MATHEMATICS OF COMPUTATION
VOLUME 46, NUMBER 173
JANUARY 1986, PAGES 135-150

Some Practical Runge-Kutta Formulas*

By Lawrence F. Shampine

Abstract. A new selection is made of the most practical of the many explicit Runge-Kutta formulas of order 4 which have been proposed. A new formula is considered, formulas are modified to improve their quality and efficiency in agreement with improved understanding of the issues, and formulas are derived which permit interpolation. It is possible to do a lot better than the pair of Fehlberg currently regarded as "best".

1. Introduction. In [19] the author and H. A. Watts compared many pairs of explicit Runge-Kutta formulas with the goal of selecting the most suitable for an effective code. A pair of formulas of orders four and five due to Fehlberg [4] were chosen for implementation. Codes based on this pair showed up very well in extensive numerical comparisons [2], [11], [21], so the pair is now regarded as the method of choice at orders $(4, 5)$.

It is time to reconsider the choice made in [19]. For one thing, Dormand and Prince [1] subsequently derived a very efficient pair which needs to be considered. For another, certain issues of quality and efficiency are better understood now, and there has been an accumulation of evidence that the Fehlberg pair, though very good, is not all one might hope for. Most important of the reasons for a reconsideration is the recent progress [9], [15] made in providing "interpolation" for Runge-Kutta methods. In the author's opinion, this capability and the features it makes possible will be the hallmark of the next generation of Runge-Kutta codes.

In Section 2 we compare the leading possibilities with respect to efficiency, quality, and stability. We propose modifications to existing pairs to improve their behavior according to one or another of these criteria. Section 3 is devoted to "interpolation." Schemes of orders 4 and 5 are derived for the Dormand-Prince pair which are counterparts of formulas derived by Horn [9] for the Fehlberg pair. After a brief summary of the main points of comparison in Section 4, the recommended formulas are collected for the convenience of the implementor.

2. Accuracy, Quality, and Stability. In [19] we considered a great many explicit Runge-Kutta formulas as candidates for the basis of an effective code. Three pairs of formulas were selected as the main contenders of order four. The pair chosen for implementation in RKF45 [19] (and its successor DERKF [20]) is due to Fehlberg. This pair is that given by Fehlberg in [4], but it is the second of two pairs he gave in the original report [3]. The first of these pairs was also one of the three contenders. A pair due to Shintani [22] was the third main possibility.

Received May 18, 1984; revised December 10, 1984.

1980 *Mathematics Subject Classification*. Primary 65L05.

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

The RKF45 code has been extremely successful, appearing, for example, in a major software library and three textbooks. It performed very well in extensive comparisons with other codes [21]. Hull and Enright [10] wrote a research code, RKF4, based on this pair which also showed up well in extensive comparisons [2]. As a consequence, the Fehlberg #2 pair, or for short, the Fehlberg pair, is considered a standard for judging new formulas at this order.

The present investigation began when the author realized how to modify the Fehlberg #1 pair to make it more efficient and more reliable than the Fehlberg #2 pair. The same idea improves the quality of the Shintani pair. After the study [19], Dormand and Prince [1] presented a pair of formulas they call RK5(4)7M which is more efficient than our modification of the Fehlberg #1 pair. It is, however, desirable to modify RK5(4)7M a little, too.

Recent investigations [9], [15] have shown how to "interpolate" with the Fehlberg and Shintani pairs. In the next section we show how to accomplish this with the Dormand-Prince pair. Because of their great efficiency and because interpolation is practical with them, we shall study here only these three pairs and their variants.

The initial value problem

$$y' = f(x, y), \qquad y(a) \text{ given},$$

is to be solved by an explicit Runge-Kutta formula. Such a formula advances from an approximation y_n of the solution at x_n one step of length h to an approximation y_{n+1} of the solution at $x_{n+1} = x_n + h$ by a recipe of the form

$$\begin{aligned} f_0 &= f(x_n, y_n), \\ f_j &= f\left(x_n + \alpha_j h,\ y_n + h \sum_{k=0}^{j-1} \beta_{j,k} f_k\right), \qquad j = 1, \dots, s, \\ y_{n+1} &= y_n + h \sum_{j=0}^{s} c_j f_j. \end{aligned} \tag{2.1}$$

Here, the constants α_j, $\beta_{j,k}$, c_j define the formula.

The local solution at (x_n, y_n), $u(x)$, is defined by

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

For sufficiently smooth f, a Taylor expansion about (x_n, y_n) leads to expressions for the local error: The first defines the principal error function ϕ,

$$u(x_n + h) - y_{n+1} = h^{p+1}\phi + O(h^{p+2}), \tag{2.2}$$

and the second provides more detail,

$$\begin{aligned} u(x_n + h) - y_{n+1} &= h^{p+1} \sum_{j=1}^{\lambda_{p+1}} T_{p+1,j} D_{p+1,j} \\ &\quad + h^{p+2} \sum_{j=1}^{\lambda_{p+2}} T_{p+2,j} D_{p+2,j} + O(h^{p+3}). \end{aligned} \tag{2.3}$$

Accordingly, formula (2.1) for y_{n+1} is said to be of order p. Here the elementary differentials $D_{p+i,j}$ are functions only of f and (x_n, y_n), so depend only on the problem. The truncation error coefficients $T_{p+i,j}$ depend only on the formula. An introduction to such expansions is provided by Chapter 10 of [6]. The statement that a method is of order p is expressed by the equations of condition, $T_{q,j} = 0$ for $j = 1, \ldots, \lambda_q$, $0 \leqslant q \leqslant p$. Expressions for the truncation error coefficients $T_{q,j}$ in terms of the coefficients defining the formula can be found, e.g., in Chapter 10 of [6] and in [1].

It is presumed that in addition to (2.1), there is available another formula using the same f_j, namely

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

which is a formula of order $p + 1$. Then

$$\begin{aligned} y_{n+1}^E - y_{n+1} &= \left(u(x_n + h) - y_{n+1}\right) - \left(u(x_n + h) - y_{n+1}^E\right) \\ &= \left(u(x_n + h) - y_{n+1}\right) + O(h^{p+2}) \end{aligned}$$

furnishes a computable estimate of the local error of y_{n+1}.

The user specifies a norm $\|\cdot\|$ and an error tolerance τ. At each step, codes attempt to select h so as to satisfy a local error requirement. Two possibilities are seen. Error per step (EPS) requires

$$\|\text{local error}\| \leqslant \tau$$

and error per unit step (EPUS) requires

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

The formulas are developed on the assumption that the integration is advanced with the approximation y_{n+1}. Another possibility is to continue the integration with the higher-order result y_{n+1}^E. This is called local extrapolation. All four possibilities are seen in practice, but the most popular codes today, DO2PAF [5], RKF45, and DVERK [12], all use EPS and local extrapolation. For this reason we concentrate on this case. Which choices are made have important implications that are discussed in [16].

The Fehlberg pair needs only the minimal number of evaluations of f, or "stages," to achieve order 5, namely 6. The Shintani pair requires 7. Dormand and Prince exploit an idea used earlier by Fehlberg [4] and others. They derive a fifth-order result y_{n+1}^E and then add the stage $f(x_{n+1}, y_{n+1}^E)$ for the computation of a fourth-order result y_{n+1}. If the step succeeds and if local extrapolation is done, this stage is the same as the first stage of the next step. We do not count it as a seventh stage in the current step, but rather we say it is "free" because it is just an early formation of a stage counted in the next step. It *is* an extra stage when the current step is rejected, but step-size selection algorithms are effective enough to make rejected steps unusual, so it is fair to say that this pair costs nearly the same as Fehlberg's. If local extrapolation is not done, the stage is not used in the next step, and then the pair costs 7 evaluations per step.

Traditionally, formulas are compared by solving test problems. The expansion (2.3) of the local error makes it clear that the relative performance of two formulas depends on the problem considered. For this reason, one must solve a lot of problems numerically to determine which formula is "usually" better. Another way to compare formulas of the same order is by examination of their truncation errors.

Schemes for the automatic selection of the step size depend on the dominance of the leading term of the local error expansion (2.3), so that some attention is given in the codes to making this so. Accordingly, if the leading term in (2.3) for one formula is rather smaller than that of another formula of the same order and cost, the first formula will normally be more efficient, and in about the same proportion. Because the elementary differentials $D_{p+1,j}$ depend on the problem, we cannot compare these terms directly. Still, if we ask about the "typical" behavior over a large ensemble of problems, we might reasonably hope that the relative sizes of the coefficients of the elementary differentials, i.e., the truncation error coefficients, will tell us how the methods "usually" compare. One measure of the "size" of the coefficients used for this purpose is

$$\|T_{p+1}\|_2 = \left(\sum_{j=1}^{\lambda_{p+1}} T_{p+1,j}^2 \right)^{1/2}.$$

In some respects the two ways of comparing formulas are complementary. When they can both be applied, e.g., to some aspects of efficiency, the author's experience has been that a thoughtful comparison of truncation error coefficients provides more detail and more reliable conclusions than does the study of a battery of numerical tests as is done in [2], [11], [21].

We define efficiency for a formula as the distance advanced in a single step divided by the cost of the step. A conventional and useful way to measure "cost" here is to count the number of function evaluations made. When we speak of the "first measure of efficiency" we mean that the step size used is the optimal one for a given tolerance τ. When we speak of the "second measure of efficiency" we mean that the step size used is the optimal one for achieving a given accuracy ε. Experimentally, the first measure corresponds to counting the number of function evaluations needed to solve a problem when a tolerance τ is given the code. In contrast, the second measure relates the cost to the accuracy ε of the approximate solutions returned by the code. A more complete discussion of these issues can be found in [16]. For now, we discuss only the first measure of efficiency. In this measure, we must consider the two possible implementations of EPUS and EPS. For a given τ and EPUS the optimal step size h_A satisfies

$$h_A \tau \doteq \|h_A^{p+1} \phi_A\| \tag{2.4}$$

for a principal error function ϕ_A (from (2.2)) depending on the formula A and f at the current point (x_n, y_n). If the cost of a step is C_A evaluations of f, the efficiency is about

$$\frac{h_A}{C_A} \doteq \frac{1}{C_A} \left(\frac{\tau}{\|\phi_A\|} \right)^{1/p}.$$

When compared to a formula B, the relative efficiency is about

$$\left(\frac{h_A}{C_A} \right) \Big/ \left(\frac{h_B}{C_B} \right) \doteq \frac{C_B}{C_A} \left(\frac{\|\phi_B\|}{\|\phi_A\|} \right)^{1/p}. \tag{2.5}$$

As we discussed earlier, we shall use the ratio $\|T_{p+1}^B\|/\|T_{p+1}^A\|$ as a way of assessing the size of $\|\phi_B\|/\|\phi_A\|$ for a "typical" problem in a large ensemble of problems. This then provides a computable measure of efficiency.

TABLE 1

Measures of the size of truncation error coefficients of some good Runge-Kutta formulas

Formula	order p	$\|T_5\|_2$	$\|T_6\|_2$	$\|T_7\|_2$
Fehlberg	5	0	.0034	.0068
	4	.0018	.0058	.0094
Shintani	5	0	.0011	.0018
	4	.00040	.0014	.0021
modified	4	.0016	.0026	.0032
Dormand-Prince	5	0	.00040	.0040
	4	.0012	.0018	.0041
modified	4	.00079	.0012	.0039

If we compare the efficiency of the Dormand-Prince pair to the Fehlberg pair, reference to Table 1 leads to

$$\frac{6}{6}\left(\frac{.0018}{.0012}\right)^{1/4} \doteq 1.11.$$

This says that the Dormand-Prince pair is about 11% more efficient. Dormand and Prince [1] present results of computations on a standard set of test problems for pure absolute error tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-9}$. The total costs were

	evaluations of f	successful steps
Fehlberg	226208	36363
Dormand-Prince	209305	33126

The observed relative efficiency is

$$\frac{226208}{209305} \doteq 1.08.$$

The agreement looks too good to be true. It is. This is made clear by computing the average cost of a successful step to be about 6.22 in the one case and 6.32 in the other. A successful step costs 6 evaluations, so it is clear that failed steps played an important role. A number of problems in this test set are so easy that even for the comparatively stringent tolerances specified, how good the initial step size is, how failed steps are handled, and output play important roles. The fact that Dormand and Prince implemented the formulas in identical fashion does minimize the variation usually seen when comparing codes.

The efficiency of the Shintani pair relative to the Fehlberg pair is

$$\frac{6}{7}\left(\frac{.0018}{.00040}\right)^{1/4} \doteq 1.25$$

which is much more efficient. As we have noted, EPS is preferred in practice. Because $1/p$ is replaced by $1/(p+1)$ in (2.5), the differences are less marked with EPS. The efficiencies for the Dormand-Prince and Shintani pairs compared to the Fehlberg pair are then 1.08 and 1.16, respectively.

Given a pair (y_{n+1}, y^E_{n+1}) of embedded formulas of orders 4 and 5, respectively, the result

$$y'_{n+1} = \alpha y_{n+1} + (1 - \alpha) y^E_{n+1} \tag{2.6}$$

is another embedded formula of order 4 for any $\alpha \neq 0$. It is easily seen that

$$\|T'_5\| = |\alpha| \, \|T_5\|,$$

so that we can form a formula that is as accurate as we like. How accurate should we make it? To answer this, we need to consider the quality of a pair of formulas.

Algorithms for the adjustment of the step size assume that the leading term in the expansion (2.3) of the local error dominates. Except in very special circumstances, this is true when h is small enough, but how small is "enough"? Arguing as before, the smaller $\|T_6\|$ is compared to $\|T_5\|$, the better the behavior for "large" h (equivalently, crude tolerances τ). If we compare the estimated local error to its true value, we have

$$\begin{aligned}(u(x_n + h) - y_{n+1}) - (y^E_{n+1} - y_{n+1}) &= u(x_n + h) - y^E_{n+1} \\ &= h^6 \sum_{j=1}^{\lambda_6} T^E_{6,j} D_{6,j} + O(h^7).\end{aligned}$$

Although we are supposing that the leading term of the expansion (2.3) of the local error dominates, we consider here how well *all* the terms are estimated. The size of the local error is indicated by $\|T_5\|$, and we see now that the error made by its estimator is indicated by $\|T_6^E\|$. Thus, the smaller $\|T_6^E\|$ is, compared to $\|T_5\|$, the better the local error estimator. It is not so important to have a very accurate estimate, but a reasonable accuracy which can be relied upon even at crude tolerances *is* important. The matter is more serious when local extrapolation is done; then we really do need a y^E_{n+1} which is more accurate than y_{n+1}.

Doubling, or Richardson extrapolation, is considered to provide a superior error estimate when applied to one of the four-stage, fourth-order formulas. When written as a pair of formulas, it is found that $\|T_6\|$ is no more than about 1.5 times $\|T_5\|$ [17] and that $\|T_6^E\|$ is about half $\|T_5\|$. The quality is demonstrated by computations in [18] which also support the claim that a good error estimate is provided by a pair of England. This pair has $\|T_6\|$ about 1.5 times $\|T_5\|$ and $\|T_6^E\|$ a little smaller than $\|T_5\|$. Other pairs from [18], [19] with good error estimates show similar ratios. This gives us an idea as to appropriate relative sizes of truncation errors for good quality formulas.

According to the measures of the truncation errors given in Table 1, the Fehlberg pair is of slightly better quality than the Shintani pair but much worse than the Dormand-Prince pair. By the standards of the good pairs cited, the Fehlberg pair does not have the quality one might hope for.

We have modified the fourth-order Shintani formula by forming the linear combination (2.6) with $\alpha = 4$. The quality of the modified pair is comparable to that of England's and others which showed up well in [18], [19]. Unfortunately, the new pair is less efficient than the Fehlberg pair in the first measure of efficiency. With

EPS the relative efficiency is

$$\frac{6}{7}\left(\frac{.0018}{.0016}\right)^{1/5} \doteq 0.88.$$

A situation opposite that of the Shintani pair occurs with the Dormand-Prince pair. The latter is unnecessarily conservative. We have chosen to modify the fourth-order formula as in (2.6) with $\alpha = 2/3$. This improves its efficiency relative to the Fehlberg pair in the first measure with EPS to 1.18, which is quite significant, while still meeting the standard of quality proposed. From the sizes of the truncation error coefficients, we are led to expect that the local error will be estimated much more accurately with this pair than with the Fehlberg pair. We also expect that the local error estimate will be much more reliable at crude tolerances, hence that local extrapolation is better justified. This was our intention because the structure of the formula presupposes local extrapolation. It is worth remarking that even if one does not do local extrapolation, the new formula is as accurate as Fehlberg's. The relative efficiency is then

$$\frac{6}{7}\left(\frac{.0018}{.00079}\right)^{1/5} \doteq 1.01.$$

The first measure of efficiency is blind to the effect of local extrapolation [16]. This is because the selection of the step size is based on the lower-order formula and no attention is paid to the extra accuracy achieved by local extrapolation. We turn now to the second measure of efficiency which is concerned with the accuracy achieved. For a given accuracy ε achieved, one has

$$\varepsilon \doteq \left\| h^{p+2} \phi^E \right\|,$$

where the principal error function ϕ^E is that of the higher-order formula of the pair. Arguing as before, the relative efficiency in the second measure is

$$\frac{C_B}{C_A}\left(\frac{\|T_B^E\|}{\|T_A^E\|}\right)^{1/(p+2)}.$$

Comparing the Shintani pair to the Fehlberg pair in this way leads to

$$\frac{6}{7}\left(\frac{.0034}{.0011}\right)^{1/6} \doteq 1.03,$$

and comparing the Dormand-Prince pair to Fehlberg's pair leads to 1.43. Dormand and Prince present several plots comparing efficiency in this second measure. They note that for one problem an absolute accuracy of 10^{-6} was achieved in 1450 evaluations of f with the Fehlberg pair, and in 800 with their pair. This relative efficiency of 1.81 is in reasonable agreement with our prediction of 1.43 when one keeps in mind that the observed efficiency is for a single problem.

Figure 1 shows the stability regions of the three fifth-order formulas scaled for equal work. Figure 2 shows the fourth-order Fehlberg formula and the modified Shintani and Dormand-Prince fourth-order formulas. Because the formulas do not differ a great deal with respect to stability, this issue is not important to the selection of the best pair.

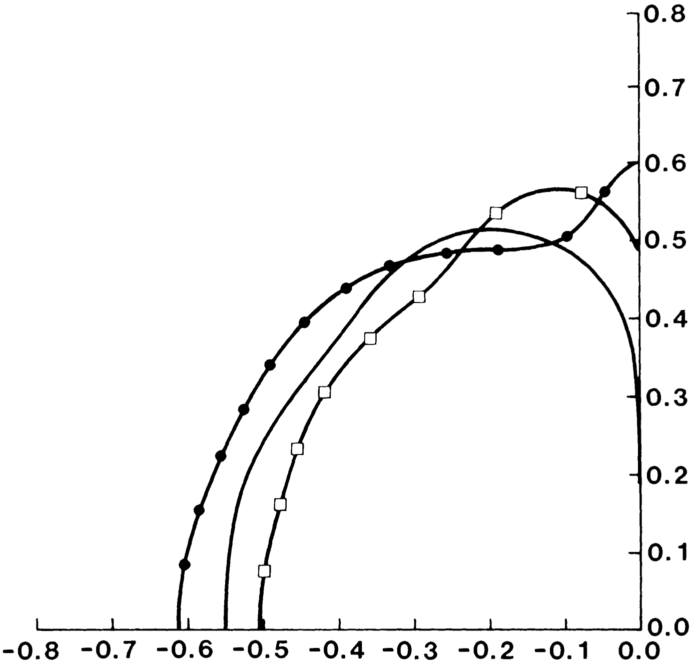

FIGURE 1

Stability regions of the fifth-order formulas considered. ●—●—● *Fehlberg* #2, — *Dormand-Prince*, □—□—□ *Shintani*.

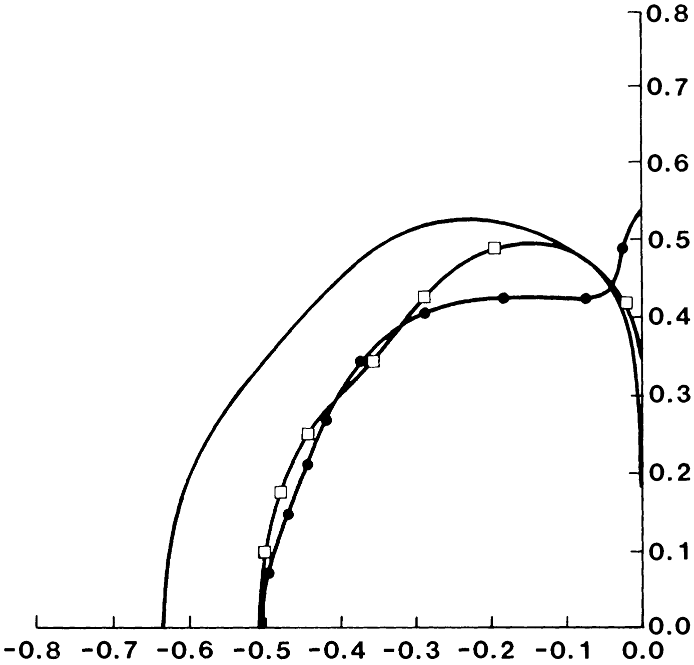

FIGURE 2

Stability regions of the fourth-order formulas considered. ●—●—● *Fehlberg* #2, — *modified Dormand-Prince*, □—□—□ *modified Shintani*.

3. Interpolation. Traditionally, one of the strongest advantages of Adams methods in comparison to Runge-Kutta methods has been the capability of producing approximate solutions at any x, not just mesh points, by interpolation. This allows the step size to be chosen more or less independently of output with the consequence that the integration can be more efficient. The capability has other implications. One is that the error behaves more regularly when the step size is not altered for output. For some tasks, such as finding where an associated algebraic equation has a root, the capability is nearly indispensable. The next generation of Runge-Kutta codes will have the capability. Interestingly, its theoretical justification is far better than that of the procedures in the Adams codes.

One approach to interpolation has been developed and applied to the Fehlberg pair by K. Horn [7], [8], [9]. The idea is an extension of the idea of an embedded pair. In addition to the usual step from (x_n, y_n) of length h as in (2.1), we independently advance the integration a step of length $h^* = \sigma h$ by another formula according to

$$f_0 = f(x_n, y_n),$$

$$f_j^* = f\left(x_n + \alpha_j^* h^*,\ y_n + h^* \sum_{k=0}^{j-1} \beta_{j,k}^* f_k^*\right), \qquad j = 1, \dots, s^*, \tag{3.1}$$

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

If we take

$$\alpha_j^* = \alpha_j/\sigma, \qquad j = 1, \dots, s,$$
$$\beta_{j,k}^* = \beta_{j,k}/\sigma, \qquad 0 \leqslant k < j \leqslant s,$$

then $f_j^* = f_j$ for $j = 1, \dots, s$. It may be necessary to add stages $(s^* > s)$ to achieve a desired order. Notice that a new formula is derived for each σ of interest. The code is attempting to control the error of the fourth-order formula in stepping to $x_n + h$. It is appropriate, then, to compare the error of this new formula to that of the basic fourth-order formula. A little care is needed. The local error expansion for the new formula has the same form as (2.3),

$$u(x_n + h^*) - y_{n+1}^* = h^{*p+1} \sum_{j=1}^{\lambda_{p+1}} T_{p+1,j}^* D_{p+1,j} + O(h^{*p+2})$$

and the $D_{p+1,j}$ are the same in the two expressions. In the limit process considered, (x_n, y_n) is fixed as is σ (hence the new formula) and $h \to 0$. The factor

$$h^{*p+1} = \sigma^{p+1} h^{p+1},$$

so in comparing the error of the new formula to that of (2.1), we need to compare $\sigma^{p+1}\|T_{p+1}^*\|$ to $\|T_{p+1}\|$.

Notice that the analysis of the error here refers to the approximation of the *local* solution and, indeed, the whole "interpolation" issue is, in this analysis, purely local. This is the main reason why the approach is so much more soundly based than interpolation in the Adams codes.

Horn [9] showed that for the Fehlberg pair, there is a fourth-order result available at $\sigma = 3/5$ for "free"—meaning that no extra stages are necessary. With one extra stage she produces fourth-order results for all σ. The formula has the c_j^* as cubic polynomials in σ so that the scheme is easy to implement. For any *given* σ, she shows how to produce a fifth-order result with two extra stages. With five extra stages she shows how to produce a fifth-order result for all σ. The coefficients for the last two procedures were not supplied in rational form. A serious disadvantage of Horn's "interpolants" is that they do not connect smoothly from one step to the next, i.e., $y_{n+1}^*(\sigma)$ does not tend to y_{n+1} as $\sigma \to 1$.

Shampine [15] proposes and justifies a local interpolation procedure. He observes that interpolation is done only after a successful step so that one can obtain $f(x_{n+1}, y_{n+1})$ as the first evaluation of the next step. Thus approximations to the solution and derivative are available at both ends of the step. Some methods produce other accurate approximations within the step, e.g., Shintani's scheme produces both a fourth- and a fifth-order result at the midpoint. The accurate approximate solution and derivative values computed in the step from x_n to x_{n+1} are interpolated by a polynomial, a quartic in the case of Shintani's formula. By virtue of interpolating to solution and derivative approximations at both ends of the step, the polynomial approximations on the various $[x_n, x_{n+1}]$ connect up to form a globally C^1 piecewise polynomial interpolant. It does not matter whether local extrapolation is done, nor even if one decides at each step whether to do local extrapolation, as in [13]. Although apparently quite different, the approach is a special case of embedding and so can be analyzed in the same way.

We have considered using the fourth-order result at $\sigma = 3/5$ in the Fehlberg pair to obtain a free interpolant. Unfortunately, the result at this point is not very accurate as compared to the result at the endpoint and we do not recommend this. The result obtained at $\sigma = 1/2$ with Horn's scheme requiring an extra stage has very nearly the same accuracy as the fourth-order result at the end of the step, namely, for $\sigma = 1/2$,

$$\sigma^5 \|T_5^*\|_2 / \|T_5\|_2 \doteq 0.97.$$

We propose that Horn's scheme with $\sigma = 1/2$ be used with the Fehlberg pair to form a triple of formulas. Local quartic interpolation is used for other σ so as to avoid the lack of continuity from step to step of Horn's interpolant. Alternatively, the fifth-order formula at $\sigma = 1/2$ could be used at a cost of two function evaluations. This is not our preference because the fourth-order result is good enough, the cost is greater, and the coefficients and implementation are less convenient.

We have examined a number of formulas for their potential in connection with interpolation, including an improvement for a general class of methods [17] and the first Fehlberg formula. Here we just report our investigation of the Dormand-Prince pair. It appears to be possible to obtain a fourth-order result for any given σ with no extra stages. We also considered adding one stage and found that it is apparently possible to obtain a fifth-order result then at any given σ. These results are obtained with one fewer extra function evaluations than Horn needed for the Fehlberg pair. This is not surprising because the Dormand-Prince pair is really a seven-stage pair;

it is just *effectively* a six-stage pair when overlap with the following step is taken into account. Thus, in a way, the extra evaluation Horn needed with the six-stage Fehlberg pair is built into the Dormand-Prince pair. In contrast to the derivation of Dormand and Prince, we, like Horn, employ the notation of Fehlberg [3].

Let us define

$$P_{kj} = \sum_{l=1}^{k-1} \beta_{kl}\alpha_l^j, \qquad j = 1,2,3.$$

The pair of Dormand and Prince satisfies the simplifying assumptions

$$\tfrac{1}{2}\alpha_k^2 = P_{k1}, \tag{3.2}$$

$$\tfrac{1}{3}\alpha_k^3 = P_{k2} \tag{3.3}$$

for $k = 2,\ldots,6$. We shall consider two cases. One does no extra function evaluations ($s = 6$). The other does one extra evaluation ($s = 7$), and in this second case we insist that (3.2), (3.3) hold for the additional stage with $k = 7$.

We shall take $c_1^* = 0$ and solve the order condition of order 1 to get

$$c_0^* = 1 - \sum_{k=2}^{s} c_k^*.$$

Then assuming that (3.2) and (3.3) hold for $k \geqslant 2$, the remaining equations of condition through order 4 for (3.1) have the reduced form

$$\sum_{k=2}^{s} c_k^*\alpha_k^j = \frac{\sigma^j}{j+1}, \qquad j = 1,2,3, \tag{3.4}$$

$$\sum_{k=2}^{s} c_k^*\beta_{k1} = 0. \tag{3.5}$$

Case I. We seek a "free" approximation at $x_n + \sigma h$ and so have $s = 6$. On using the facts that $\alpha_6 = 1$ and $\beta_{61} = 0$, we can rewrite (3.4), (3.5) as

$$\sum_{k=2}^{5} c_k^*\alpha_k^j = \frac{\sigma^j}{j+1} - c_6^*, \quad j = 1,2,3, \qquad \sum_{k=2}^{5} c_k^*\beta_{k1} = 0.$$

For a given σ we regard c_6^* as a parameter and solve the linear system for c_2^*, c_3^*, c_4^*, c_5^*. All the α_k and β_{k1} are given, and it turns out that the matrix is nonsingular. This specifies $c_2^*,\ldots,c_5^*$ as linear functions of c_6^*.

There are nine truncation error coefficients $T_{5,i}$ which would need to be set to zero to get a fifth-order formula. This is not possible in the present circumstances, so we ask if it is possible to select c_6^* to obtain an "optimal" fourth-order formula. Everything is specified except for c_6^* and the $c_2^*,\ldots,c_5^*$ which are linear functions of c_6^*. The c_k^* appear linearly in the truncation error coefficients so that

$$T_{5,i} = \zeta_i + \rho_i c_6^*, \qquad i = 1,\ldots,9,$$

for suitable ζ_i, ρ_i. But then it is easy to determine the c_6^* which minimizes $\|T_5\|_2$, namely,

$$c_6^* = -\sum_{i=1}^{9} \zeta_i\rho_i \Big/ \sum_{i=1}^{9} \rho_i^2.$$

The scheme described was programmed in exact rational arithmetic. The formula for $\sigma = 1/2$ is given in the next section. Its accuracy is quite pleasing. When referred to the *modified* fourth-order formula,

$$\sigma^5 \| T_5^* \|_2 / \| T_5 \|_2 \doteq 0.51$$

which is considerably more favorable than with Horn's interpolant which costs a function evaluation. It is as good as that of the England formula considered as an example in [15].

Case II. Now a stage is added so that $s = 7$. Five of the equations of condition at order 5 are satisfied if, in addition to (3.4), (3.5),

$$(3.6) \qquad \sum_{k=2}^{7} c_k^* \alpha_k^j = \frac{\sigma^j}{j+1}, \qquad j = 4,$$

$$(3.7) \qquad \sum_{k=2}^{7} c_k^* \alpha_k \beta_{k1} = 0.$$

In these equations α_7, β_{71}, σ are to be regarded as parameters. We much prefer $\sigma = 1/2$, but are prepared to choose a different value, if necessary, to make the system solvable. A value $\alpha_7 = 1/2$ is reasonable, considering the other α_k used by Dormand and Prince, and is convenient. In our limited experimentation the equations (3.4)–(3.7) always formed a nonsingular system for the determination of $c_2^*, \ldots, c_7^*$. The coefficient $c_7^* \neq 0$, because only fourth-order formulas can be found with $c_7^* = 0$.

Three equations of condition at order 5 are satisfied if, in addition,

$$\sum_{k=2}^{7} c_k^* P_{k3} = \frac{\sigma^4}{20}.$$

We have not yet specified β_{7j} for $j = 2, \ldots, 6$. This will be done so as to satisfy the simplifying assumptions (3.2), (3.3) for $k = 7$ and this equation. A little manipulation shows that the three equations are equivalent to

$$\beta_{72}\alpha_2 + \beta_{73}\alpha_3 + \beta_{74}\alpha_4 = \frac{1}{2}\alpha_7^2 - [\beta_{71}\alpha_1 + \beta_{75}\alpha_5 + \beta_{76}\alpha_6],$$

$$\beta_{72}\alpha_2^2 + \beta_{73}\alpha_3^2 + \beta_{74}\alpha_4^2 = \frac{1}{3}\alpha_7^3 - [\beta_{71}\alpha_1^2 + \beta_{75}\alpha_5^2 + \beta_{76}\alpha_6^2],$$

$$\beta_{72}\alpha_2^3 + \beta_{73}\alpha_3^3 + \beta_{74}\alpha_4^3 = \frac{T}{c_7^*} - [\beta_{71}\alpha_1^3 + \beta_{75}\alpha_5^3 + \beta_{76}\alpha_6^3],$$

where

$$T = \frac{\sigma^4}{20} - \sum_{k=2}^{6} c_k^* P_{k3}.$$

For the Dormand-Prince pair, $\alpha_2 = 0.3$, $\alpha_3 = 0.8$, $\alpha_4 = 8/9$, so the Vandermonde matrix here is nonsingular, which allows us to determine β_{72}, β_{73}, β_{74} in terms of known quantities and the parameters α_7, β_{71}, σ and β_{75}, β_{76}. It only appears that both β_{75} and β_{76} are free. Because $\alpha_5 = \alpha_6 = 1$, only the sum $\beta_{75} + \beta_{76}$ appears here and elsewhere. Let us then set $\beta_{75} = 0$ and continue with β_{76} as a free parameter.

There is only one equation of condition left to satisfy at order 5,

$$\sum_{k=4}^{7} c_k^* \left[\sum_{l=3}^{k-1} \beta_{kl} \left(\sum_{m=2}^{l-1} \beta_{lm} P_{m1} \right) \right] = \frac{\sigma^4}{120}.$$

(This can be simplified, but we leave it in the form we dealt with numerically.) The values β_{72}, β_{73}, β_{74} are linear functions of β_{76}. Examination of this last equation shows that for given α_7, β_{71}, σ, it is a linear equation in the variable β_{76}. Provided that it is nonsingular, we can determine β_{76} so that it is satisfied and the resulting formula is of order 5.

After some experimentation in real arithmetic, we programmed the process described in exact rational arithmetic. The results were checked via another program for computing truncation error coefficients. A modest amount of experimentation led to the coefficients given in the next section.

Let us now go into the matter of implementation a little. When using a scheme which provides a result for "free", i.e., the Shintani fifth-order result or our fourth-order result, it is convenient always to form the result at the midpoint. (The result can then be used to improve a relative error control as described in [14].) At every step the value and slope at both ends and the value at the midpoint are returned to the user. If the user should want to interpolate within this step, it is then easy to do quartic interpolation to these five data points.

The implementation is less obvious with a scheme requiring an extra stage such as our fifth-order result or Horn's fourth-order result. We certainly do not want to form the result at the midpoint at every step. (We would then have 7-stage formulas instead of 6.) A relatively convenient way to proceed is to form at each step the two vectors

$$v = y_n + h \sum_{k=0}^{6} \beta_{7k} f_k, \tag{3.8}$$

$$w = y_n + \frac{h}{2} \sum_{j=0}^{6} c_j^* f_j \tag{3.9}$$

and return them along with the value and slope at both ends of the step. This is only one more vector of storage per step than in the simpler case. If the user should want to interpolate within this step, he would first compute

$$f_7 = f\left(x_n + \frac{h}{2}, v\right) \tag{3.10}$$

and then

$$y_{n+1/2}^* = w + \frac{h}{2} c_7^* f_7. \tag{3.11}$$

Now quartic interpolation can be done to find as many approximate solution values as are needed.

It should not be assumed that the schemes with an extra stage cannot compete with the "free" schemes. Properly used, the extra evaluation is done only on those steps where interpolation is required. At stringent tolerances such steps are comparatively rare. At crude tolerances such steps may be frequent, but then there are comparatively few steps altogether. It must be kept in mind that only one extra evaluation is required, independent of the number of interpolations to be made in the step. This is pertinent at crude tolerances and crucial when a point is being located, as for example, when one seeks to find that x for which a given function $g(x, y(x)) = 0$. The interpolation capability allows the step size to be chosen more or less independently of where answers are desired. This is often advantageous and when answers are desired at many *specific* points, it is enormously better than the traditional scheme of adjusting the step size so as to step exactly to a specific output point.

4. An Evaluation. How one intends to use formulas is important to the decision about which formula to use. This author's preference is to use the error per step criterion with local extrapolation and to emphasize measuring efficiency in the sense of cost to achieve a given accuracy. From this point of view the case for the scheme composed of the basic Dormand-Prince Runge-Kutta process with their fifth-order formula, the simple modification of the fourth-order formula given in Section 2, and the fourth-order formula for the midpoint given in Section 3 along with local quartic interpolation is very strong. This Dormand-Prince-Shampine (DPS) triple is a lot more efficient than the Fehlberg-Horn triple when local extrapolation is done. This is so in both measures of efficiency, but especially in the second. It is also a triple of superior quality; in particular, local extrapolation is better justified than with the Fehlberg pair. Differences between the formulas as regards their stability are not easy to interpret. Fortunately, they are also not particularly important. The Shintani triple appears to be the best as regards interpolation because a fifth-order result is available at the midpoint. The DPS triple would follow with its free fourth-order result at the midpoint. We have argued that the DPS triple with fifth-order result at the midpoint is competitive, as is the Fehlberg-Horn triple with fourth-order result at midpoint. These arguments are based only on the order of the results available for interpolation and their cost. At present, we are doing a more delicate truncaticn error analysis of some of these possibilities which we shall report on another occasion.

Because of their importance, we collect here the DPS formulas recommended:

α_j			$\beta_{j,k}$			
0						
$\frac{1}{5}$	$\frac{1}{5}$					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				
$\frac{4}{5}$	$\frac{44}{45}$	$\frac{-56}{15}$	$\frac{32}{9}$			
$\frac{8}{9}$	$\frac{19372}{6561}$	$\frac{-25360}{2187}$	$\frac{64448}{6561}$	$\frac{-212}{729}$		
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$\frac{-5103}{18656}$	
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$

c_j—result at $x_n + h$		c_j^*—result at $x_n + \frac{1}{2}h$
order 4	order 5	order 4
$\frac{1951}{21600}$	$\frac{35}{384}$	$\frac{6{,}025{,}192{,}743}{30{,}085{,}553{,}152}$
0	0	0
$\frac{22642}{50085}$	$\frac{500}{1113}$	$\frac{51{,}252{,}292{,}925}{65{,}400{,}821{,}598}$
$\frac{451}{720}$	$\frac{125}{192}$	$\frac{-2{,}691{,}868{,}925}{45{,}128{,}329{,}728}$
$\frac{-12231}{42400}$	$\frac{-2187}{6784}$	$\frac{187{,}940{,}372{,}067}{1{,}594{,}534{,}317{,}056}$
$\frac{649}{6300}$	$\frac{11}{84}$	$\frac{-1{,}776{,}094{,}331}{19{,}743{,}644{,}256}$
$\frac{1}{60}$	0	$\frac{11{,}237{,}099}{235{,}043{,}384}$

The formulas are used according to (2.1). The result at the midpoint is computed according to

$$y_{n+1/2} = y_n + \frac{1}{2}h \sum_{j=0}^{6} c_j^* f_j.$$

Notice that $f_6 = f(x_{n+1}, y_{n+1})$ when local extrapolation is done. At the completion of a successful step, the value and slope at both ends, namely (y_n, f_0) and (y_{n+1}, f_6), are available along with $y_{n+1/2}$. Interpolation is done by a local quartic interpolation to this data.

An alternative for the interpolation is to compute (3.8), (3.9) at each step with the coefficients that follow and then, if interpolation at this step is desired, to do the function evaluation (3.10) and then form $y_{n+1/2}^*$ according to (3.11). This furnishes a fifth-order result at the midpoint for the local quartic interpolation. The coefficients are

$\beta_{7,k}$	c_k^*
$\frac{-33{,}728{,}713}{104{,}693{,}760}$	$\frac{7{,}157}{37{,}888}$
2	0
$\frac{-30{,}167{,}461}{21{,}674{,}880}$	$\frac{70{,}925}{82{,}362}$
$\frac{7{,}739{,}027}{17{,}448{,}960}$	$\frac{10{,}825}{56{,}832}$
$\frac{-19{,}162{,}737}{123{,}305{,}984}$	$\frac{-220{,}887}{2{,}008{,}064}$
0	$\frac{80{,}069}{1{,}765{,}344}$
$\frac{-26{,}949}{363{,}520}$	$\frac{-107}{2{,}627}$
	$\frac{-5}{37}$

5. Acknowledgments. J. R. Dormand kindly brought to the author's attention his important work with Prince, supplied preprints and reprints of their work, and provided valuable details not appearing in [1]. L. S. Baca carried out most of the computations reported here. I. Gladwell and an anonymous referee provided the author with exceptionally helpful criticism. The author acknowledges with gratitude his debt to all.

Numerical Mathematics Division
Sandia National Laboratories
Albuquerque, New Mexico 87185

1. J. R. Dormand & P. J. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, v. 6, 1980, pp. 19–26.

2. W. H. Enright & T. E. Hull, "Test results on initial value methods for non-stiff ordinary differential equations," *SIAM J. Numer. Anal.*, v. 13, 1976, pp. 944–961.

3. E. Fehlberg, *Low-Order Classical Runge-Kutta Formulas with Stepsize Control and Their Application to Some Heat Transfer Problems*, Rept. NASA TR R-315, George C. Marshall Space Flight Center, Marshall, Alabama, 1969.

4. E. Fehlberg, "Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme," *Computing*, v. 6, 1970, pp. 61–71.

5. I. Gladwell, "Initial value routines in the NAG library," *ACM Trans. Math. Software*, v. 5, 1979, pp. 386–400.

6. G. Hall & J. M. Watt, eds., *Modern Numerical Methods for Ordinary Differential Equations*, Clarendon Press, Oxford, 1976.

7. M. K. Horn, *Scaled Runge-Kutta Algorithms for Handling Dense Output*, Rept. DFVLR-FB81-13, DFVLR, Oberpfaffenhofen, F.R.G., 1981.

8. M. K. Horn, *Scaled Runge-Kutta Algorithms for Treating the Problem of Dense Output*, Rept. NASA TMX-58239, L. B. Johnson Space Center, Houston, Texas, 1982.

9. M. K. Horn, "Fourth- and fifth-order, scaled Runge-Kutta algorithms for treating dense output," *SIAM J. Numer. Anal.*, v. 20, 1983, pp. 558–568.

10. T. E. Hull & W. H. Enright, *A Structure for Programs that Solve OrdinaryDifferential Equations*, Rept. 66, Dept. Comp. Sci., Univ. of Toronto, Canada, 1974.

11. T. E. Hull, W. H. Enright, B. M. Fellen & A. E. Sedgwick, "Comparing numerical methods for ordinary differential equations," *SIAM J. Numer. Anal.*, v. 9, 1972, pp. 603–637.

12. T. E. Hull, W. H. Enright & K. R. Jackson, *User's Guide for* DVERK—*a Subroutine for Solving Non-Stiff ODE's*, Rept. 100, Dept. Comp. Sci., Univ. of Toronto, Canada, 1976.

13. L. F. Shampine, "Local extrapolation in the solution of ordinary differential equations," *Math. Comp.*, v. 27, 1973, pp. 91–97.

14. L. F. Shampine, *Robust Relative Error Control*, Rept. SAND82-2320, Sandia National Laboratories, Albuquerque, New Mexico, 1982.

15. L. F. Shampine, "Interpolation for Runge-Kutta methods," *SIAM J. Numer. Anal.*, v. 22, 1985, pp. 1014–1027.

16. L. F. Shampine, "The step sizes used by one-step codes for ODEs," *IMACS J. Numer. Anal.*, v. 1, 1985, pp. 95–106.

17. L. F. Shampine, "Local error estimation by doubling," *Computing*, v. 34, 1985, pp. 179–190.

18. L. F. Shampine & H. A. Watts, "Comparing error estimators for Runge-Kutta methods," *Math. Comp.*, v. 25, 1971, pp. 443–455.

19. L. F. Shampine & H. A. Watts, *Practical Solution of Ordinary Differential Equations by Runge-Kutta Methods*, Rept. SAND76-0585, Sandia National Laboratories, Albuquerque, New Mexico, 1976.

20. L. F. Shampine & H. A. Watts, *DEPAC-Design of a User Oriented Package of ODE Solvers*, Rept. SAND79-2374, Sandia National Laboratories, Albuquerque, New Mexico, 1980.

21. L. F. Shampine, H. A. Watts & S. M. Davenport, "Solving non-stiff ordinary differential equations—the state of the art," *SIAM Rev.*, v. 18, 1976, pp. 376–411.

22. H. Shintani, "On a one-step method of order 4," *J. Sci. Hiroshima Univ. Ser. A-I*, v. 30, 1966, pp. 91–107.