stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE NW 130/82 JUNI
(DEPARTMENT OF NUMERICAL MATHEMATICS)

K. DEKKER & J.G. VERWER

ESTIMATING THE GLOBAL ERROR OF RUNGE-KUTTA APPROXIMATIONS

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 65L05

1982 CR. Categories: 5.17

Estimating the global error of Runge-Kutta approximations *)

by

K. Dekker & J.G. Verwer

ABSTRACT

The user of a code for solving the initial value problem for ordinary differential systems is normally left with the difficult task of assessing the accuracy of the numerical result returned by the code. Even when the code reports an estimate of the global error, the question may remain whether this estimate is correct, i.e. whether the user can rely on the estimate. This paper proposes a simple idea of measuring the reliability of the global error estimate with the aim of assisting the user in the validation of the numerical result. The idea is put into practice with the existing code GERK (ACM Algorithm 504) developed by Shampine and Watts.

KEY WORDS & PHRASES: *Numerical analysis, Mathematical software, Ordinary differential equations, Runge-Kutta methods, Global error estimation, Richardson extrapolation*

---

*) This report will be submitted for publication elsewhere.

1. INTRODUCTION

This paper deals with the problem of computing reliable estimates for the global error of numerical approximations to the exact solution y(x) of the initial value problem

$$(1.1)\qquad \dot{y} = f(x,y), \qquad a \le x \le b, \quad y(a) = y_a,$$

where f is supposed to be a sufficiently smooth, real-valued vector function. We restrict ourselves to non-stiff systems and classical explicit Runge-Kutta approximations (see e.g. [2,5,9]).

Let us first introduce some notations and definitions. The initial value problem (1.1) is integrated on a grid

$$(1.2)\qquad G_N = \{x_n \in [a,b],\ n = 0(1)N, \text{ with } x_0 = a,\ x_{n-1} < x_n,\ x_N = b\}$$

to obtain the approximations y_n, where $y_0 = y_a$ and, for $n = 0,1,\ldots,N-1$,

$$(1.3)\qquad \begin{aligned} y_{n+1} &= y_n + h_n \sum_{i=1}^{m} b_i k_i, \quad h_n = x_{n+1} - x_n, \\ k_i &= f(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{i-1} a_{ij} k_j). \end{aligned}$$

The scalar parameters a_{ij}, b_i and c_i define the Runge-Kutta scheme. The grid G_N needs not to be uniform and, as is common practice, may be determined during the integration process through some stepsize control mechanism. It will be assumed that for N sufficiently large the minimal and maximal steplengths behave like $O(N^{-1})$. More specifically, we assume the existence of a piecewise constant function $\theta:[a,b] \to [\theta_{min},\theta_{max}]$, $0 < \theta_{min} \le \theta_{max}$, such that for N sufficiently large $h_n = \theta(x_n)H_N = \theta(x_n)\theta_{max}/N$, $n = 0(1)N-1$. If this natural assumption is satisfied, we are assured of the existence of an asymptotic expansion in H_N for the global discretization errors

$$(1.4)\qquad \varepsilon_n := y_n - y(x_n), \quad n = 1,\ldots,N.$$

See STETTER [9] for a detailed analysis. If we let f be M times differentiable (in some neighbourhood of y(x)), then functions e_j, $j = p,\dots,M$, exist independent of $H = H_N$ such that

$$(1.5) \qquad \varepsilon_n = \sum_{j=p}^{M} H^j e_j(x_n) + O(H^{M+1}).$$

Here p denotes the order of accuracy of the Runge-Kutta method. The existence of these asymptotic expansions for ε_n forms the basis for most of the error estimation techniques.

The usual approach in the literature on global error estimation is to compute a first approximation for ε_n, $est_n^{(1)}$ say, which satisfies a relation of the form

$$(1.6) \qquad est_n^{(1)} = H^p e_p(x_n) + H^{p+1} v(x_n) + O(H^{p+2}).$$

Here v(x) is some function different from $e_{p+1}(x)$. The user of a code which delivers an estimate like $est_n^{(1)}$ will normally be interested in the global error. Anyhow, it is reasonable to assume that most users wish to rely on the estimate. Otherwise the extra effort spent is of no use. In this respect global error estimation has to be approached in an essentially different way than local error estimation. The importance of local error estimation lies in stepsize control, while the reliability of the local estimate is of less importance than its additional costs. When reporting global error estimates however, one should make higher demands on reliability than on efficiency for the reason just mentioned. In fact, from the user's point of view, the computation of a highly reliable global error estimate might be considered as important as an efficient computation of the approximation itself.

These considerations lead us to the conclusion that it might be desirable to compute a second and more accurate estimate $est_n^{(2)}$ satisfying

$$(1.7) \qquad est_n^{(2)} = H^p e_p(x_n) + H^{p+1} e_{p+1}(x_n) + O(H^{p+2}),$$

and to compare this result with the first estimate $est_n^{(1)}$.

One way to do this is to check whether

$$(1.8) \qquad r_{est} \overset{c}{:=} est_n^{(2)} / est_n^{(1)}, \quad \overset{c}{=} \text{ means componentwise operation,}$$

is sufficiently close to one. The quantity r_{est} is a first order approximation to the true error ratio r_{true}, i.e. $r_{est} = r_{true} + O(H)$, where

$$(1.9) \qquad r_{true} :\overset{c}{=} est_n^{(2)}/\varepsilon_n.$$

If r_{est} is close to one and $est_n^{(2)}$ is of an acceptable magnitude, one has a strong indication that $est_n^{(2)}$ is an accurate estimate. We believe that the reliability of automatic codes for our initial value problem (1.1) is greatly enhanced if the asymptotic quality of the global error estimation can be verified.

The objective of this paper is to put this idea into practice and to show that it is useful. Our starting point is the existing Runge-Kutta code GERK developed by SHAMPINE & WATTS [8]. This code is based on a Fehlberg (4.5)-pair [1] and computes a first estimate $est_n^{(1)}$ by means of global Richardson extrapolation. The decision to concentrate on GERK is based on the fact that this code is very suitable for the task we have set ourselves.

2. GERK AND GLOBAL RICHARDSON EXTRAPOLATION

Global Richardson extrapolation involves parallel integration with the same method on different grids. The use of Richardson extrapolation for estimating the global discretization error of one-step integration methods is well-known (see HENRICI [2], p.81, LETHER [6] and STETTER [9], p. 157). When using non-equidistant grids, which we assume, it is only allowed to change the stepsize at grid points where the various approximations are combined in the extrapolation process.

SHAMPINE & WATTS [8] have implemented global Richardson extrapolation on top of the Runge-Kutta code RKF45 which is based on a Fehlberg (4.5)-pair. They called the resulting code GERK. This code computes two parallel solutions and estimates the global error at the finest grid (grid G_{2N} of Fig. 1). By computing a third parallel solution, on a grid G_{3N} as shown in Fig. 1, the same idea can be used for obtaining two estimates $est_n^{(1)}$ and $est_n^{(2)}$ of the global error at the grid G_{3N}. Having two estimates of the global error available we then can measure the accuracy of these estimates as outlined in the introduction.

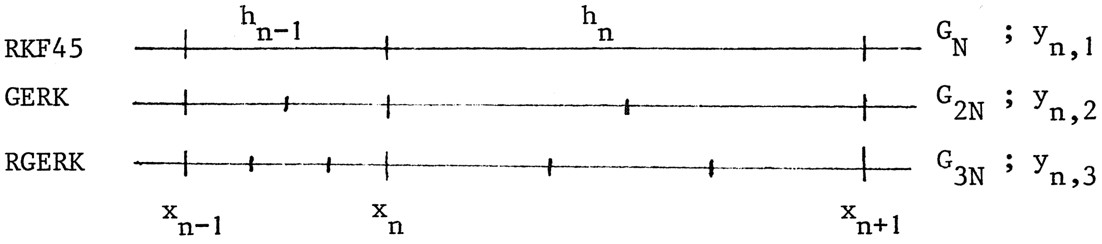

Figure 1.

Let us give some details. Consider three coherent grids as shown in Figure 1. Apply on these grids some Runge-Kutta method of order p to obtain at the points $x = x_n$ the approximations $y_{n,1}$, $y_{n,2}$ and $y_{n,3}$. Let $\varepsilon_{n,i} := y_{n,i} - y(x_n)$. Then

$$(2.1) \qquad \varepsilon_{n,i} = \sum_{j=p}^{M} (H/i)^j e_j(x_n) + O(H^{M+1}), \qquad i = 1,2,3.$$

We now define our estimates by

$$(2.2) \qquad \begin{aligned} est_n^{(1)} &:= (y_{n,2} - y_{n,3})/(1.5^p-1), \\ est_n^{(2)} &:= (1+\eta)est_n^{(1)} - \eta(y_{n,1} - y_{n,3})/(3^p-1), \end{aligned}$$

where $\eta = (1-(1.5^{p+1}-1)/(1.5^p-1))/((1.5^{p+1}-1)/(1.5^p-1) - (3^{p+1}-1)/(3^p-1))$. Relations (1.5) - (1.7) are satisfied if H is replaced by H/3 and ε_n by $\varepsilon_{n,3}$. Hence we estimate the error of the most accurate solution $y_{n,3}$.

The code GERK computes the solutions $y_{n,1}, y_{n,2}$ on the grids G_N, G_{2N} and delivers at the points x_n the global error estimate $(y_{n,1}-y_{n,2})/(2^p-1)$, where $p = 5$. Thus it also reports the more accurate solution $y_{n,2}$. The stepsize selection of GERK is based on a mixed relative-absolute local error control on the coarsest grid G_N by using the imbedded 4-th order scheme (see [8], section 4 for details). Control on the coarsest grid protects the parallel integration, where no control is performed, against instability. It shall be clear now that it is possible to place our estimation procedure on top of GERK without drastic changes. Only minor modifications are required. These are the implementation of the third parallel integration on G_{3N} and the implementation of the estimates (2.2). Further, the modified code, which we have named RGERK, should report the numerical solution $y_{n,3}$, the estimates $est_n^{(2)}$ and r_{est}.

We wish to emphasize that we did not modify the stepsize and local error control. This implies that for a given value of the local tolerance input parameters RGERK computes exactly the same solutions $y_{n,1}$ and $y_{n,2}$ as GERK does. GERK, in turn, computes the same solution $y_{n,1}$ as RKF45. It should be noted that in normal situations the global error of $y_{n,3}$ shall be somewhat smaller than the prescribed local tolerance values. This is because we report the solution computed on the finest grid G_{3N}, whereas the local error and stepsize control is performed on the coarsest grid G_N. It is perhaps clarifying to observe that the grids G_N, G_{2N} and G_{3N} are determined in the course of the integration, viz. by the stepsize control.

Finally, a few remarks on the cost ratios of RKF45, GERK and RGERK. When we consider the coarsest grid G_N, RKF45 uses six f-evaluations per step, GERK eightteen, and RGERK thirty-six. However, they report the solution at G_N, G_{2N} and G_{3N}, respectively. Hence one has to take the accuracy at the three grids into consideration. On the asymptotic basis we thus arrive at the ratios 6 : 9 : 12. In practice the cost ratios, in terms of the numbers of f-evaluations, will slightly differ from the asymptotic ratios. Normally they will be somewhat larger. For further practical information we would like to refer to SHAMPINE & WATTS [8].

3. MEASURING THE RELIABILITY OF THE GLOBAL ERROR ESTIMATES

A code like GERK computes a numerical solution of (1.1) and reports at the same time an estimate $est_n^{(1)}$ of the global discretization error. Experiments reported by Shampine and Watts show that their estimate $est_n^{(1)}$ will be reliable in many cases. Nevertheless, in real life computation the user of GERK is left with the difficult task of assessing the accuracy of the estimate himself. If it is in doubt, which already may be very difficult to establish, one could apply the code a second time with a more stringent local error tolerance and then by comparison try to get more insight in the accuracy of the reported quantities. The theoretical support of the technique of reintegration is difficult to give, however, when using a stepsize determined by local error control. To assist the user in his validation of the numerical result we prefer to compute the quantity $r_{est} \overset{c}{=} est_n^{(2)}/est_n^{(1)}$ introduced in equations (1.8), (2.2). From a theoretical point of view, the use of r_{est} is fully justified. In this section we will consider r_{est} in some more detail.

For convenience of presentation we now restrict ourselves to a single differential equation. First we introduce the quantity

$$(3.1) \qquad r := (1.5^p-1)(y_{n,1}-y_{n,3})/(y_{n,2}-y_{n,3})(3^p-1),$$

and observe that r_{est} can be written as a function of r, viz.

$$(3.2) \qquad r_{est}(r) = 1 + \eta - \eta r.$$

In fact r has a similar meaning as r_{est} being the quotient of two different estimates of $\varepsilon_n = \varepsilon_{n,3}$. Equation (3.2) shows the range of r_{est}. The equality $r_{est}(1) = 1$ follows immediately from the observation that both r and r_{est} tend to 1 if $H \to 0$. For $p = 5$ we have

$$(3.3) \qquad r_{est}(r) = (422 - 121r)/301.$$

Note that $r_{est}(0) \simeq 1.4$, which means that if r_{est} is close to 1.4 at least one of the estimates $est_n^{(1)}$ or $est_n^{(2)}$ is very inaccurate. Generally, too small or too large r_{est}-values mean that at least one of these estimates is wrong. One should observe, however, that $est_n^{(2)}$ is a more accurate estimate than $est_n^{(1)}$ (cf. (1.6),(1.7)). In other words, r_{est} normally will be a conservative approximation for $r_{true} = est_n^{(2)}/\varepsilon_n$.

The main question is of course, which range of r_{est}-values is still meaningfull. We have tried to answer this question in two ways, viz. theoretically and experimentally. The experiments are discussed in the next section. Here we discuss our theoretical answer.

Assume that in equation (2.1) the errors $\varepsilon_{n,i}$ can be represented by infinite series. Let $e_p(x_n) \neq 0$ and introduce

$$(3.4) \qquad \alpha_j := H^{j-p} e_j(x_n)/e_p(x_n), \qquad j \geq p.$$

Substitution into (3.1) yields

$$(3.5)\qquad r = \frac{1 + (1-3^{-p})^{-1} \sum_{j=p+1} (1-3^{-j})\alpha_j}{1 + (2^{-p}-3^{-p})^{-1} \sum_{j=p+1} (2^{-j}-3^{-j})\alpha_j}$$

By imposing bounds for α_j, $j \geq p+1$, one can obtain bounds for r, r_{est} and r_{true}. The idea is to compare these bounds. We will consider α_j-values satisfying

$$(3.6)\qquad |\alpha_j| \leq c^{j-p},\ j \geq p+1 \text{ and } 0 < c < 1.$$

The smaller c, the more dominance of the error term $H^p e_p(x_n)$ is supposed by this condition. The following results were obtained.

LEMMA 1. *Let* $p = 5$ *and denote* $\alpha = (\alpha_6,\alpha_7,\ldots)$, $\alpha^- = (-c,-c^2,\ldots)$ *and* $\alpha^+ = (c,c^2,\ldots)$. *Suppose that* $0 < c \leq 2/7$. *For all sequences* α *the elements of which satisfy condition* (3.6), *it then holds that*

$$(3.7)\qquad r(\alpha^-) = \frac{1-\frac{c}{2}}{1-c}\ \frac{1 - \frac{848}{363}c + \frac{2}{3}c^2}{1 - \frac{860}{633}c + \frac{1}{3}c^2} \leq r(\alpha) \leq r(\alpha^+) = \frac{1-\frac{c}{2}}{1-c}\ \frac{1-\frac{40}{121}c}{1-\frac{65}{211}c}\ .$$

PROOF. Substitute $p = 5$ into (3.5) and write $r(\alpha) = N(\alpha)/D(\alpha)$. Differentiating $r(\alpha)$ to α_k, $k \geq 6$, yields

$$D^2(\alpha)\frac{\partial r(\alpha)}{\partial \alpha_k} = \frac{243}{242}(1-3^{-k}) - 32\frac{243}{211}(2^{-k}-3^{-k}) +$$

$$32\frac{243}{242}\frac{243}{211}\sum_{j=6}\{(1-3^{-k})(2^{-j}-3^{-j}) - (2^{-k}-3^{-k})(1-3^{-j})\}\alpha_j \geq$$

$$\frac{243}{242} - \frac{1}{2}\frac{243}{211} - 32\ \frac{243}{242}\ \frac{243}{211}\sum_{j=6}(2^{-j}+2^{-k})c^{j-5} =$$

$$\frac{243}{242}\ \frac{90}{211} - \frac{243}{242}\ \frac{243}{211}\left(\frac{c}{2-c} + 2^{5-k}\frac{c}{1-c}\right) \geq$$

$$\frac{243}{242*211}\left(90 - 243\left(\frac{c}{2-c} + \frac{1}{2}\frac{c}{1-c}\right)\right)$$

for all $k \geq 6$ and $0 < c < 1$. The last expression is positive for all c between 0 and 2/7. Further,

$$D(\alpha) = 1 + 32\,\frac{243}{211} \sum_{j=6} \{(2^{-j}-3^{-j})\alpha_j > 1 - \frac{243}{211} \sum_{j=6} 2^{5-j}c^{j-5} > 0$$

if $0 < c \leq 2/7$. Hence for all $k \geq 6$, $\partial r(\alpha)/\partial\alpha_k > 0$ if $0 < c \leq 2/7$, which implies that for these values of c, $r(\alpha)$ takes its minimum and maximum at $\alpha = \alpha^-$ and $\alpha = \alpha^+$, respectively. □

Substitution of (3.4) into r_{true} given by (1.9) yields

$$(3.8)\quad r_{\text{true}} = \frac{1+(1+\eta)(2^{-p}-3^{-p})^{-1}\sum_{j=p+1}(2^{-j}-3^{-j})\alpha_j - \eta(1-3^{-p})^{-1}\sum_{j=p+1}(1-3^{-j})\alpha_j}{1+3^p\sum_{j=p+1}3^{-j}\alpha_j}$$

and, for p = 5

$$(3.9)\qquad r_{\text{true}} = \frac{1 + 243\frac{64}{301}\sum_{j=6}(2^{-j}-3^{-j} - \frac{1}{128} + \frac{1}{128}3^{-j})\alpha_j}{1 + 243\sum_{j=6}3^{-j}\alpha_j}.$$

<u>LEMMA 2</u>. *Let* $p = 5$ *and denote* $\alpha = (\alpha_6,\alpha_7,\ldots)$, $\alpha^- = (-c,-c^2,\ldots)$, $\alpha^* = (-c,c^2,c^3,\ldots)$. *Suppose that* $0 < c \leq 602/845$. *For all sequences* α *the elements of which satisfy condition* (3.6), *it then holds that*

$$(3.10)\qquad r_{\text{true}}(\alpha^*) = 1 - \frac{81}{602}\,\frac{c^2}{(1-c)(1-\frac{c}{2})(1-\frac{2c}{3}+\frac{2c^2}{9})} \leq r_{\text{true}}(\alpha) \leq$$

$$\leq 1 + \frac{81}{602}\,\frac{c^2}{(1-c)(1-\frac{c}{2})(1-\frac{2c}{3})} = r_{\text{true}}(\alpha^-).$$

<u>PROOF</u>. We write $r_{\text{true}}(\alpha) = P(\alpha)/Q(\alpha)$ and note that $Q(\alpha)$ is positive, because

$$Q(\alpha) = 1 + 243 \sum_{j=6} 3^{-j}\alpha_j \geq 1 - \sum_{j=6} (c/3)^{j-5} > 0,$$

if $0 < c < 3/2$. Similarly, we have

$$P(\alpha) = 1 + 243\,\frac{64}{301} \sum_{j=6} (2^{-j}-3^{-j} - \frac{1}{128} + \frac{1}{128}\,3^{-j})\alpha_j \geq$$

$$\geq 1 - \frac{243}{602} \sum_{j=6} c^{j-5} > 0$$

if $0 < c \le \frac{602}{845}$. Differentiating $r_{true}(\alpha)$ to α_k, $k \ge 7$, yields

$$Q^2(\alpha)\,\frac{\partial r_{true}(\alpha)}{\partial \alpha_k} = 243\,\frac{64}{301}\,(2^{-k}-3^{-k}-\frac{1}{128}+\frac{1}{128}\,3^{-k})Q(\alpha) - \frac{243}{3^k}\,P(\alpha) \le 0,$$

as $2^{-k}-3^{-k}-\frac{1}{128}+\frac{1}{128}3^{-k} < 0$ for $k \ge 7$ and both $Q(\alpha)$ and $P(\alpha)$ are positive. Finally,

$$Q^2(\alpha)\,\frac{\partial r_{true}(\alpha)}{\partial \alpha_6} = \frac{1}{3}(Q(\alpha) - P(\alpha)).$$

At the maximum we have $P(\alpha)/Q(\alpha) > 1$, so that the derivative with respect to α_6 is negative; thus the maximum is obtained for $\alpha_6 = -c$ and $\alpha_k = -c^{k-6}$, $k \ge 7$. At the minimum $P(\alpha)/Q(\alpha) < 1$, so the derivative with respect to α_6 is positive and the minimum is obtained for $\alpha_6 = -c$, $\alpha_k = c^{k-6}$, $k \ge 7$. □

Using (3.3) Lemma 1 yields bound for r_{est} under condition (3.6), where $0 < c \le 2/7$. Under the same condition Lemma 2 yields bounds for r_{true}, but now for $0 < c \le 602/845$. Table 2 shows these bounds for a number of values for $c \le 602/845$.

	r_{est}		r_{true}	
c	lower	upper	lower	upper
$\frac{1}{100}$	.998	1.002	1.000	1.000
$\frac{1}{30}$	.993	1.007	1.000	1.000
$\frac{1}{10}$	.978	1.023	.998	1.002
$\frac{1}{5}$	.951	1.060	.991	1.009
$\frac{1}{4}$	.936	1.087	.985	1.015
$\frac{2}{7}$	.923	1.110	.978	1.022
$\frac{1}{2}$			.876	1.135
$\frac{7}{10}$			.474	1.634

Table 2. Bounds for r_{est} and r_{true}.

4. PERFORMANCE OF THE MODIFIED GERK CODE

The purpose of this section is to give practical evidence to our view that the use of a second estimate $est_n^{(2)}$ greatly enhances the reliability of the global error estimation procedure. Further we want to give an answer to the question of section 3 how to interpret a reported r_{est}-value.

We have subjected the code RGERK to various experiments. In sections 4.1 - 4.5 we present results, in some detail, for five different example problems. In section 4.6 we have collected some statistics on the well-known test set of Hull et al. [3]. All computations have been carried out on a CDC Cyber 750. The arithmetic precision of this computer is about 14 decimal digits (48 bits).

4.1. An unstable problem

To begin with we give an example of the behaviour of RGERK on the mathematically unstable problem

$$(4.1) \qquad \dot{y} = 10(y-x^2), \quad y(0) = 0.02,$$

which has the general solution $y(x) = 0.02 + 0.2x + x^2 + ce^{10x}$. Following [7] we solve this problem on the interval [0,2] using pure relative local error control. Table 3 contains for various tolerances the global error ε_n, r_{true} and r_{est}, measured at the end point $x = 2$; ND denotes the number of f-evaluations. For the sake of comparison, we have also inserted results of GERK. The numbers in the parentheses stand for exponents of 10.

- log of tolerance	RGERK				GERK		
	ε_n	r_{true}	r_{est}	ND	ε_n	r_{true}	ND
1	-1.5(+4)	.77	1.34	181	-7.2(+4)	.11	91
2	-1.2(+3)	.96	1.24	294	-7.4(+3)	.38	150
3	-6.1(+1)	1.00	1.12	602	-4.2(+2)	.68	314
4	-4.4(+0)	1.00	1.06	1003	-3.1(+1)	.83	517
5	-4.0(-1)	1.00	1.04	1491	-2.9(+0)	.90	771
6	-4.0(-2)	1.00	1.02	2011	-2.9(-1)	.94	1021
7	-4.0(-3)	1.00	1.02	2680	-3.0(-2)	.96	1348
8	-4.1(-4)	1.00	1.01	4084	-3.1(-3)	.97	2050
9	-4.3(-5)	.95	1.00	6450	-3.1(-4)	.98	3228
10	-7.9(-6)	.49	.98	10266	-3.4(-5)	.93	5136
11	-5.7(-6)	.14	.86	13038	-1.2(-5)	.80	6522

Table 3. Mathematically unstable problem (4.1)

We see that for (4.1) most r_{true}-values are surprisingly close to one, even for the larger tolerances for which the errors ε_n are very large due to the unstable growth. Hence on this problem the estimate $est_n^{(2)}$ is very reliable. Furthermore, we can observe a very good agreement between the r_{est}-values and r_{true}-values. Recall that r_{est} close to one indicates that both $est_n^{(1)}$ and $est_n^{(2)}$ are very good estimates. Further, we once more note that r_{est} is a conservative estimate for r_{true} since $est_n^{(2)}$ is more accurate than $est_n^{(1)}$. This implies that normally r_{true} will be closer to one than r_{est} (cf. the remark in section 3). In fact we have observed this in all our experiments.

Note that for the tolerance values 10^{-10}, 10^{-11} the estimates are contaminated by roundoff errors. For RGERK the effect is already observable for 10^{-9}. We refer to SHAMPINE & WATTS [8] for some discussion on roundoff. Actually, since the order of accuracy is $p = 5$, the codes are not meant to be used with very small tolerance values. Furthermore, the control is performed on the coarsest grid, while the reported approximation comes from the finest grid. For stable problems, we advise to choose tolerance values between 10^{-1} and 10^{-7}, say.

4.2. The restricted 3-body problem

Our second example is the restricted 3-body problem

$$
\begin{aligned}
&\ddot{u}_1 = 2\dot{u}_2 + u_1 - \mu^*(u_1+\mu)/r_1^3 - \mu(u_1-\mu^*)/r_2^3, \\
&\ddot{u}_2 = -2\dot{u}_1 + u_2 - \mu^* u_2/r_1^3 - \mu u_2/r_2^3, \\
&r_1 = [(u_1+\mu)^2+u_2^2]^{\frac{1}{2}}, r_2 = [(u_1-\mu^*)^2+u_2^2]^{\frac{1}{2}},\ \mu = 1/82.45,\ \mu^* = 1-\mu, \\
&u_1(0) = 1.2,\ \dot{u}_1(0) = 0,\ u_2(0) = 0,\ \dot{u}_2(0) = -1.04935750983032,
\end{aligned}
\tag{4.2}
$$

which has also been used by Shampine and Watts. Using absolute local error control we have integrated this difficult problem over the first period P = = 6.192169331131964. Table 4 contains results for the endpoint x = P. These results belong to that component for which the error of RGERK, in absolute value, is maximal.

- log of tolerance	RGERK ε_n	RGERK r_{true}	RGERK r_{est}	RGERK ND	GERK ε_n	GERK r_{true}	GERK ND
1	-2.1(+1)	-.44	1.40	355	2.3(+1)	-.03	193
2	-1.3(+1)	-.18	1.39	1494	-1.9(+0)	-.03	810
3	1.6(-2)	.95	1.40	2009	8.7(-2)	-.03	1055
4	2.1(-5)	1.05	1.27	2856	1.3(-4)	.30	1506
5	1.9(-6)	1.04	1.14	4171	1.3(-5)	.64	2191
6	1.4(-7)	1.02	1.06	6257	1.0(-6)	.83	3269
7	7.8(-9)	1.03	1.04	9445	5.9(-8)	.89	4873

Table 4. Restricted 3-body problem (4.2)

Table 4 shows again very satisfactory results, except for the larger tolerance values 10^{-1} - 10^{-3}. For 10^{-1} and 10^{-2} r_{est} fails to indicate that the error estimate $est_n^{(2)}$ is inaccurate. In both cases, however, one can deduce from the magnitude of $est_n^{(2)}$ and $est_n^{(1)}$ that the results are unreliable.

It remains necessary to consider the magnitude of $est_n^{(2)}$ and $est_n^{(1)}$. Further we see that for 10^{-3} the estimate $est_n^{(2)}$ is very good, while r_{est} has nearly the same value as in the first two cases. We have already predicted this situation in section 3 where we established that r_{est}-values close to 1.4 may be meaningless.

4.3. A problem with a peaked solution.

Consider the initial value problem

$$\text{(4.3)} \qquad \dot{y} = -32xy\, \ell n\, 2, \; -1 \le x \le 1, \; y(-1) = 2^{-10},$$

with the peaked solution $y(x) = 2^{6-16x^2}$. We have taken this problem from LETHER [6]. For $x < 0$ the problem is unstable. Hence for $x < 0$ we will find global errors which increase with x due to unstable growth. On the other hand, for $x > 0$ the problem becomes highly stable for increasing x. Hence for $x > 0$ the errors should decrease again, as x increases.

We have solved this problem using pure relative local error control. For the tolerance 10^{-4} we have tabulated ε_n, r_{est}, r_{true} and ND for several values of $x_n \in [-1,+1]$ (see Table 5). For the remaining grid points similar r_{true} and r_{est}-values were found. Hence it can be concluded that the estimation procedure delivers a true copy of the global error behaviour over the complete integration interval.

	RGERK				GERK		
x_n	ε_n	r_{true}	r_{est}	ND	ε_n	r_{true}	ND
-.884	-7.6(-9)	.99	1.08	155	-5.3(-8)	.78	83
-.604	-2.0(-6)	.99	1.08	443	-1.4(-5)	.79	227
-.409	-2.2(-5)	.99	1.08	587	-1.5(-4)	.79	299
.078	-1.5(-4)	.98	1.04	767	-1.1(-3)	.90	389
.307	-5.1(-5)	1.00	1.11	844	-3.6(-4)	.70	430
.617	-2.7(-6)	1.00	1.07	1080	-2.0(-5)	.82	558
.817	-1.4(-7)	1.00	1.04	1296	-1.0(-6)	.90	666
1.000	-4.0(-9)	.99	1.02	1584	-3.0(-8)	.95	810

Table 5. Results for problem (4.3)

4.4. Mildly stiff problems

Though explicit Runge-Kutta codes cannot effectively be used for stiff problems, they may be of value for problems exhibiting a mild stiffness. However, when integrating a mathematically stable problem and numerical stability limits the stepsize, global Richardson extrapolation will always perform badly (see also SHAMPINE & WATTS [8]). More precisely, any estimate which makes use of results from the coarsest grid will be conservative. The reason is that the solutions on the finer grids are not troubled by numerical stability because the local error control, which now prevents the computation from becoming unstable, is performed on the coarsest grid. Often this implies that due to the stability of the problem and of the computation, the true global error at the finer grids is smooth and small when compared with the global error at the coarsest grid. This causes, fortunately enough, conservative estimates, but also large oscillations in r_{est}, as well as in r_{true}.

To see how our estimation procedure performs on a mildly stiff problem, we have shown in Table 6 some results for the simple problem

$$(4.4) \qquad \dot{y} = -100(y - \frac{x}{x+1}) + \frac{1}{(x+1)^2}, \quad x \geq 0, \qquad y(0) = 0.$$

The general solution is given by $y(x) = e^{-100x}y(0) + x/(x+1)$. Since we take $y(0) = 0$, only the rather smooth solution $x/(x+1)$ has to be computed. Table 6 contains results of the n-th step, where n = 10, 19, 30, 39, 50, from the integration under absolute local error control with the tolerance 10^{-3}. Recall that $\varepsilon_{n,1}$ denotes the error at the coarsest grid.

We observe that $\varepsilon_{n,1}$ oscillates and slightly increases with n, whereas ε_n smoothly decreases with n. This results in increasing and oscillating r_{true}-values. Note that r_{est} detects this behaviour in a satisfactory way. This is because $r_{est} = r_{true} * \varepsilon_n/est_n^{(1)}$ and $est_n^{(1)}$ is based on results from the second and third grid. Hence for mildly stiff problems $est_n^{(1)}$ is to be preferred above $est_n^{(2)}$, provided of course that the local error control has to prevent the numerical instability. If the code is applied with a maximal stepsize, chosen in such a way that absolute stability is taken care of, the estimation procedure will perform in a normal way.

n	x_n	$\varepsilon_{n,1}$	ε_n	$est_n^{(1)}/\varepsilon_n$	r_{true}	r_{est}
10	.347	-1.4(-4)	2.6(-7)	1.19	2.56	2.15
19	.679	2.6(-4)	1.4(-7)	1.20	-1.40	-1.17
30	1.084	-2.0(-4)	5.9(-8)	1.19	7.45	6.25
39	1.418	3.6(-4)	4.5(-8)	1.19	-11.37	-9.57
50	1.823	-3.4(-4)	3.0(-8)	1.19	20.57	17.33

Table 6. Results of RGERK for problem (4.4)

4.5. A problem with an oscillatory solution

Our next example is the linear problem [7]

$$(4.5)\qquad \begin{aligned} \dot{u}_1 &= \frac{1}{2}\frac{u_1}{x+1} - 2xu_2, & u_1(0) &= 1, & 0 &\le x \le 8, \\ \dot{u}_2 &= \frac{1}{2}\frac{u_2}{x+1} + 2xu_1, & u_2(0) &= 0, & 0 &\le x \le 8, \end{aligned}$$

with solution $u_1(x) = \sqrt{x+1}\,\cos(x^2)$, $u_2(x) = \sqrt{x+1}\,\sin(x^2)$. Observe that both components oscillate with increasing frequency as x increases. When solving this problem numerically the true global error appears to fluctuate rather strongly and takes on negative, as well as positive values. This lack of smoothness interferes with the error estimation, i.e. r_{est} also fluctuates. Here we even encountered negative values for r_{est} caused by a wrong sign in the first estimate $est_n^{(1)}$.

Problem (4.5) has been solved using absolute local error control for the tolerance 10^{-4}. Results are shown in Table 7. For RGERK we tabulated the percentages of the total number of times the pair (r_{true}, r_{est}) belongs to the regions indicated in the table. In the counting we considered all grid points and both components. In the second line from below we summed the percentages column-wise, while the last line from below in the table contains the corresponding r_{true}-percentages for GERK. All entries have been rounded to one decimal place. Hence an empty square does not necessarily mean that the score is exactly equal to zero.

r_{est}

−∞	1/4	1/2√2	1/2	1/√2	1.0	√2	2	2√2	4	+∞
+∞										
1.4					7.6					
1.3					1.1					
1.2					4.7	.4				
1.1					1.4	11.2				
1.0					.4	34.2				
.9						21.2				
.8						8.3				
.7						2.9				
.6						.7				
−∞	.7		.4	.7	1.8	2.2	.4			
	.7		.4	.7	17.0	81.1	.4			
	9.4	1.1	6.1	9.4	34.9	27.0	6.1	1.8	1.1	3.2
−∞	1/4	1/2√2	1/2	1/√2	1.0	√2	2	2√2	4	+∞

r_{true}

Table 7. Percentages for problem (4.5).

Inspection of Table 7 gives rise to several interesting conclusions. For RGERK almost all r_{true}-values are close to one; 98.1 percent lies between $1/\sqrt{2}$ and $\sqrt{2}$. For GERK this percentage is given by 61.9, showing that $est_n^{(2)}$, as expected, is more accurate than the estimate delivered by GERK. Further, r_{est} shows an appreciable degree of reliability in detecting the quality of the estimation. The percentage for the region $1/\sqrt{2} \le r_{true} \le \sqrt{2}$, $.6 \le r_{est} \le 1.3$ is given by 85.4.

4.6. Performance of RGERK on the test set of Hull et al.

To gain further insight in the use of computing a second and more accurate estimate $est_n^{(2)}$, we have applied RGERK to the five problem classes of [3]. For all 25 problems we used, componentwise, the local error criterion

$$|\text{estimated local error}| \leq \text{tolerance} *| \text{ solution } | + 10^{-14}$$

for the tolerances 10^{-3}, 10^{-5} and 10^{-7}. Hence, in normal cases, a relative error control. The results of this experiment are shown in Tables 8, 9 and 10 in exactly the same way as done in the preceding section for problem (4.5). However, here we calculated the percentages per problem and then averaged over the whole collection.

To show the influence of roundoff we have performed all integrations twice, in single precision and in double precision. Roundoff contaminates the estimates in cases where the errors and the estimates are extremely small. For example, for this reason Shampine and Watts exclude class C from their tables. Comparison of the results for single precision and double precision clearly shows that the majority of failures of the estimation procedure is caused by roundoff.

To facilitate the interpretation of Tables 8,9 and 10, we have collected total percentages for five interesting regions in (r_{true},r_{est})-space in Table 11. Region I $\cup$ II shows the number of times that $est_n^{(2)}$ approximates ε_n rather accurately. Region II shows the number of times that r_{est} should be considered as too conservative. Region III $\cup$ IV $\cup$ V shows the number of times that the accuracy of $est_n^{(2)}$ is not so good. Fortunately, in most of the cases this has been detected by r_{est}, according to the percentages of region III. Regions IV and V show the most interesting information, viz. the percentages of the number of times that r_{est} fails to indicate an inaccurate global error estimation. By way of illustration we show these percentages for two different ranges for r_{true}. When considering the failure percentages the reader should realize that we deal with a collection of 25 initial value problems which are divided into 5 different classes, each class having its own degree of difficulty. Again we note that the majority of failures is caused by roundoff.

r_{est}

−∞ 1/4 1/2√2 1/2 1/√2 √2 2 2√2 4 +∞

+∞										
1.4	4.3		.1	.7	5.8	7.2	2.3	2.2	.3	1.8
1.3	3.4	.1	.1	.4	.5	1.4	.2	.1	.1	.1
1.2	.1	.2			1.0	2.8	.5	.1	.1	.1
1.1	.2				2.7	6.4	.3	.1		
1.0	.3	.1	.2		2.8	7.3	.2	.1		
.9	1.0				7.1	4.4	.1	.1		
.8	.2			.3	7.4	1.4	.1	.3		
.7			.1	.1	2.2	1.9	.1			
.6	.2				1.2	.5	.1	.1		.1
−∞	7.6	.1	.6	1.8	1.2	1.4	.4	.2	.1	1.0
RGERK	17.3	.6	1.1	3.3	31.8	34.8	4.2	3.1	.5	3.2
GERK	32.3	1.6	4.0	7.1	12.5	19.7	7.8	4.6	5.0	5.4

r_{true}

+∞										
1.4	1.7		.1	.7	5.8	7.2	2.4	2.1	.2	1.9
1.3	1.8	.1	.1	.4	.4	1.5	.2	.1	.1	.1
1.2		.2			.9	2.8	.5	.1	.1	.1
1.1	.2				2.7	6.5	.2			
1.0					4.3	7.7	.2	.1		
.9	.4				10.6	4.4	.1	.1		
.8				.3	7.5	1.6	.1	.3		
.7			.1		2.2	2.2	.1			
.6	.2				1.1	.6	.1	.1		
−∞	7.5	.1	.6	1.7	1.3	1.5	.3	.2	.1	1.1
RGERK	11.8	.5	.8	3.3	37.0	35.8	4.1	3.1	.5	3.2
GERK	27.6	1.7	3.4	6.9	14.1	23.0	8.1	4.5	5.3	5.4

−∞ 1/4 1/2√2 1/2 1/√2 1.0 √2 2 2√2 4 +∞

r_{true}

Table 8. Results for tolerance 10^{-3}. Upper table contains single precision results. Lower table contains double precision results.

r_{est}

r_{true}	−∞ 1/4	1/4 1/2$\sqrt{2}$	1/2$\sqrt{2}$ 1/2	1/2 1/$\sqrt{2}$	1/$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$ 2	2 2$\sqrt{2}$	2$\sqrt{2}$ 4	4 +∞
+∞ 1.4	1.8			.3	1.1	1.9	.8			.4
1.4 1.3	1.6		.1	.1	.4	.6				
1.3 1.2	.1				1.0	1.8				
1.2 1.1	.1		.1		2.3	8.0	.1			
1.1 1.0	.4	.1	.1	.1	6.7	23.8				
1.0 .9	1.0			.1	17.7	8.1	.1	.1		
.9 .8	.2			.1	4.5	2.3				
.8 .7					1.7	.6	.1			
.7 .6					.9	.2	.1	.1		
.6 −∞	1.3	.1	.2	4.2	1.2	.4	.4	.2	.1	.3
RGERK	6.5	.2	.4	4.8	37.4	47.7	1.7	.4	.1	.8
GERK	10.2	.8	3.1	7.7	33.3	30.3	5.7	6.0	.9	2.0

r_{true}	−∞ 1/4	1/4 1/2$\sqrt{2}$	1/2$\sqrt{2}$ 1/2	1/2 1/$\sqrt{2}$	1/$\sqrt{2}$ 1.0	1.0 $\sqrt{2}$	$\sqrt{2}$ 2	2 2$\sqrt{2}$	2$\sqrt{2}$ 4	4 +∞
+∞ 1.4	.2			.3	1.1	1.6	1.1			.3
1.4 1.3	.4			.1	.4	.6				
1.3 1.2					.9	1.8				
1.2 1.1					2.3	8.1				
1.1 1.0					6.9	25.1				
1.0 .9	.2				21.3	7.7				
.9 .8					4.5	2.4				
.8 .7					1.6	.8	.1			
.7 .6					.9	.2	.1	.1		
.6 −∞	1.1	.1	.2	4.2	1.2	.5	.4	.2	.1	.4
RGERK	1.9	.1	.3	4.5	41.3	48.9	1.9	.3	.1	.7
GERK	6.5	.8	2.7	7.5	34.1	33.7	5.6	5.9	1.1	2.0

<u>Table 9</u>. Results for tolerance 10^{-5}. Upper table contains single precision results. Lower table contains double precision results.

r_{est}

	−∞	1/4	1/2√2	1/2	1/√2	√2	2	2√2	4	+∞
1.4	1.7				.4	.8				
1.3	2.5			.1	.2	.3				
1.2	.3				.4	.7				
1.1	.2				1.4	1.4				
1.0	1.0	.3	.2	.1	9.7	32.3	.1	.1		.1
.9	.8	.1	.2	.4	14.4	18.8	.4	.1	.1	.1
.8	.3			.1	1.4	1.1				
.7	.1			.1	4.4	.4				
.6	.1				.3	.2				
−∞	.6		.1	.1	.5	.3	.2			
RGERK	7.4	.4	.6	.9	33.0	56.3	.8	.3	.1	.2
GERK	6.9	.4	1.3	3.0	43.6	36.5	6.4	.7	.4	.9

→ r_{true}

1.4					.5	1.1				
1.3	.2				.3	1.2				
1.2					.5	.5				
1.1					1.4	1.6				
1.0					10.0	34.5				
.9	.1				27.1	10.7				
.8					1.4	1.4				
.7					4.7	.5				
.6					.3	.2				
−∞	.2			.3	.5	.3	.1			
RGERK	.6			.3	46.6	51.8	.2			.2
GERK	3.5	.3	.9	2.3	44.8	39.8	6.6	.7	.4	.8

−∞ 1/4 1/2√2 1/2 1/√2 1.0 √2 2 2√2 4 +∞

→ r_{true}

Table 10. Results for tolerance 10^{-7}. Upper table contains single precision results. Lower table contains double precision results.

		single precision			double precision		
		10^{-3}	10^{-5}	10^{-7}	10^{-3}	10^{-5}	10^{-7}
I	$1/\sqrt{2}\le r_{true}\le\sqrt{2}, .6\le r_{est}\le 1.3$	49.1	79.4	86.9	55.1	84.7	94.7
II	$1/\sqrt{2}\le r_{true}\le\sqrt{2}, r_{est}\le .6, r_{est}\ge 1.3$	17.4	5.8	2.4	17.7	5.6	3.7
III	$r_{true}\le 1/\sqrt{2}, r_{true}\ge\sqrt{2}, r_{est}\le .6, r_{est}\ge 1.3$	28.0	11.8	5.4	23.5	9.0	1.4
IV	$1/4\le r_{true}\le 1/\sqrt{2}, \sqrt{2}\le r_{true}\le 4, .6\le r_{est}\le 1.3$	3.4	1.2	2.4	2.8	.4	.1
V	$r_{true}\le 1/4, r_{true}\ge 4, .6\le r_{est}\le 1.3$	2.1	1.8	2.9	.9	.3	.1

Table 11. Percentages for five regions in (r_{true}, r_{est})-space.

The roundoff problem can be significantly diminished by using double precision. The additional runtime may be considerable, however. We found a factor of approximately 2.5. An alternative way to cope with the round-off problem is to distinguish between extremely small estimates and estimates of a more realistic magnitude. Table 12 contains single precision results for two intervals for $est_n^{(2)}$, viz. $|est_n^{(2)}| > 10^{-10}$ and $|est_n^{(2)}| \le 10^{-10}$. The percentages were taken over the corresponding subsets defined by these inequalities.

	$\lvert est_n^{(2)}\rvert > 10^{-10}$			$\lvert est_n^{(2)}\rvert \le 10^{-10}$		
Regions	10^{-3}	10^{-5}	10^{-7}	10^{-3}	10^{-5}	10^{-7}
I	46.4	84.2	96.9	57.3	68.3	75.6
II	22.2	6.5	2.5	3.1	3.8	2.3
III	27.1	8.7	.6	30.5	19.2	10.8
IV	3.7	.6		2.2	2.7	5.2
V	.6			6.9	6.0	6.1
totals	75.0	70.1	52.8	25.0	29.9	47.2

Table 12. Percentages for two intervals for $est_n^{(2)}$.

5. CONCLUSIONS

The percentages of Table 12 show that in case of extremely small estimates the reliability is insufficient. This cannot be avoided since the failures are due to roundoff effects. Fortunately, one is usually not interested in a very accurate estimate of extremely small errors, so these results are not as bad as they look.

The reliability is much larger if $est_n^{(2)}$ itself is not very small. In fact, for 10^{-7} the score for region V is exactly zero, while for region IV only a few failures out of approximately 20.000 data points were found. The score for region II is still too large, however. This is caused by inaccuracies in $est_n^{(1)}$. For 10^{-3} and 10^{-5} the reliability is less, as expected. In particular for 10^{-3}, a current tolerance value for a 5-th order code, the score for region I ($est_n^{(1)}$ and $est_n^{(2)}$ both accurate) is too low, whereas the score for II, IV, V is too high. Part of the failures for the larger tolerances is of course due to a failure of the asymptotics. However, numerical instability at the coarsest grid G_N (cf. section 4.4) also influences the results in a negative way.

Therefore we intend to continue our investigations with an estimation procedure which performs local error-stepsize control on the coarsest grid G_N and which performs global error estimation only on the finer grids $G_{2N},\ldots,G_{PN}$, $P \geq 3$. Herewith we avoid non-smoothness effects which might interfere with the estimation procedure. Such a procedure delivers P-2 estimates $est_n^{(i)}$, $i = 1,\ldots,P-2$, satisfying $est_n^{(i)} = \varepsilon_{n,P} + O(H^{p+i})$. The results of the present investigation show that a value $P > 3$ should be investigated. The cost ratio, in terms of f(y)-evaluations, is given by $(P+1)/2$. Hence, for a given accuracy, the additional computer time for the global estimation will be roughly a factor $(P+1)/2-1$ of the computer time required when no global error estimation is performed. In this respect it is worthwhile to observe that global Richardson extrapolation is uncommonly attractive for users who have a parallel computer at their disposal. Depending on the number of processors, the additional computer time can then be greatly reduced, even to zero (see e.g. JOUBERT & MAEDER [4]).

ACKNOWLEDGEMENT. The authors gratefully acknowledge the programming assistance of Mrs. M. Louter-Nool.

REFERENCES

[1] FEHLBERG, E., *Low-order classical Runge-Kutta formulas with step-size control and their application to some heat transfer problems.* NASA Tech. Rep. TR R-315, George C. Marshall Space Flight Center, Marshall, Ala.

[2] HENRICI, P., *Discrete Variable Methods for Ordinary Differential Equations.* Wiley, New York, 1962.

[3] HULL, T.E., ENRIGHT, W.H., FELLEN, B.M., & SEDGWICK, A.E., *Comparing numerical methods for ordinary differential equations.* SIAM J Numer. Anal. 9 (1972). 603-637.

[4] JOUBERT, G., & A. MAEDER, *Solution of differential equations with a simple parallel computer.* ISNM series Vol. 68, pp. 137-144.

[5] LAMBERT, J.D., *Computational methods in ordinary differential equations.* Wiley, New York, 1974.

[6] LETHER, F.G., *The use of Richardson extrapolation in one-step methods with variable step-size.* Math. Computation 20 (1966), 379-385.

[7] MERLUZZI, P. & BROSILOW, C., *Runge-Kutta integration algorithms with built-in estimates of the accumulated truncation error.* Computing 20 (1978) 1-16.

[8] SHAMPINE, L.F. & WATTS, H.A., *Global error estimation for ordinary differential equations.* ACM Transactions on Mathematical Software 2 (1976) 172-186.

[9] STETTER, H.J., *Analysis of discretization methods for ordinary differential equations.* Springer-Verlag, Berlin-Heidelberg-New York, 1973.

MC NR

35238