Computing 34, 179–190 (1985)

Computing
© by Springer-Verlag 1985

Short Communication / Kurze Mitteilung

Local Error Estimation by Doubling*

L. F. Shampine, Albuquerque

Received March 19, 1984; revised October 22, 1984

Abstract — Zusammenfassung

Local Error Estimation by Doubling. Doubling, or Richardson extrapolation, is a general principle for the estimation of the local error made by a one-step method for the numerical solution of the initial value problem for a system of ordinary differential equations. Some contributions are made to the theory of doubling. Principles of comparing explicit Runge-Kutta formulas are reviewed and illustrated in a balanced appraisal of doubling in the context of fourth order formulas. Some apparently contradictory comparisons in the literature are explained.

AMS Subject Classification: 65L05.

Key words: Doubling, Richardson extrapolation, error estimates.

Lokale Fehlerschätzung durch Schrittverdopplung. Schrittverdopplung, oder Richardson-Extrapolation, ist ein allgemeines Prinzip zur Schätzung des lokalen Fehlers eines Einschrittverfahrens zur numerischen Lösung eines Anfangswertproblems für Systeme gewöhnlicher Differentialgleichungen. Es werden einige Beiträge zur Theorie dieses Vorgehens gemacht. Die Prinzipien zum Vergleich expliziter Runge-Kutta-Formeln werden betrachtet und durch eine ausgewogene Bewertung des Verdoppelns bei Formeln vierter Ordnung illustriert. Einige scheinbar widersprüchliche Vergleiche aus der Literatur werden aufgeklärt.

1. Introduction

Doubling, or Richardson extrapolation, is a general principle for the estimation of the local error made by a one-step method for the numerical solution of the initial value problem for a system of ordinary differential equations. When the first codes for the initial value problem were written, doubling was the only way known to estimate the local error for Runge-Kutta formulas. Later pairs of embedded formulas were developed as an alternative, and some proved to be much more efficient in experiment. As a consequence doubling fell into disfavor.

According to tests of Enright and Hull [4], an embedded pair of formulas of orders four and five due to Fehlberg is much more efficient than the classical four stage,

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

fourth order formula with doubling. Tests reported by Deuflhard [1] say that the reverse is true, and according to Wanner [8], the two schemes behave much the same. Who is right? And why did these researchers come to such different conclusions?

We aim to provide a balanced appraisal of doubling in the context of fourth order, explicit Runge-Kutta methods, to make a couple of interesting and useful new observations about doubling, and to illustrate how to compare Runge-Kutta formulas: In the next section we provide a new, elementary proof of the doubling error estimator and use it to establish the existence of a higher order result at the middle of the step. We also observe that an explicit Runge-Kutta formula with error estimate by doubling actually forms an embedded pair. Whether this is a good pair depends on the competition available. The following section reviews some fundamentals of comparing Runge-Kutta formulas. In section four we compare various possibilities and explain some of the experimental results in the literature. Essential points are that how a formula is used and how efficiency is measured can be at least as important as the formula itself.

2. Doubling

The problem

$$y' = f(x, y), \quad a \le x \le b, \tag{2.1}$$

$$y(a) \quad \text{given} \tag{2.2}$$

is to be solved numerically. A one-step method successively produces approximations $y_n \doteq y(x_n)$ on a mesh $a = x_0 < x_1 < \dots$. Each step size $h = x_{n+1} - x_n$ is chosen automatically so as to satisfy a specified accuracy requirement, and the integration continues until b is reached or the task is judged impossible (or unreasonable).

An explicit Runge-Kutta formula for stepping from x_n to x_{n+1} has the form

$$\begin{aligned} f_0 &= f(x_n, y_n) \\ f_j &= f\left(x_n + \alpha_j h,\, y_n + h \sum_{k=0}^{j-1} \beta_{j,k} f_k\right) \quad j = 1, \dots, s \\ y_{n+1} &= y_n + h \sum_{j=0}^{s} c_j f_j. \end{aligned} \tag{2.3}$$

The local solution $u(x)$ at (x_n, y_n) is defined as that solution of (2.1) which satisfies

$$u(x_n) = y_n.$$

The local error of the formula is defined to be

$$y_{n+1} - u(x_{n+1}).$$

For a sufficiently smooth f, the formula is said to have the principal error function $\phi(x, y)$ and to be of order p if

$$y_{n+1} - u(x_{n+1}) = h^{p+1} \phi(x_n, y_n) + O(h^{p+2}). \tag{2.4}$$

One speaks of an embedded pair of formulas of orders p and $p+1$ if there is another linear combination of y_n and the f_j of (2.3) producing a result of order $p+1$, namely,

$$y_{n+1}^* = y_n + h \sum_{j=0}^{s} c_j^* f_j.$$

A computable estimate of the local error of y_{n+1} is provided by

$$\begin{aligned} y_{n+1} - y_{n+1}^* &= ((y_{n+1} - u(x_{n+1})) - (y_{n+1}^* - u(x_{n+1})) \\ &= y_{n+1} - u(x_{n+1}) + O(h^{p+2}). \end{aligned}$$

Such a pair provides a computationally convenient way to advance an integration one step and to estimate the associated local error.

The principle of doubling, or Richardson extrapolation, is a way of providing a local error estimate for any given formula (2.3), indeed, for any reasonable one-step method. A step of length h is taken from x_n to x_{n+1} to produce the result $\tilde{y}_{n+1}$. Two steps of length $h/2$ are taken from x_n to produce the result y_{n+1}. A suitable linear combination of the two results provides an estimate of the local error of y_{n+1}.

A key fact in the justification of doubling is that asymptotically the local error after two steps is the sum of the individual local errors. This is not often justified, and when it is, a technical proof based on Taylor expansions is given. The following elementary proof is more general and offers more insight.

Suppose that for a given α, a step is taken to $x_{n+\alpha} = x_n + \alpha h$ with a certain one-step method which yields $y_{n+\alpha}$. The local error, le_1, of this step is

$$\mathrm{le}_1 = y_{n+\alpha} - u(x_{n+\alpha}).$$

Suppose that for a given β, a step is then taken to $x_{n+\alpha+\beta} = x_{n+\alpha} + \beta h$ with a possibly different one-step method. The local solution $v(x)$ is that solution of (2.1) such that

$$v(x_{n+\alpha}) = y_{n+\alpha},$$

and the local error of the second step is, by definition,

$$\mathrm{le}_2 = y_{n+\alpha+\beta} - v(x_{n+\alpha+\beta}).$$

The local error of the two steps viewed as a single step of length $(\alpha+\beta)h$ from x_n is

$$\mathrm{le} = y_{n+\alpha+\beta} - u(x_{n+\alpha+\beta}).$$

The expression for le can be decomposed as

$$\begin{aligned} \mathrm{le} &= y_{n+\alpha+\beta} - v(x_{n+\alpha+\beta}) + v(x_{n+\alpha+\beta}) - u(x_{n+\alpha+\beta}) \\ &= \mathrm{le}_2 + \mathrm{le}_1 + [(v(x_{n+\alpha+\beta}) - u(x_{n+\alpha+\beta})) - \mathrm{le}_1]. \end{aligned}$$

We shall show that the expression in brackets is $O(h \, \| \mathrm{le}_1 \|)$. It is important to understand the circumstances. Here (x_n, y_n), α, β, and the formulas are fixed, and we consider what happens as $h \to 0$. Obviously $u(x)$ is fixed, but $v(x)$ is not. The fact that both $x_{n+\alpha}$ and $y_{n+\alpha}$ vary with h is what makes the result non-trivial.

It is presumed as usual that f satisfies a uniform Lipschitz condition:

$$\| f(x,w) - f(x,z) \| \le L \, \| w - z \| .$$

A standard bound then states that

$$\begin{aligned}\| v(x)-u(x)\| &\leq \exp\left(L(x-x_{n+\alpha})\right) \| v(x_{n+\alpha})-u(x_{n+\alpha})\| \\ &=\exp\left(L(x-x_{n+\alpha})\right) \| \mathrm{le}_1 \| \quad \text{for} \quad x \geq x_{n+\alpha}.\end{aligned} \tag{2.5}$$

From the integrated form of (2.1),

$$y(x)=y(x_{n+\alpha})+\int_{x_{n+\alpha}}^{x} f(t,y(t))\,dt,$$

we are led after some obvious manipulation to

$$\left(v(x_{n+\alpha+\beta})-u(x_{n+\alpha+\beta})\right)-\mathrm{le}_1=\int_{x_{n+\alpha}}^{x_{n+\alpha+\beta}} f(t,v(t))-f(t,u(t))\,dt.$$

Then on using the Lipschitz condition and (2.5), we have

$$\|\left(v(x_{n+\alpha+\beta})-u(x_{n+\alpha+\beta})\right)-\mathrm{le}_1\| \leq \| \mathrm{le}_1 \| \, L \int_{x_{n+\alpha}}^{x_{n+\alpha+\beta}} \exp\left((L(t-x_{n+\alpha}))\right) dt.$$

The right hand side here is $\| \mathrm{le}_1 \| \left(\exp(L\beta h)-1\right)$, which is $O(h \, \| \mathrm{le}_1 \|)$ as we aimed to show.

Returning now to doubling, we observe that

$$\tilde{y}_{n+1}-u(x_{n+1})=h^{p+1}\,\phi(x_n,y_n)+O(h^{p+2}).$$

The result just proved about the local error after two (half) steps says that

$$y_{n+1}-u(x_{n+1})=(h/2)^{p+1}\,\phi(x_n,y_n)+(h/2)^{p+1}\,\phi(x_{n+1/2},y_{n+1/2})+O(h^{p+2}).$$

However, assuming ϕ is differentiable,

$$\phi(x_{n+1/2},y_{n+1/2})=\phi(x_n,y_n)+O(h),$$

so that

$$y_{n+1}-u(x_{n+1})=2(h/2)^{p+1}\,\phi(x_n,y_n)+O(h^{p+2}). \tag{2.6}$$

It is then easy to see that

$$\frac{\tilde{y}_{n+1}-y_{n+1}}{2^p-1}=y_{n+1}-u(x_{n+1})+O(h^{p+2}). \tag{2.7}$$

If the local error is estimated by doubling, results of order p are available at both $x_{n+1/2}$ and x_{n+1} and quite a few codes provide both of them to users. It is well-known that a higher order result is available at x_{n+1}. Indeed, the estimate (2.7) makes it clear that

$$y^*_{n+1}=y_{n+1}-\frac{\tilde{y}_{n+1}-y_{n+1}}{2^p-1} \tag{2.8}$$

is a result of order $p+1$. Apparently it has not been observed before that a higher order result is also available at $x_{n+1/2}$. We saw first that the local error of y_{n+1} is asymptotically the sum of the errors made in the two half steps and then that these latter two errors are asymptotically equal. As a consequence

$$\mathrm{le}_1 = y_{n+1/2} - u(x_{n+1/2}) = \frac{1}{2}\,\mathrm{le} + O(h\,\|\,\mathrm{le}_1\,\|)$$
$$= \frac{1}{2}\left(\frac{\tilde{y}_{n+1} - y_{n+1}}{2^p - 1}\right) + O(h^{p+2}).$$

Consequently,

$$y^*_{n+1/2} = y_{n+1/2} - \frac{1}{2}\left(\frac{\tilde{y}_{n+1} - y_{n+1}}{2^p - 1}\right)$$

is of order $p+1$, too. Besides the obvious value of a second higher order result at each step, the information available is enough to allow a local interpolation [11] at low to moderate orders – a very valuable capability.

It is still common for authors to describe a formula with local error estimated by doubling as something quite distinct from a pair of embedded formulas. This is simply not so. When an explicit Runge-Kutta formula is the basic one, the result y^*_{n+1} of (2.8) is easily seen to be the result of an explicit Runge-Kutta formula and y_{n+1} is embedded in this formula. The local error estimate (2.7) is obviously $y_{n+1} - y^*_{n+1}$, the customary one for embedded formulas. The same is true of other kinds of one-step methods, too, but a little care is sometimes needed in the statement. For instance, Rosenbrock methods are semi-implicit, one-step methods which involve the Jacobian of f. In general the Jacobian might be evaluated at several places in a step, but considerations of work have led authors to concentrate on those formulas evaluating the Jacobian only at (x_n, y_n). In considering an embedded pair, one might well require this of both formulas. However, if doubling is used, there is an evaluation of the Jacobian at $(x_{n+1/2}, y_{n+1/2})$. Although the process can be viewed as an embedding, to do so requires one to allow Rosenbrock methods with more than one Jacobian evaluation per step.

In our view, doubling with explicit Runge-Kutta formulas is simply a general way to construct an embedded pair. The merits of the pair have to be compared to those of competing pairs individually. Doubling may be attractive because it may be difficult to derive a pair with good properties in any other way. As Deuflhard [1] points out, it is hard to construct even one good formula of high order, much less a pair. Kaps et al. [8] observe that with other kinds of one-step methods stability is crucial, and it may be extremely difficult to construct a pair such that both members have good stability. The computation of $\tilde{y}_{n+1}$ is being used solely for the estimate of the local error. One suspects that when the matter is not too complicated, one can estimate the error more cheaply. On the other hand, doubling provides two solution approximations in a step which at moderate orders can be used for interpolation. Later we shall consider some specific examples in some detail.

3. Comparing Formulas

In order to compare formulas it is necessary to know how they were implemented in two respects. Also it turns out that two quite different ways of measuring efficiency are employed in the literature. These issues are discussed in detail in [10]. Here we note the essentials.

The user of a code supplies a tolerance τ and specifies a norm in which error is to be measured. Codes based upon a criterion of error per step (EPS) select the step size h at each step so that

$$\| \text{local error} \| \leq \tau .$$

Codes based on error per unit step (EPUS) require

$$\| \text{local error} \| \leq h\tau .$$

This is accomplished by supposing that the leading term in the asymptotic expansion of the local error dominates as in (2.4) and approximating

$$\| \text{local error} \| \doteq \| y_{n+1} - y_{n+1}^* \| \doteq \| h^{p+1} \phi(x_n, y_n) \| . \tag{3.1}$$

The step size is selected so as to satisfy an accuracy requirement imposed on the formula of order p. However, a result y_{n+1}^* of order $p+1$ is available. Some codes advance the integration with the higher order result. This is called local extrapolation. Although some consideration has been given to deciding at each step whether to do local extrapolation [9], all the popular codes either never do it or do it all the time.

One way of measuring efficiency is to measure the cost, say by the number of evaluations of f, for a given tolerance. Another way is to measure the cost to achieve a given accuracy of solution. The first is exemplified by the tests of Hull et al. [7]. The second is exemplified by the tests of Shampine et al. [15]. It is, of course, possible to do both as Dormand and Prince [2] have done.

It is clear that for sufficiently small τ (hence h) EPUS selects a smaller step size than EPS. In the first measure of efficiency it is much less efficient; indeed, it "looks" as if a method of lower order were being used. The smaller step size results in more accuracy which is taken into account by the second measure of efficiency. In this measure the two criteria behave much the same.

Local extrapolation does not affect the choice of step size significantly because at each step

$$\phi(x_n, y_n) \doteq \phi(x_n, y_n^*).$$

This means that in the first measure of efficiency there is virtually no difference between doing local extrapolation and not. In the second measure of efficiency local extrapolation is much more efficient because of the higher order accuracy achieved.

Obviously it is essential to know which error criterion is used, whether local extrapolation is done, and the measure of efficiency in order to interpret properly numerical experiments.

Taylor series expansion of the local error of a formula of order p results in

$$y_{n+1} - u(x_{n+1}) = h^{p+1} \sum_{i=1}^{i_{p+1}} \tau_{p+1,i} D_{p+1,i} + h^{p+2} \sum_{i=1}^{i_{p+2}} \tau_{p+2,i} D_{p+2,i} + \dots .$$

Here the truncation error coefficients τ_{ki} depend only on the formula, and the derivatives D_{ki} depend only on f and (x_n, y_n). Expressions for the truncation error coefficients in terms of the coefficients defining the formula can be found in the literature, e.g., [2]. It is then easy enough to program their evaluation for the basic formula and its error estimating companion.

Because of the presence of the D_{ki}, the behavior of the local error of a Runge-Kutta method depends on the specific problem. This is an important reason why the traditional comparisons of formulas by solving a set of test problems such as [1, 4, 7, 15] are of limited value. There are other important reasons, too. Among them are defects in the test set such as problems which are "too" easy so that how the initial step size is chosen, where and how output is obtained, step failures, and details of the implementation are more important than the formula and such as problems which have a special form which is not representative and which may even be exceptionally good or bad for a given formula. It has become accepted that comparing the τ_{ki} of one formula to those of another provides valuable information about the relative merits of the formulas. This author would go farther: A thoughtful study of the truncation error coefficients provides more information and more reliable information about Runge-Kutta formulas than do battery tests.

In our comparison of formulas we shall compare the sizes of truncation error coefficients. The idea is that, say, if the $\tau_{p+1,i}$ are much smaller in magnitude for formula A than for a formula B of equal cost, then "usually" formula A is more accurate and by about the relative difference in size. Several measures of size are seen, all being norms of the coefficients of a given order, e.g.,

$$\| \tau_{p+1} \|_1 = \sum_{i=1}^{i_{p+1}} | \tau_{p+1,i} | .$$

In our work we have always looked at the three norms $\| \cdot \|_1$, $\| \cdot \|_2$, $\| \cdot \|_\infty$. As a rule, the coefficients are such that the relative behavior is about the same in the various norms so we display only one. If the formulas do not cost the same per step, we need to take this into account in a way which we illustrate in the next section.

4. Some Comparisons

We have emphasized that doubling is a way to produce a pair of formulas and that each such pair has to be compared with its competitors. Most of the literature concerned with doubling and explicit Runge-Kutta methods has focussed on the use of four stage, fourth order formulas. In this more restricted context we can assess the merits of doubling reasonably well and at the same time explain some of the apparently contradictory comparisons in the literature.

Doubling allows us to advance two half steps with any four stage, fourth order formula and then to estimate the local error by a whole step at a cost of three extra evaluations of f. It is reasonable to ask if an error estimating companion can be found which requires fewer extra evaluations. Shintani [16] showed that if a particular basic formula is used, then the error can be estimated with only one additional evaluation of f. England [3] has given a family of such procedures. A procedure of some popularity is based on the formula

$$\begin{aligned}
f_0 &= f(x_n, y_n) \\
f_1 &= f(x_n + h/2, y_n + h f_0/2) \\
f_2 &= f(x_n + h/2, y_n + h(f_0 + f_1)/4) \\
f_3 &= f(x_n + h, y_n + h(-f_1 + 2f_2)) \\
y_{n+1} &= y_n + h(f_0 + 4f_2 + f_3)/6.
\end{aligned} \tag{4.1}$$

We shall consider two schemes based on (4.1). Both use the fourth order result y_{n+1} generated from two half steps with (4.1). Measures of the truncation error coefficients of this result are given in Table 1 under the heading "England" with order $p=4$. The principal error function of this formula of eight stages is related to that of (4.1) as in (2.6). The procedure "ED" estimates the local error of this formula by the general principle of doubling as described in Section 2. The associated fifth order result y^*_{n+1} arises as in (2.8). Measures of its truncation error coefficients are given in Table 1 under the heading "England-Doubling". The pair costs 11 evaluations of f per step. England produces a different fifth order result y^*_{n+1} with only one extra stage so that his pair costs 9 evaluations of f per step. Measures of its truncation errors are also shown in Table 1.

Table 1. *Measures of the size of truncation error coefficients for some popular formulas given to two significant figures*

Formula	p	$\lVert\tau_{p+1}\rVert_1$	$\lVert\tau_{p+2}\rVert_1$	$\lVert\tau_{p+3}\rVert_1$
England	4	.0019	.0031	.0048
	5	0	.0015	.0037
doubling	5	0	.00093	.0024
Classical	4	.0022	.0035	.0054
Gill	4	.0019	.0031	.0048
Fehlberg	4	.0033	.018	.039
	5	0	.011	.028

Suppose that local extrapolation is not done. Then we have an easy comparison of England's scheme to the ED scheme. The two formulas produce the same result, but doubling costs 22% more (11 vs. 9 evaluations). The schemes differ only in their estimates of the local error. Examination of the truncation error coefficients suggests that the local error estimate provided by doubling is more accurate, but that both are satisfactory. Experimental work of Shampine and Watts [13] confirms this assessment.

The next question we address is whether the basic formula (4.1) is particularly unfavorable to doubling. Only a few of the four stage, fourth order formulas have been considered seriously. One due to Gill leads to measures of the truncation error coefficients which cannot be distinguished in Table 1 from those of the fourth order result of England. The "classical" formula leads to measures of truncation errors so close to those of England's fourth order formula that we expect no practical difference. There has been some effort devoted to finding the "best" four stage, fourth order formula for use with doubling error estimate [5]. It seems that the ED scheme can be taken as representative. In particular, the ED scheme should produce very much the same results as the classical formula with doubling – the CD scheme.

We have predicted that if local extrapolation is not done, then England's scheme is about 22% more efficient than the ED scheme and further that the ED scheme and CD scheme behave much the same. Shampine and Watts [12] present substantial experiments comparing England's scheme to the CD scheme. A criterion of EPS was used. The detailed results provided allow one to measure efficiency either way. Naturally, the results depend on the individual problem and tolerance, but over the whole set of tests England's scheme is about 22% more efficient than the CD scheme.

The embedded pair of England is not the most efficient. At present a pair due to Fehlberg is the most popular at orders four and five. A detailed comparison of the CD scheme, England's pair, Fehlberg's pair, and others has been given by Shampine and Watts [14]. The comparisons made so far in this section have been easy because the formulas used to advance the integration have been the same, or behaved nearly the same. When they differ, we must take into account which of EPUS or EPS is used and which measure of efficiency is chosen. To illustrate the comparison of pairs in general and to explain some numerical experiments in the literature, we shall compare the CD scheme to Fehlberg's pair. The experiments of Enright and Hull [4] used codes based on EPUS and the first measure of efficiency so we shall analyze this case. For later reference we note that their code implementing the Fehlberg pair is called RKF4 [6].

Given a tolerance τ, the step size h used by the classical formula with doubling satisfies

$$h\tau \doteq h^5 \, \| \phi \|,$$

so that

$$h \doteq (\tau / \| \phi \|)^{1/4}$$

when EPUS is the criterion. Similarly for the Fehlberg pair, the step size H is

$$H \doteq (\tau / \| \Psi \|)^{1/4}.$$

The relative cost per distance advanced is

$$\frac{11}{h} \Big/ \frac{6}{H} \doteq \frac{11}{6} \left(\frac{\| \phi \|}{\| \Psi \|} \right)^{1/4}$$

where we take into account that the one scheme costs 11 evaluations per step and the other 6. Of course, the principal error functions ϕ, Ψ depend on the problem, but a rough guide to their relative size is provided by the relative size of the truncation error coefficients. Thus the relative cost for a given tolerance τ is roughly

$$\frac{11}{6} \left(\frac{.0022}{.0033} \right)^{1/4} \doteq 1.66,$$

that is, the CD scheme is about 66% more expensive. At the most stringent tolerance, when testing effects are least important, the experiments of Enright and Hull [4] show a difference of 53%, which is in reasonable agreement with the prediction.

These tests received a great deal of attention and were important in directing scientists to more effective codes. Unfortunately they also contributed to a general repudiation of doubling which is not justified.

We have seen that the typical four stage, fourth order formulas combined with doubling are very inefficient when compared to one of the best embedded formulas like Fehlberg's. This is, however, so when local extrapolation is not done. The fact is that all of the currently popular Runge-Kutta codes do use local extrapolation (and EPS). One might hope that the extra work that doubling spends to get an error estimate be compensated by a better estimate and accordingly, a more accurate higher order formula. Although the Fehlberg pair has been outstandingly successful, its behavior at crude tolerances leaves something to be desired. Comparison of the size of the truncation error coefficients for the leading terms of the fourth and fifth order formulas shows that the step size must be rather smaller than for, say, the England pair to get a good estimate of the local error. Also, the step size must be rather smaller before the leading term in the expansion of the local error of the fourth order formula dominates, as is needed for the adjustment of the step size. Some experimental results which supports the claim that the Fehlberg pair is a little less reliable than the CD scheme or the England pair can be found in [4, 14]. We do not wish to overstate the case, but we do believe that local extrapolation is somewhat less satisfactory with the Fehlberg pair at crude tolerances.

The first measure of efficiency is blind to the effect on accuracy of local extrapolation. Thus the conclusions drawn earlier that the Fehlberg pair is much more efficient than the CD or ED schemes remains true when local extrapolation is done. The second measure of efficiency reveals the effect of local extrapolation. To illustrate how conclusions drawn earlier are altered in this measure, we shall compare the Fehlberg pair to the ED scheme when local extrapolation is done. Now we suppose that τ is the accuracy *achieved*. Arguing as before with EPUS, we find that the relative efficiency is

$$\frac{11}{h}\bigg/\frac{6}{H} \doteq \frac{11}{6}\left(\frac{\|\phi\|}{\|\Psi\|}\right)^{1/5} \doteq \frac{11}{6}\left(\frac{.00093}{.011}\right)^{1/5} \doteq 1.12.$$

In this case the principal error functions ϕ, Ψ are those of the fifth order formulas. Now the formula based on doubling is only 12% more expensive. This is quite a different conclusion than the one reached when local extrapolation was not done. Doubling is not at all a bad idea in these circumstances.

When we did the numerical comparisons [15], we emphasized that we were comparing *codes*. Conclusions about the formulas on which the codes are based can be drawn only when the greatest care is exercised; the implementation and the way the results are evaluated may completely obscure the effect of the formula. A case in point are the computations of Deuflhard [1] which seem to say that the classical formula with doubling is much more efficient than the Fehlberg pair. A clue is furnished by the fact that the codes do not appear to have the same order. The RKF4 code used is based on EPUS and does not do local extrapolation. It is clear from the paper that the doubling code written by Deuflhard is based on EPS and does do local extrapolation. The first measure of efficiency was used. In this measure the difference between the two codes due to local extrapolation is innocuous. However, EPUS is much less efficient than EPS; as we have remarked, the effect is somewhat like using a lower order formula. The difference in the criteria is strong enough to reverse the difference in efficiency of the formulas. Interestingly, if Deuflhard had chosen the second measure of efficiency, he would have still found doubling the more efficient. In this measure the difference in criterion would be innocuous, but local extrapolation would cause the doubling code to be of higher order.

We quote from the paper of Kaps et al. [8], "Secondly, preliminary calculations of G. Wanner [31] showed that for explicit methods there is little difference between the two techniques. (Wanner compared Fehlberg's embedded order 4(5) method RKF4 (using order 4 for step continuation and order 5 for error estimate) and the classical 4th order Runge-Kutta formula using the extrapolated value for step continuation)." This oral communication to Kaps does not provide enough detail that we can fully explain the conclusion that there is not much difference. Although the Fehlberg pair is more efficient than the CD scheme, the integration is advanced at order 4 in RKF4 and at order 5 in the code based on the CD scheme. This difference in implementation alone is enough to explain why the CD scheme can complete on equal terms.

5. Conclusions

A new, elementary proof of the doubling error estimate was given and it was shown how to obtain a higher order result at the midpoint of each step. It was emphasized that for explicit Runge-Kutta methods, doubling is not conceptually different from embedded pairs; it is a general way to construct an embedded pair. The merits of a pair constructed in this way must be compared individually to those of competing pairs.

Principles of comparing Runge-Kutta formulas were reviewed and applied to various fourth order formulas with doubling estimate and to the Fehlberg pair. Some apparently contradictory experimental results in the literature were explained. It is not enough to say that a code implements a particular formula. At a minimum, it is necessary to indicate whether the criterion of error per step or error per unit step was used, whether local extrapolation was done, and whether efficiency was measured in the sense of cost for given tolerance or cost for given accuracy achieved.

Acknowledgments

Conversations and correspondence with P. Kaps brought the question investigated here to the author's attention and got him interested in explaining some of the experimental work reported. L. S. Baca provided the author with valuable computational support.

References

[1] Deuflhard, P.: Recent progress in extrapolation methods for ordinary differential equations. Rept. 224, SFB 123, University of Heidelberg, Federal Republic of Germany, July 1983.
[2] Dormand, J. R., Prince, P. J.: A family of embedded Runge-Kutta formulae. J. Comp. Appl. Math. *6,* 19–26 (1980).
[3] England, R.: Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations. Computer J. *12,* 166–170 (1969).
[4] Enright, W. H., Hull, T. E.: Test results on initial value methods for non-stiff ordinary differential equations. SIAM J. Numer. Anal. *13,* 944–961 (1976).
[5] Hull, T. E., Johnston, R. L.: Optimum Runge-Kutta methods. Math. Comp. *18,* 306–310 (1964).
[6] Hull, T. E., Enright, W. H.: A structure for programs that solve ordinary differential equations. Rept. 66, Dept. Comp. Sci., University of Toronto, Canada, 1974.
[7] Hull, T. E., Enright, W. H., Fellen, B. M., Sedgwick, A. E.: Comparing numerical methods for ordinary differential equations. SIAM J. Numer. Anal. *9,* 603–637 (1972).
[8] Kaps, P., Poon, S., Bui, T. D.: Rosenbrock methods for stiff ODEs: A comparison of Richardson extrapolation and the embedding technique. Institute for Mathematics I, Univ. of Innsbruck, Austria, 1983.
[9] Shampine, L. F.: Local extrapolation in the solution of ordinary differential equations. Math. Comp. *27,* 91–97 (1973).
[10] Shampine, L. F.: The step sizes used by one-step codes for ODEs. Appl. Num. Math. *1,* 95–106 (1985).
[11] Shampine, L. F.: Interpolation for Runge-Kutta methods. SIAM J. Numer. Anal. (to appear).
[12] Shampine, L. F., Watts, H. A.: Efficient Runge-Kutta codes. Rept. SC-RR-70-615, Sandia National Laboratories, Albuquerque, N.M., September 1970.
[13] Shampine, L. F., Watts, H. A.: Comparing error estimators for Runge-Kutta methods. Math. Comp. *25,* 445–455 (1971).
[14] Shampine, L. F., Watts, H. A.: Practical solution of ordinary differential equations by Runge-Kutta methods. Rept. SAND76-0585, Sandia National Laboratories, Albuquerque, N.M., December 1976.
[15] 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).
[16] Shintani, H.: Two-step processes by one-step methods of order 3 and of order 4, J. Sci. Hiroshima Univ. *A-130,* 183–195 (1966).

L. F. Shampine
Numerical Mathematics Division
Sandia National Laboratories
Albuquerque, NM 87185, U.S.A.

Verleger: Springer-Verlag KG, Mölkerbastei 5, A-1010 Wien. — Herausgeber: Prof. Dr. Hans J. Stetter, Institut für Angewandte und Numerische Mathematik der Technischen Universität Wien, Wiedner Hauptstraße 6—10, A-1040 Wien. — Redaktion: Wiedner Hauptstraße 6—10, A-1040 Wien. — Hersteller: Satz Austro-Filmsatz Richard Gerin, Zirkusgasse 13, A-1020 Wien, Druck Paul Gerin, Zirkusgasse 13, A-1021 Wien. — Verlagsort: Wien. — Herstellungsort: Wien. — Printed in Austria.