

The goal of Gragg's smoothing was to improve the stability. The absolute stability regions depend on the sequence $\{n_i\}$ used to specify the rate at which the step size h_i is decreased in the successive subintegrations. Of course the work in terms of function evaluations also depends on this sequence. To properly compare methods, one must scale the regions by the cost of the formula. We have already seen that any quadratic smoothing, such as the ideal smoothing or Shampine's, leads to the same absolute stability region as not smoothing. Thus for any choice of $\{n_i\}$, the stability region for Shampine's smoothing is obtained from that of not smoothing by contracting the region in the ratio $c/(c+1)$ where c is the cost in function evaluations of the unsmoothed formula. Because c increases fairly rapidly as the order of extrapolation increases, there is no practical difference in the stability of these procedures except at low orders.

To proceed further with our examination of stability properties, we had to specify $\{n_i\}$ and compute the stability regions in the usual way. We chose to follow Deuflhard in the April 30, 1981, version of his code DIFEX 1, which seems to be the most efficient extrapolation code at this time. He uses the slowest possible rate of decrease of step size, the double harmonic sequence {2, 4, 6, ...}, with the maximum number of extrapolations limited to 9. Associated with the report [11] is a microfiche which gives the scaled stability regions for

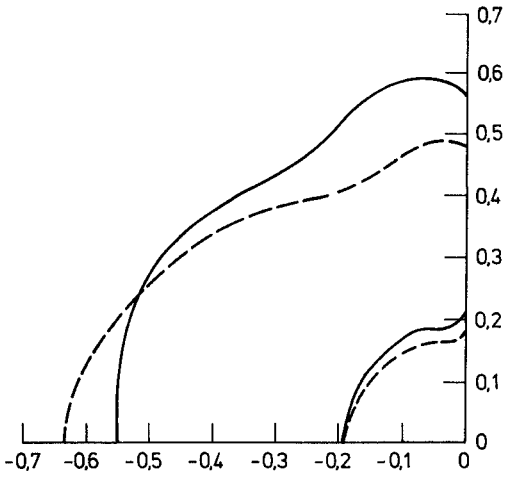

Fig. 1. Scaled stability regions of orders 4 and 10. The larger regions correspond to order 4. Dashed line identifies result with Gragg's smoothing, solid line identifies result with no smoothing

Gragg's smoothing and no smoothing for orders 4, 6, ..., 20. In Fig. 1 we display the results for orders 4 and 10.

The situation for the higher orders is easily described. At orders 10 and up, the stability region for not smoothing includes that for Gragg's smoothing. Shampine's smoothing is virtually identical to not smoothing. Although the situation is clear enough, it is not the basis for a choice because the differences are not large enough to be important in practice.

At order 4 the stability regions for Gragg's smoothing and not smoothing are comparable. Their shapes change gradually through orders 6 and 8 to those plotted of order 10. Not smoothing is somewhat to be preferred because the boundary of the region is more circular, hence the method treats problems more uniformly. At order 4 the region for not smoothing is contracted by 5/6 to yield the region for Shampine's smoothing; it is therefore worst of the three at this order. At order 6 the contraction factor is 10/11 for which the difference in stability is already of little practical importance.

A key factor is when one starts testing for convergence in the extrapolation array. As explained in Sect. 3 it may be necessary to smooth to recognize an acceptable result as early as one would like. Smoothing also is a little more robust by evaluating f throughout the step and may be a little better in finite precision arithmetic. A strong argument for smoothing is the robustness supplied by the test of Sect. 5. Our preference is to smooth. It is difficult to choose between Gragg's and Shampine's smoothings. Gragg's smoothing will be somewhat more efficient if stability is causing the step size to be restricted. Shampine's smoothing will be somewhat more efficient if the step size is restrained for other reasons, e.g., output or a lack of smoothness, or if there are step failures.

7. Conclusion

It is remarkable that the role of Gragg's smoothing of the extrapolated midpoint rule has been so long misunderstood, and even more remarkable that the extrapolation process itself already accomplishes Gragg's goal. As we have seen, the advantages of smoothing do not come from stability at all. Besides coming to a new understanding of the role of smoothing, we proposed a new smoothing with attractive features and an easy way to improve the robustness of extrapolation codes.

References

1. Bader, G., Deuflhard, P.: A semi-implicit mid-point rule for stiff systems of ordinary differential equations. Preprint Nr. 114. University of Heidelberg, 1981
2. Bulirsch, R., Stoer, J.: Numerical treatment of ordinary differential equations by extrapolation methods. Numer. Math. **8**, 1–13 (1966)
3. Deuflhard, P.: Order and stepsize control in extrapolation methods. Numer. Math. (to appear)
4. Gragg, W.B.: Repeated extrapolation to the limit in the numerical solution of ordinary differential equations. Thesis, University of California at Los Angeles, 1964
5. Gragg, W.B.: On extrapolation algorithms for ordinary initial value problems. SIAM J. Numer. Anal. **2**, 384–403 (1965)
6. Hussels, H.G.: Schrittweitensteuerung bei der Integration gewöhnlicher Differentialgleichungen mit Extrapolationsverfahren. Diplomarbeit Universität Köln, Math. Inst. 1973
7. Milne, W.E., Reynolds, R.R.: Stability of a numerical solution of differential equations. J. ACM **6**, 196–203 (1959) and **7**, 45–56 (1960)
8. Shampine, L.F.: Limiting precision in differential equation solvers. Math. Comput. **28**, 141–144 (1974)
9. Shampine, L.F.: Implementation of Rosenbrock codes. ACM Trans. Math. Software **8**, 93–113 (1982)
10. Shampine, L.F.: Type-insensitive ODE codes based on extrapolation methods. SIAM J. Sci. Stat. Comput. (to appear)
11. Shampine, L.F., Baca, L.S.: Should the extrapolated midpoint rule be smoothed? Report SAND 82-0317. Sandia National Laboratories, Albuquerque, NM, 1982
12. Shampine, L.F., Watts, H.A., Davenport, S.M.: Solving nonstiff ordinary differential equations – the state of the art. SIAM Rev. **18**, 376–411 (1976)
13. Stoer, J.: Extrapolation methods for the solution of initial value problems and their practical realization, pp. 1–21. In: Proc. Conf. on Numer. Soln of Ordinary Differential Equations, University of Texas at Austin. Nr. 362 in Lecture Notes in Math. Berlin, Heidelberg, New York: Springer, 1974
14. Stoer, J., Bulirsch, R.: Introduction to numerical analysis. Berlin, Heidelberg, New York: Springer, 1980

Received June 25, 1982