A technique for comparing automatic quadrature routines*

J. N. Lyness and J. J. Kaganove

Applied Mathematics Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, USA

The present unconstrained proliferation of automatic quadrature routines is a phenomenon which is wasteful in human time and computing resources. At the root of the problem is an absence of generally acceptable standards or benchmarks for comparing or evaluating such routines. In this paper we describe a general technique, based on the nature of the performance profile, which can be used for evaluation of routines.

(Received February 1976)

1. Introduction

There are many automatic quadrature routines in existence and more are being constructed daily. A few of these have been published in the open literature. Most subroutine libraries contain several locally developed automatic quadrature routines. There are many reasons for this proliferation, the principal one being that there are no generally accepted standards or benchmarks by which one routine can be compared with another. Thus, any individual who constructs a routine can find some problems for which it is more efficient than an existing available routine, and with this evidence, arrange for its inclusion in the local subroutine library. Existing routines are not removed because there are other problems for which they are more efficient than the new routine.

It is generally agreed that the structure of automatic quadrature routines is sufficiently complicated to preclude the possibility of comparison or evaluation by analytic means and that numerical experiments have to play a basic role. The questions to which we devote ourselves in this paper are how such experiments should be constructed and how their results should be interpreted.

In Section 2 we discuss the background of the testing problem. In particular, testing automatic quadrature routines is intrinsically more complicated than testing simpler function evaluation routines. We describe briefly some previous work in this area (the Battery experiment) and draw attention to some of its defects.

The balance of the paper is devoted to describing and discussing a new evaluation technique. We term this the *performance profile* method since it is motivated by the nature of the performance profile described in Lyness and Kaganove (1976). We draw heavily on the discussion in that paper which we refer to as CAQR.

In Section 3 we introduce *problem families* and report on the form of our experiments which produce *statistical distribution functions*. In Section 4 we describe how a problem oriented user might use these statistical distribution functions to decide between various automatic quadrature routines. The discussion leads naturally to the suggestion that a quantity denoted by $v(E_{\text{quad}}(s, \varepsilon_{\text{req}}))$ be used as a basis for comparison of different routines. Briefly, $v(E_{\text{quad}}(s, \varepsilon_{\text{req}}))$ is the average number of function values required by the routine to integrate members of a specified problem family when the quadrature routine tolerance parameter has been set in such a way that an accuracy ε_{req} is obtained with probability s.

In Section 5 we discuss the sort of results obtained in this way, illustrated by graphical machine output using three rather mediocre routines from our local library. An important experimental result is that the results of comparisons based on $v(E_{\text{quad}}(s, \varepsilon_{\text{req}}))$ are almost independent of the value of s assigned for the comparison. If this were not the case, the technique would not be of practical use.

Finally, in Section 6, some of the deeper implications of this technique are discussed.

2. Background: the battery experiment

The problem of evaluating and comparing numerical software is not new. Considerable success has been achieved in this field for many types of numerical software. However, for software dealing with the more sophisticated problems, serious questions about evaluation remain.

There seems to be a natural division of software into problems for which a finite decision process is known, and problems for which it is known that no such process exists. This is discussed at some length in our previous paper, CAQR. In this paper we deal specifically with automatic quadrature routines which have a calling sequence of the type

QUAD(A,B,EPQUAD,FUN, . . .) .

The features which such routines have in common, but which are not shared by routines based on a finite decision process, include the following:

1. It is possible for the routine, however well coded, to return an entirely incorrect result. This happens because the exact arithmetic algorithm on which it is based is unreliable, and this property is quite independent of machine arithmetic characteristics.
2. There is a cost versus reliability trade-off. One can alter any routine to make it significantly more reliable, if at the same time one is prepared to make it significantly more expensive.
3. Minor changes in the choice of integrand function may lead to major differences in performance. (This is a consequence of the nature of the performance profile discussed in CAQR, Section 3.)

These properties make comparative evaluation a difficult process. This paper is devoted to the design of a sophisticated testing procedure which can differentiate between several different routines, all of which have these properties.

In the discussion we use the following notation:

$\varepsilon_{\text{quad}}$: the value of the input tolerance parameter EPQUAD.
ε_{req} : the tolerance required by a user.
ε_{act} : the accuracy of the result actually returned by a routine.
v : the number of function values required by the routine.

To illustrate the basic difficulty we consider an isolated set of results. Suppose, for a specified problem P_1, we use three different routines Q_1, Q_2, Q_3 with $\varepsilon_{\text{quad}} = 10^{-3}$ and the results are as follows:

*Work performed under the auspices of the US Energy Research Development Administration.

$$\left.\begin{array}{l} Q_1: \varepsilon_{act} = 0{\cdot}05 \times 10^{-3};\ \ \nu = 220 \\ Q_2: \varepsilon_{act} = 0{\cdot}2 \ \ \times 10^{-3};\ \ \nu = 200 \\ Q_3: \varepsilon_{act} = 1{\cdot}1 \ \ \times 10^{-3};\ \ \nu = 100 \end{array}\right\} P_1 \qquad (2.1)$$

The question is: 'Which is the better routine for this problem and this accuracy?' One can make good arguments in favour of any of these by using different bases for comparison. Suppose one carried out another experiment using a slightly different integrand function and found

$$\left.\begin{array}{l} Q_1: \varepsilon_{act} = 0{\cdot}8 \times 10^{-3};\ \ \nu = 180 \\ Q_2: \varepsilon_{act} = 500 \times 10^{-3};\ \ \nu = \ 30 \\ Q_3: \varepsilon_{act} = 0{\cdot}1 \times 10^{-3};\ \ \nu = 210 \end{array}\right\} P_2\ . \qquad (2.2)$$

It is unlikely that the same basis for comparison would yield the same choice of routine.

Even with this basic situation, it might be hoped that if one carried out a sufficiently large number of numerical experiments using many different integrand functions, the overall results would reveal some recognisable trend of preference. Two major attempts in this direction are fairly well known, and have the same overall structure. One of these, reported in Casaletto, Picket and Rice (1969) was carried out at Purdue University. The other, reported in Kahaner (1971) was carried out at Los Alamos Scientific Laboratory. We refer to these projects as battery experiments and we describe one in broad outline.

Kahaner's investigation involves $N_P(= 21)$ different integrand functions, together with limits of integration, and $N_Q(= 11)$ different automatic quadrature routines. Each integral is evaluated by each routine using $N_E(= 8)$ different tolerances ε_{quad}. Since the true value of the integral is known in each case, he obtains from each of these $N_P N_Q N_E(= 1848)$ runs two results ε_{act} and ν. Kahaner also recorded the machine time. In his article, Kahaner gives complete details of the N_P different integrand functions, and source listings of each automatic quadrature routine for which a readily accessible reference is not available. He also gives the complete set of results for three of the values of ε_{quad}. This list of $3N_P N_Q(= 693)$ triplet entries occupies 17 printed pages.

Thus, this project is exceptionally well documented. Any reader may examine the results and, if he wishes, form his own conclusions. But such an examination of the results shows that it is very difficult to extract from them much in the way of an overall conclusion. Most routines did very well in some problems and very badly in others.

Kahaner's experimental technique is completely objective; if among his routines there had existed one overall superior routine, this investigation would have found it. However, interpretation of these results turned out to be quite subjective. In the end, based principally but not exclusively on an 'average reliability' and an 'average speed' for each routine, Kahaner selected three for the Los Alamos subroutine library.

Our objections to the battery experiment turn principally on one key aspect. This is, that in an attempt to obtain wide generality, the integrand functions are chosen to be 'as different from each other' as possible.

The obvious consequence is that one does not use integrand functions which are close to one another. For example if P_1 (the problem whose results are listed in (2.1) above) is included in the list of integrand functions, then P_2 (the neighbouring problem whose results are listed in (2.2)) is not included. Since the results for P_1 and P_2 are so very different from each other, it is unfair and may be misleading to include one but not the other. If the results for P_1 are included and the user's problem happens to be P_2, he may imagine, incorrectly, that the P_1 results are applicable to his problem too. Such a conclusion would be valid only if the performance profile were smooth. Since this is not the case, the problem oriented user can be grossly misled.

An incidental defect, which could easily be corrected, is that the lack of fairness in choosing the first integrand rather than the second—or vice versa—is compounded in these experiments for the following reason. In Section 3 of CAQR, it was pointed out that if a chance failure occurs at a particular tolerance, a failure due to the same basic cause may occur at stricter tolerances. Thus, an unlucky choice of integrand function for some routine may involve it in more than one failure.

The present authors, in designing the 'performance profile' evaluation technique, have been motivated by this unfair aspect to carry out much more extensive experiments. By using a problem family of similar integrand functions (defined in the next section), we essentially include both P_1 and P_2 and many other neighbouring integrands. By the same token, it turns out that, based on the results for a problem family, a problem oriented user can take advantage of only those sections of the overall results which seem to apply to his problem. This has led to other complications which are described in the next sections.

3. The statistical distribution function

The technique which we shall propose is based on the nature of the performance profile which is discussed in detail in Section 3 of CAQR. If we are interested in a particular attribute of an integrand function we choose a *problem family*, each of whose members has this attribute. An individual member of a family is specified by assigning a numerical value to an additional parameter λ, which may appear in the integrand function $f(x; \lambda)$, or in the integration limits $a(\lambda)$ and $b(\lambda)$. λ may take any value within a specified range, i.e. $\lambda_- \leq \lambda \leq \lambda_+$. As an example one problem family is specified by

$$\begin{aligned} &a = 1,\ b = 2 \\ &f(x; \lambda) = \mu((x - \lambda)^2 + \mu^2)^{-1},\ \mu = 0{\cdot}01 \qquad (3.1) \\ &0{\cdot}998 \leq \lambda \leq 2{\cdot}02 \end{aligned}$$

Each member of this family has a peak of height 100 and half width 0·01 within or very close to the end of the integration interval. The exact value of the definite integral is of order 1, specifically between 1·00 and 3·12.

If our numerical experiments are limited to a single automatic quadrature routine (or algorithm) and a single problem family, then each individual experiment or run may be specified by two (input) parameters λ and ε_{quad}. Corresponding to each such pair we may define $\varepsilon_{act}(\lambda; \varepsilon_{quad})$ the error $|If - Qf|$ of the result returned by the routine (or algorithm) and $\nu(\lambda; \varepsilon_{quad})$ the number of function values required by the routine to return this result. A plot of the function $\varepsilon_{act}(\lambda; \varepsilon_{quad})$ against λ for a fixed value of ε_{quad} is a *performance profile*. This is illustrated in CAQR (Fig. 1).

A fundamental property of automatic quadrature routines is that the functions $\varepsilon_{act}(\lambda; \varepsilon_{quad})$ and $\nu(\lambda; \varepsilon_{quad})$ are rapidly varying discontinuous functions of λ. (As functions of ε_{quad} they are generally piecewise constant and usually but not invariably monotonic increasing and decreasing respectively.) Because of this they are not suitable as a direct measure of the efficiency of an automatic quadrature routine.

The difficulties encountered in interpreting battery type experiments may be traced to this property. By relying on individual values of the input parameters λ and ε_{quad} one has introduced into the results a significant arbitrary component which we believe is responsible for frustrating the evaluation process.

As the basis for our evaluation technique we have introduced measures which treat the problem family as a whole. One of these is

$$\nu(\varepsilon_{quad}) = \frac{1}{\lambda_+ - \lambda_-} \int_{\lambda_-}^{\lambda_+} \nu(\lambda; \varepsilon_{quad})\, d\lambda \qquad (3.2)$$

an average function value count.

In the context of class 1 software (see CAQR, Section 4) the

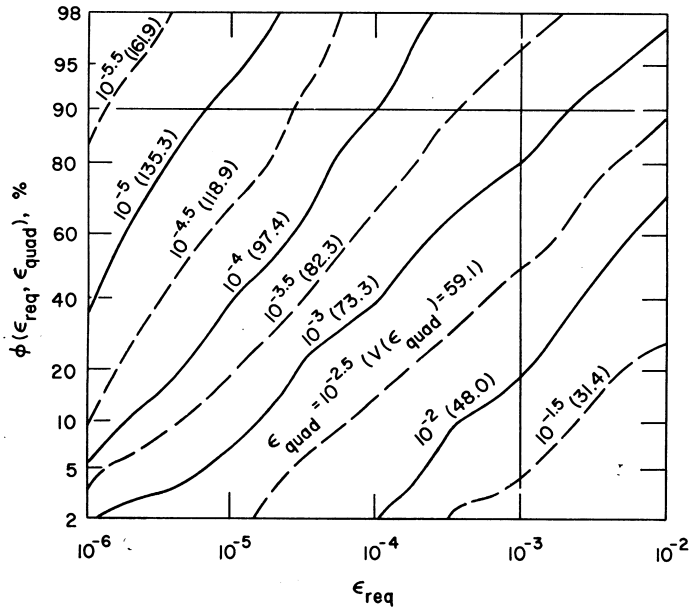

Fig. 1 The statistical distribution functions $\phi(\varepsilon_{req}, \varepsilon_{quad})$ as a function of ε_{req} for Problem family (3.1) and routine ANC4

root mean square average of ε_{act} is a simple and useful measure as errors arise mainly from machine arithmetic and are rarely greater than three or four units of the machine accuracy parameter. However, with quadrature routines there are usually a few very large errors and this sort of average would be unduly influenced by individual wild values. Instead we retain the whole distribution function

$$\phi(x; \varepsilon_{quad}) = \begin{pmatrix}\text{Proportion of values of } \lambda \text{ for which} \\ |\varepsilon_{act}(\lambda; \varepsilon_{quad})| \leq x\end{pmatrix} \quad (3.3)$$

$$= \frac{1}{\lambda_+ - \lambda_-} \int_{\lambda_-}^{\lambda_+} H(x - |\varepsilon_{act}(\varepsilon_{quad}; \lambda)|)\, d\lambda$$

where $H(t)$ stands for the unit step function (Heaviside function)

$$H(t) = \begin{matrix}1 & t > 0 \\ 1/2 & t = 0 \\ 0 & t < 0\end{matrix} . \quad (3.4)$$

We calculate these quantities using Monte Carlo integration to approximate the integrals in (3.2) and (3.3). Thus we make m runs to obtain a set of results:

$$\varepsilon_{act}(\lambda_i; \varepsilon_{quad}); \; v(\lambda_i; \varepsilon_{quad}) \quad i = 1, 2, \ldots, m .$$

The values of λ are chosen from the range (λ_-, λ_+) using a (repeatable) random number generator. The quantities

$$v_m(\varepsilon_{quad}) = \frac{1}{m} \sum_{i=1}^{m} v(\lambda_i; \varepsilon_{quad}) \quad (3.5)$$

and

$$\phi_m(x; \varepsilon_{quad}) = \frac{1}{m} \begin{pmatrix}\text{Number of values of } i \text{ for which} \\ |\varepsilon_{act}(\lambda_i; \varepsilon_{quad})| \leq x\end{pmatrix}$$

$$= \frac{1}{m} \sum_{i=1}^{m} H(x - |\varepsilon_{act}(\lambda_i; \varepsilon_{quad})|) \quad (3.6)$$

are approximations to (3.2) and (3.3), respectively.

As in any statistically based experiment, the size of the sample used has to be chosen with care with a view to the accuracy required in the results. Naturally, in our experiments we have attempted to do this. But one of the advantages of this approch is that in cases of subsequent doubt or disbelief in the data on which the conclusions are based, the distribution function can be recomputed by any interested person, and conclusions can be altered if significant differences are found. Once a problem family is defined, and a quadrature routine is chosen together with a value of ε_{quad}, the functions $\phi(x; \varepsilon_{quad})$ and $v(\varepsilon_{quad})$ are well defined and can be determined. What we have described above is one method of doing this.

We have found in practice that relatively small values of m, such as $m = 100$, are sufficient to obtain a clear idea of the form of these functions. However, we have been very cautious in any experiment whose results are recorded and have generally used $m = 1{,}000$. We believe that this has produced $\phi(x; \varepsilon_{quad})$ to within 1% for most of the range of x. Specifically for values of x for which

$$0{\cdot}01 < \phi(x; \varepsilon_{quad}) < 0{\cdot}99$$

the approximation satisfies

$$|\phi_m(x; \varepsilon_{quad}) - \phi(x; \varepsilon_{quad})| \leq 0{\cdot}01$$

and

$$|v_m(\varepsilon_{quad}) - v(\varepsilon_{quad})| \leq 0{\cdot}01\, v(\varepsilon_{quad}) .$$

Spot checks have indicated that the accuracy is better than this. We shall assume henceforth that this calculation has been properly carried out and that the functions $\phi(x; \varepsilon_{quad})$ and $v(\varepsilon_{quad})$ are available to the required accuracy.

In **Fig. 1** we illustrate the nature of the results we have obtained. Here the problem family is the one defined in (3.1) and the routine is a local one called ANC4 (an adaptive Newton Cotes routine). Portions of distribution functions $\phi(x; \varepsilon_{quad})$ expressed as a percentage are shown for a range of values of ε_{quad}. In practice, except in the case of a few pilot calculations, these statistical distribution functions are not plotted. The information from which they could be plotted is retained on tape and used directly to obtain such quantitative results as we may require. The illustration is useful in the context of describing precisely what data is being calculated and in describing the subsequent use made of this data.

Each curve is labelled with the value of ε_{quad} and in parentheses the value of $v(\varepsilon_{quad})$. The ordinate is not linear but is scaled in such a way that if $\log \varepsilon_{req}$ were normally distributed, the curve would appear as a straight line. The parts of the curves not illustrated (i.e. $\phi > 98\%$ and $\phi < 2\%$) reflect a distribution with a more pronounced tail than a normal distribution. The 'bumps and wiggles' in these curves arise from the nature of the performance profile.

4. Comparing different routines

A set of statistical distribution functions of the type illustrated in Fig. 1, corresponding to different problem families and different automatic quadrature routines, provides a wealth of information which experts might spend a great deal of time examining with a view to determining defects or advantages of particular routines in various contexts. However, we are primarily concerned here with providing a non-expert, the user, with information that he might require for his particular problem. To this end we discuss in this section how a sophisticated user might use this information. Then having determined what he would do, we automate this process. Instead of presenting to the user the raw statistical distribution functions, we present the required information in graphical form.

It seems most unlikely that a particular user will ever have a problem which coincides precisely with a member of a problem family which has already been investigated. But it has been our experience that, in any difficult problem, there is a salient feature of the integrand which is primarily responsible for difficulties to be encountered in numerical integration. If this feature can be isolated one can, as a practical measure, proceed on the basis of statistics obtained for a problem family having only this feature. Either such statistics would be already

available or they could be specially obtained.

By way of illustration, we suppose that the dominant feature in the user's problem family is a peak of the type which occurs in the problem family described in the previous section (3.1) and that he requires an accuracy $\varepsilon_{req} = 10^{-3}$. He has a choice of three automatic quadrature routines, called ANC4, ASIMP and ROMBERG, and for each he has available a set of statistical distribution functions corresponding to problem family (3.1).

First he might examine the situation with respect to ANC4. If he goes straight ahead and sets $\varepsilon_{quad} = 10^{-3}$, reference to the Fig. 1 shows that he has a success probability of $s = \phi(10^{-3}; 10^{-3}) = 79{\cdot}7\%$ and that $v(10^{-3}) = 73{\cdot}3$. That is, he must expect to obtain a less accurate result one time in five and the cost, on average, will be 73·3 function values.

If he would like to increase his success probability, he can do so simply by using a smaller value of ε_{quad}. Thus, if he uses $\varepsilon_{quad} = 10^{-3\cdot5}$, he has a success probability of obtaining $\varepsilon_{req} = 10^{-3}$ given by $\phi(10^{-3}, 10^{-3\cdot5}) = 96{\cdot}1\%$. Naturally, he pays for this in terms of additional function values, i.e. $v(10^{-3\cdot5})$ is 82·3. If he uses $\varepsilon_{quad} = 10^{-4}$, he increases this success probability to $\phi(10^{-3}, 10^{-4}) = 99{\cdot}8\%$ but the average cost is now up to $v(10^{-4}) = 97{\cdot}4$.

Thus, to make effective use of this routine in this class of problems, the user has to decide both the accuracy he requires and the probability of success he is prepared to pay for in terms of function values. Once he has fixed these parameters, he may use the figures given here to find what value of ε_{quad} to use and what it is likely to cost.

Placing the onus on the user to provide a success probability is a departure from the normal practice. If he declines to state one, he may simply use $\varepsilon_{quad} = \varepsilon_{req}$ and this is done for him by the code—usually in rather an arbitrary fashion. As a subjective comment, it seems to the authors that the user should be warned unambiguously that the routine may fail, in fact that statistically it will fail. He is more likely to take this warning seriously if he is asked to state what he wants his failure rate to be, or more precisely, what success probability he is prepared to pay for in terms of number of function values. The result of even putting questions like these to a user can only be favourable. Possibly the best result would be for him to avoid the use of the relatively expensive automatic quadrature routine, and to code his own special integration in a more reliable manner, taking into account special features of his problem. But even if he does go ahead and makes use of an automatic quadrature routine, he is at least clear in his own mind about the nature of the gamble he is taking.

The authors' main criticism of most of the documentation we have seen on quadrature routines is that this fundamental aspect of the whole problem, far from being clearly spelled out, is usually suppressed. The user is often led to believe that only in really pathological cases is there any significant chance of the routine failing at all. The truth seems to be that for moderately difficult integrand functions, if a reasonably economical routine is used, following the instructions, it fails about 5 to 15% of the time.

Returning to the problem at hand, let us suppose that this user is definitely going to use one of his three automatic quadrature routines and decides that $s = 90\%$ is sufficient. By means of a double interpolation process on the curves in Fig. 1, he can determine that if he sets $\varepsilon_{quad} = 10^{-3\cdot3}$ then $\phi(\varepsilon_{req}; \varepsilon_{quad}) = \phi(10^{-3}; 10^{-3\cdot3}) = 90\%$. The average cost is $v(\varepsilon_{quad}) = v(10^{-3\cdot3}) = 78{\cdot}7$. It is convenient to denote by E_{quad}, the value of ε_{quad} chosen in this way. E_{quad} is a function of s and ε_{req}. The colloquial manner in which this user could justify this decision is to say that he has decided, on the basis of the curves in Fig. 1, to introduce a safety factor $10^{-0\cdot3}$ into ε_{quad} so as to increase his success probability from 79·7% to 90% and his average cost from 73·3 to 78·7.

This is the situation with respect to ANC4. Now, having decided his two parameters $s = 90\%$ and $\varepsilon_{req} = 10^{-3}$, he may apply the same procedure to the statistical distribution curves corresponding to ROMBERG and ASIMP. These curves are not given here, but the result is

$$E_{quad} = 10^{-2\cdot1} \text{ and } v(E_{quad}) = 480 \text{ for ROMBERG}$$

and

$$E_{quad} = 10^{-2\cdot7} \text{ and } v(E_{quad}) = 71 \text{ for ASIMP} .$$

Based on these cost estimates, he may conclude that for these parameters $\varepsilon_{req} = 10^{-3}$ and $s = 90\%$ and this sort of problem (3.1), ROMBERG is far more expensive than the other two routines. Of these other two, ASIMP is the more economical but by a very small margin.

The process by which $E_{quad}(s, \varepsilon_{req})$ and $v(E_{quad})$ are obtained from the statistical distribution function is a standard procedure involving interpolation. There is no need to burden a prospective user with this calculation. For his purposes a plot of $E_{quad}(s, \varepsilon_{req})$ and $v(E_{quad}(s, \varepsilon_{req}))$ is sufficient and plots of this type may be obtained automatically. **Figs. 2** and **3** are plots corresponding to the problem discussed in this section. These provide the results required by the user for a wide range of ε_{req} with $s = 90\%$.

The prospective user need only glance at Fig. 3 to obtain a clear idea of the relative cost involved using these three routines when he assigns $s = 90\%$. We consider the question of the effect of the choice $s = 90\%$ (rather than say $s = 80\%$) on the outcome in the next section.

We close this section by drawing attention to the fact that the quantity $E_{quad}(s, \varepsilon_{req})$ plotted in Fig. 2 is an approximation to a well defined functional of $\phi(x; \varepsilon_{quad})$ which is defined in (3.6), namely:

Definition:
Given s and ε_{req}, $E_{quad}(s, \varepsilon_{req})$ is the smallest positive value of ε_{quad} for which

$$s = \phi(\varepsilon_{req}; \varepsilon_{quad}) . \qquad (4.1)$$

For certain values of s and ε_{req}, $E_{quad}(s, \varepsilon_{req})$ may not exist. For example, a quadrature algorithm which employs a maximum of thirty function values may be incapable of obtaining $\varepsilon_{req} = 10^{-15}$ with a success probability of 90%.

Thus the quantities plotted in Figs. 2 and 3 are approximations to mathematically defined quantities based on definitions (3.2), (3.3) and (4.1). We have calculated these using Monte Carlo integration and interpolation. They can be verified (or shown to be inaccurate) by any independent computation.

5. Discussion and practical organisation

Fig. 3 illustrates the cost of using any of three routines on members of a single problem family when one requires a stated accuracy ε_{req} with statistical confidence $s = 90\%$. Leaving aside other routines for the moment, it seems that to document fully the behaviour of these routines one requires figures corresponding to Fig. 3 for many other problem families and other values of s.

Up to this time we have treated nine problem families and up to sixteen quadrature routines. We have obtained plots for many values of s, from $s = 50\%$ up to $s = 95\%$. An outstanding common feature of these plots has been that for the same problem family, there is practically no qualitative difference between plots for different values of s. This is illustrated in **Figs 4** and **5** which have been obtained in the same manner as Fig. 3 except that $s = 95\%$ and $s = 80\%$ are used as confidence levels in place of $s = 90\%$. Except for the obvious point that more function values are required by each routine for a higher confidence level, there is no noticeable difference between these

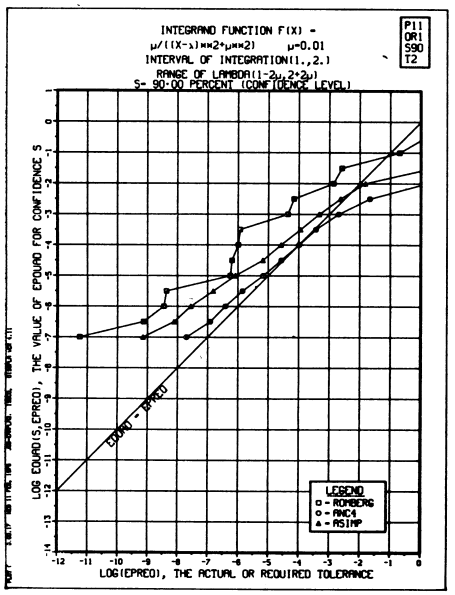

Fig. 2

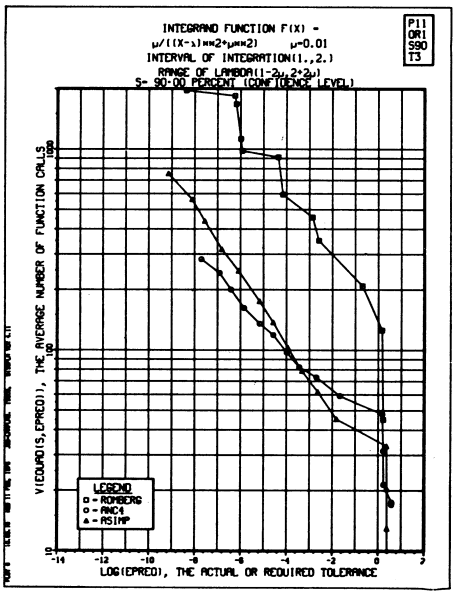

Fig. 3

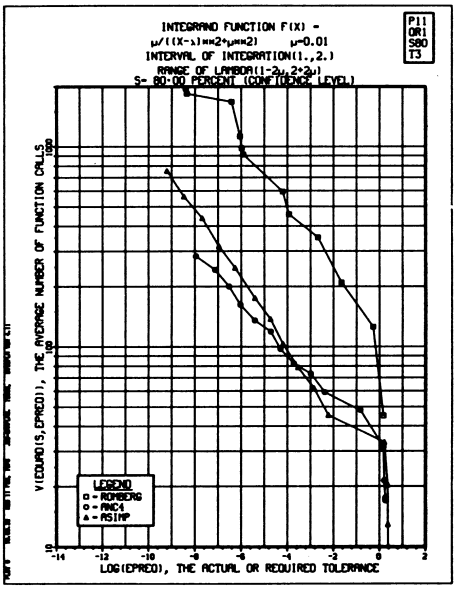

Fig. 4

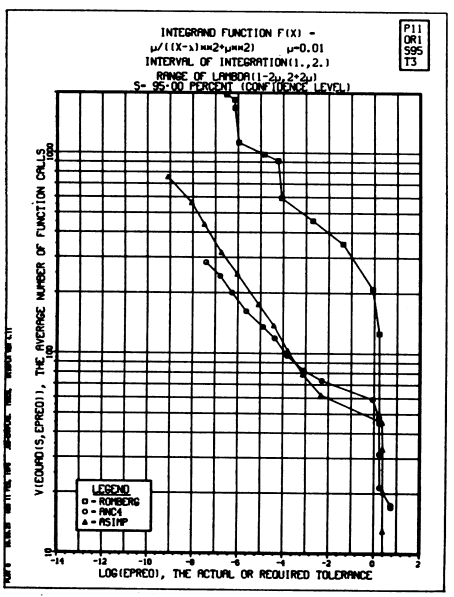

Fig. 5

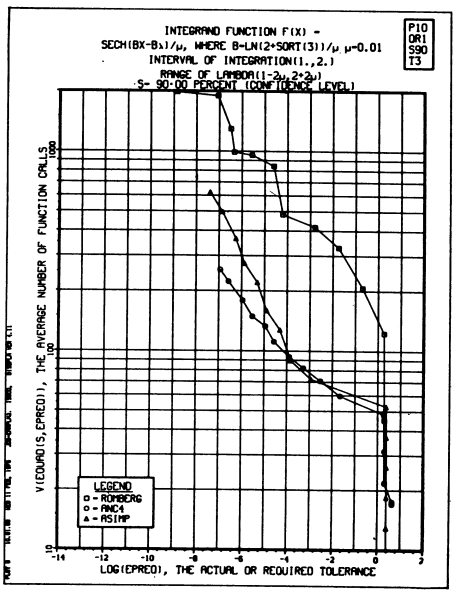

Fig. 6

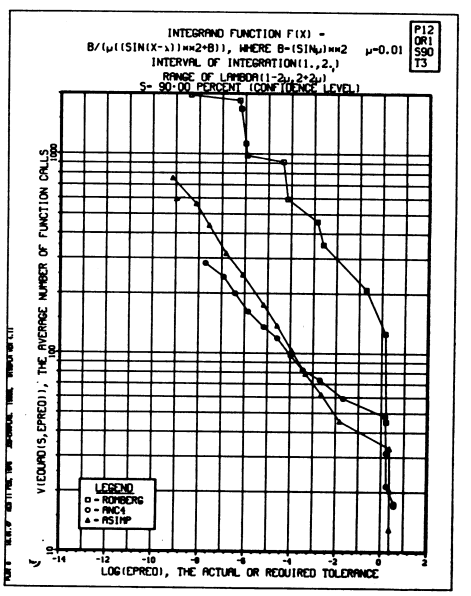

Fig. 7

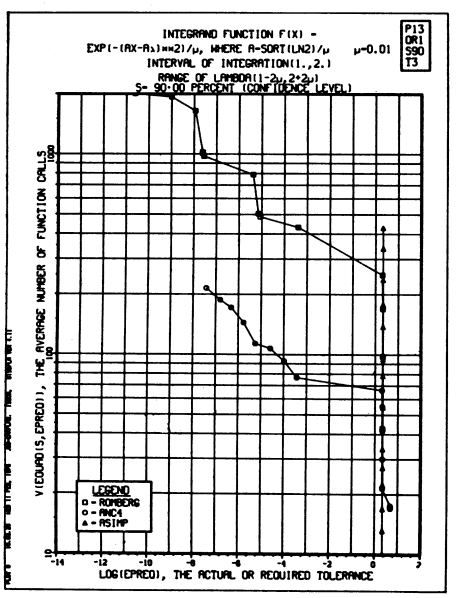

Fig. 8

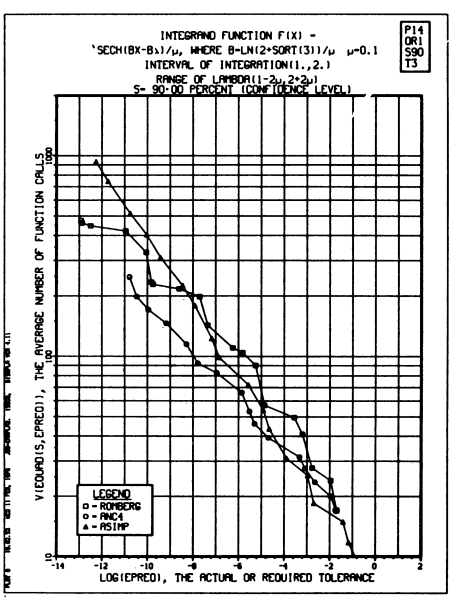

Fig. 9

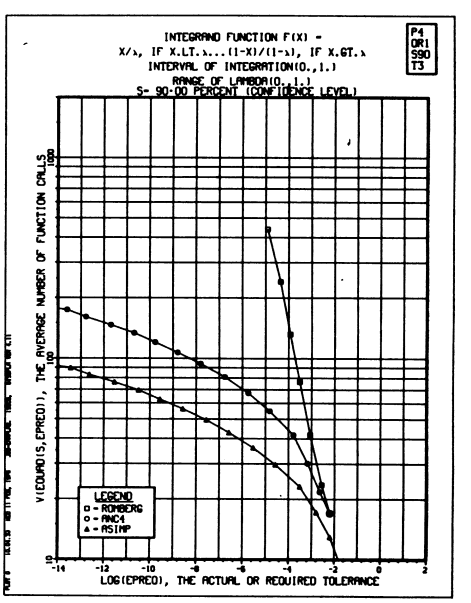

Fig. 10

figures. Even the crossover point between the ANC4 and the ASIMP plots occurs at the same value $10^{-3.6}$ of ε_{req}. We believe that this property of the results is important, and we hope that it will occur in other problem families too. If it did not occur, this method of evaluation would not be useful in practice.

In **Figs. 6, 7** and **8,** we present some of the results using problem families which resemble to some extent problem family (3.1) used in Fig. 3. Each has a peak of height 100 and half width 1/100. The specification of the problem family appears at the head of the figure. For these, the resulting plots differ qualitatively in only a minor way, except in Fig. 8 where ASIMP consistently fails.

For the interested reader, ASIMP may return a result based on only nine function evaluations while both ROMBERG and ANC4 require at least seventeen function evaluations. This turns out to be critical for the problem family illustrated in Fig. 8. Whilst, in this paper, we are concerned only with program evaluation, this figure illustrates possible uses of this evaluation technique in the context of routine construction.

In **Figs. 9** and **10,** we present some of the results for other quite different problem families. The problem illustrated in Fig. 9 is a very easy one and there is usually a factor of only two between the cost of the best and the worst routine. The integrand function corresponding to Fig. 10 has a discontinuous first derivative.

The results given in Figs. 2 to 10 are intended only to illustrate the nature of the results on which automatic quadrature routines can be evaluated. The three routines treated here were taken from our local library and do *not* represent the present state of the art. These routines, and other local routines, were used in a pilot project to see whether this method for evaluating routines is feasible. On the basis of these results we have decided to continue with the project and we give here a very brief description of its organisation.

The central feature is a data bank, stored on tape. As the relatively expensive series of numerical experiments are carried out, the results, in the form of data for construction of statistical distribution functions, are stored in the data bank. There is no need to repeat these experiments. When a new routine is submitted we carry out these experiments for this routine only, using all presently treated problem families. If a new problem family is suggested, we may carry out these experiments for all currently treated quadrature routines, using only this new problem family. All results are stored in the data bank.

Figs. 2 to 10 may be readily constructed from information in the data bank using a program for which the problem family and the selection of routines has to be specified.

At any stage, on the basis of previous results, we may drop from active consideration any routine or any problem family.

We plan to make our results available in the form of reports which can be updated from time to time. These reports will contain principally results in graphical form, like Figs. 2 to 10. Enough information will be included to enable any other interested person to repeat some or all of the experiments and confirm (or discredit) our results. A short section in each report will indicate our own conclusions from the results.

One of the motivations for this organisation is that it provides scope for growth. We do not believe that our choice of problem families is particularly enlightened, nor do we believe that the best routines have been written. But this does not necessitate delaying the evaluation process. It merely implies that evaluation is a continuing process and that current conclusions are liable to be modified or altered. The organisation of this project allows for this sort of change.

6. The role of ε_{quad}

In this paper we have suggested that the quantity $v(E_{quad}(s; \varepsilon_{req}))$ where $s = \phi(\varepsilon_{req}, E_{quad})$ should be used as a measure of the cost of using a routine and that evaluation should be based on this measure. The motivation for this choice is described in earlier sections, and the nature of the results is illustrated in the figures. In this section we discuss some of the implications of this choice.

The perceptive user will have noticed that a major implication is that the role of ε_{quad}, the input tolerance parameter, has undergone a subtle change. Certainly the effect of reducing the value of ε_{quad} is to produce on average a more accurate result at an average greater cost. However, ε_{quad} should no longer be considered as representing the required accuracy unambiguously. It represents this only in some statistical sense. When dealing with a specified problem family, once a value of the success probability s and the required accuracy ε_{req} is specified, the routine may be tuned by choosing the proper value of ε_{quad} from a plot such as the one illustrated in Fig. 2.

The role of EPQUAD as a tuning knob is illustrated by the following circumstance. As remarked in CAQR Section 2, one may take any routine and modify it by making its practical convergence criterion more stringent. By so doing, the modified routine is more reliable and more expensive to use. We consider now a case where this modification takes the form of replacing ε_{quad} internally by $10^{-4}\varepsilon_{quad}$ and we disregard the effect of any physical limit criterion. Thus the behaviour of the modified routine with $\varepsilon_{quad} = \varepsilon$ is identical with the behaviour of the unmodified routine with $\varepsilon_{quad} = 10^{-4}\varepsilon$. If one carries through the steps in the derivation of $v(E_{quad}(s, \varepsilon_{req}))$, one finds that E_{quad} for the modified routine is larger than E_{quad} for the original routine by a factor of 10^4, but that $v(E_{quad}(s, \varepsilon_{req}))$ is identical for both routines. Consequently, this method of evaluation does not distinguish between two routines which differ only in the calibration of EPQUAD. Such a change corresponds to physically unscrewing a tuning knob and attaching it to the shaft at a different angle.

In a previous paper, one of us suggested just such a change in calibration for the Adaptive Simpson Routine (Lyness, 1969, Modification 1, p. 488). We still believe such a change to be an improvement. However, the evaluation method proposed in this paper is insensitive to this change.

For a user to take full advantage of information such as that displayed in Figs. 2 to 10, he has to assign a confidence level s and use the proper value of $\varepsilon_{quad} = E_{quad}(s, \varepsilon_{req})$ to obtain accuracy ε_{req} with this particular confidence. It is not uncommon for a user to insert a 'safety factor' into ε_{quad} quite blindly and information of the type provided here might help him to do this in a less haphazard manner. However, many users are not prepared to go to this trouble. The question arises then as to what use, if any, the information in Figs. 3 to 10 is to a user who does not intend to tune his routine. A basic result which provides an answer to this question is embodied in Theorem (6.8) below. However, the proof of this theorem requires some assumptions about the set of distribution functions which we discuss next.

All individual statistical distribution functions have a non-negative gradient by definition, i.e.

$$\phi(\varepsilon + \Delta, \varepsilon_{quad}) \geq \phi(\varepsilon, \varepsilon_{quad}) \text{ for all } \Delta > 0 \,. \qquad (6.1)$$

The set of distribution functions illustrated in Fig. 1 appear to have an additional property; the curves corresponding to different values of ε_{quad} do not intersect but are arranged in order, the ones with smaller values of ε_{quad} lying to the left and the function $v(\varepsilon_{quad})$ is monotonically non-increasing with ε_{quad}.

Definition:
The set of distribution functions $\phi(x, \varepsilon_{quad})$ are termed *regular* if both

$$\phi(x, \varepsilon_{quad} + \Delta) \leq \phi(x, \varepsilon_{quad}) \text{ for all } x, \Delta > 0 \qquad (6.2)$$

and

$$v(\varepsilon_{\text{quad}} + \Delta) \leq v(\varepsilon_{\text{quad}}) \text{ for all } \Delta > 0 \ . \quad (6.3)$$

In some cases such as problem families involving highly oscillatory integrands we have found examples of non-regular sets of distribution functions. The theorem we are about to establish requires that all distribution functions be regular.

It follows from (6.2) and (4.1) that $E_{\text{quad}}(s + \Delta, \varepsilon_{\text{req}}) \leq E_{\text{quad}}(s, \varepsilon_{\text{req}})$. Thus an application of (6.3) gives

$$v(E_{\text{quad}}(s + \Delta, \varepsilon_{\text{req}})) \geq v(E_{\text{quad}}(s, \varepsilon_{\text{req}})) \text{ for all } \Delta \geq 0 \ . \quad (6.4)$$

We are concerned with comparing two different routines, routine A and routine B. For routine A we define $\phi^A(\varepsilon_{\text{req}}, \varepsilon_{\text{quad}})$, $v^A(\varepsilon_{\text{quad}})$ and $E^A_{\text{quad}}(s, \varepsilon_{\text{req}})$ to be functions defined in (3.2), (3.3) and (4.1). Similarly for routine B we define functions $\phi^B(\varepsilon_{\text{req}}, \varepsilon_{\text{quad}})$, $v^B(\varepsilon_{\text{quad}})$ and $E^B_{\text{quad}}(s, \varepsilon_{\text{req}})$. If routines A and B are among the three used in the illustrations, the curves plotted in Fig. 2 include $E^A_{\text{quad}}(s, \varepsilon_{\text{req}})$ and $E^B_{\text{quad}}(s, \varepsilon_{\text{req}})$ and the curves plotted in Fig. 3 include $v^A(E^A_{\text{quad}}(s, \varepsilon_{\text{req}}))$ and $v^B(E^B_{\text{quad}}(s, \varepsilon_{\text{req}}))$.

We now compare a situation in which the user requires an accuracy ε_{req} and employs on one hand routine A with $\varepsilon_{\text{quad}} = \varepsilon_A$ and on the other hand routine B with $\varepsilon_{\text{quad}} = \varepsilon_B$. The *subscripts* refer to a particular value used in a particular experiment. The *superscripts* indicate a function depending on the routine used. We suppose that, using routine A with $\varepsilon_{\text{quad}} = \varepsilon_A$, one obtains accuracy ε_{req} with confidence level s_A, i.e.

$$s_A = \phi^A(\varepsilon_{\text{req}}, \varepsilon_A) \text{ or } \varepsilon_A = E^A_{\text{quad}}(s_A, \varepsilon_{\text{req}}) \ . \quad (6.5)$$

When using routine B, with $\varepsilon_{\text{quad}} = \varepsilon_B$ one obtains accuracy ε_{req} with confidence level s_B, i.e.

$$s_B = \phi^B(\varepsilon_{\text{req}}, \varepsilon_B) \text{ or } \varepsilon_B = E^B_{\text{quad}}(s_B, \varepsilon_{\text{req}}) \ . \quad (6.6)$$

We recall that, in a plot such as that in Fig. 3, we should broadly regard routine A as better than routine B if the curve $v^A(E^A_{\text{quad}}(s, \varepsilon_{\text{req}}))$ is below the curve $v^B(E^B_{\text{quad}}(s, \varepsilon_{\text{req}}))$. This is the condition (6.7) of the following theorem.

Theorem:
Given two routines A and B whose distribution functions are regular, given arbitrary positive values ε_{req}, ε_A and ε_B, with s_A and s_B as defined above, and given

$$v^A(E^A_{\text{quad}}(s, \varepsilon_{\text{req}})) < v^B(E^B_{\text{quad}}(s, \varepsilon_{\text{req}})) \text{ for } s = s_A \quad (6.7)$$

then one or both of the following inequalities is satisfied.

(a) $s_A > s_B$

(b) $v^A(\varepsilon_A) < v^B(\varepsilon_B)$. (6.8)

Proof:
We show that if (a) is violated, (b) is true. We suppose that

$$s_B = s_A + \Delta \quad \Delta > 0 \quad (6.9)$$

and apply regularity condition (6.4) to routine B, setting $s = s_A$. This gives

$$v^B(E^B_{\text{quad}}(s_B, \varepsilon_{\text{req}})) \geq v^B(E^B_{\text{quad}}(s_A, \varepsilon_{\text{req}})) \ ; \quad (6.10)$$

setting $s = s_A$ in (6.7) gives

$$v^A(E^A_{\text{quad}}(s_A, \varepsilon_{\text{req}})) < v^B(E^B_{\text{quad}}(s_A, \varepsilon_{\text{req}})) \ . \quad (6.11)$$

In view of (6.5) and (6.6), this pair of inequalities imply inequality (6.8) (b) which establishes the theorem.

Note:
The condition that (6.7) is satisfied for $s = s_A$ may be replaced by the condition that (6.7) is satisfied for $s = s_B$ in the conditions for the theorem without invalidating the result.

The purport of this theorem is illustrated by the following example. A user whose program involves integrating functions which resemble (3.1) may be using routine B (ROMBERG) quite successfully. In view of Figs. 3, 4 and 5 he might be convinced that routine A (ANC4) is more appropriate. He may well ask what the immediate effect of switching to routine A might be.

The immediate effect could be disastrous, unless he chooses a suitable value of ε_A for routine A. However, one may make the following assertions which are independent of the value of ε_A used.

If the average cost remains the same, then the quality of the approximation is better using A. If the average quality of the approximation remains the same, the average cost is less using A. The change cannot adversely affect both the cost and the quality, but it will either improve the quality, or reduce the cost, or both. On the other hand, switching from routine A to routine B is certain to affect either the cost or the quality adversely and may affect both adversely.

Information of this type may also be useful to a user who is constructing his program and has to choose a quadrature routine. Normally one carries out a certain amount of tuning in any case. He is in a position to start with the more appropriate routine.

However, it must be emphasised that all this depends on the conditions of the theorem being satisfied. Thus the statistical distribution curves for both routines must be regular and (6.7) must hold for a value of s (s_A or s_B) which is unknown to the user. In practice it is not usually feasible to verify rigorously that these conditions are satisfied.

A set of results, such as the ones illustrated here, could well provide information helpful to subroutine library selection. It is beyond the scope of this paper to discuss this question in any detail. But we are quite ready, as a consequence of the results reported in this paper, to recommend to our librarian that the ROMBERG routine should be removed. Reinstatement of this routine should be considered only if a problem *family* for which it is efficient when compared with the other routines is discovered.

7. Concluding remarks

The method for evaluating quadrature routines described in this paper is, of course, simply one of many conceivable methods. A technique to evaluate methods for evaluating quadrature routines is beyond the scope of this paper. We state in this section what we consider to be some of the advantages and disadvantages of this method.

Advantages

1. The quantities on which the decisions are based are mathematically defined and can be recalculated. It is a repeatable experiment.
2. Once a problem family has been selected, there is apparently no bias in the treatment. If a routine does badly for a specified problem family, there is no defence along the lines that an unlucky choice of integrand functions was responsible.
3. The results are realistic in the sense that they relate to a 'likelihood of failure'. There is no implication that a routine can or should be completely reliable.
4. The results are in a convenient form for one to select an appropriate routine for a particular problem. They are problem oriented.
5. The conclusions up to this point are compatible with common experience.
6. It is possible to add routines and problem families and so build on currently available results.

Defects

1. If the problem families are known it is possible to 'rig' a routine to do all the integrations exactly using perhaps a dozen function values.

2. The choice of problem families is a subjective element which remains in this evaluation procedure.
3. It is a relatively expensive procedure.
4. To obtain full benefit, the user has to 'tune' the value of ε_{quad}.
5. Only accuracy and economy are tested; i.e. warning messages, etc. are disregarded.

Acknowledgement

We should like to acknowledge assistance from Mr. Kevin Karplus of Stanford University in designing the graphical output.

References

CASALETTO, J., PICKETT, M., and RICE, J. R. (1969). A Comparison of some Numerical Integration Programs, *SIGNUM*, Vol. 4, No. 3, pp. 30-40.

KAHANER, D. K. (1971). Comparison of Numerical Quadrature Formulas, pp. 229-259, *Mathematical Software*, J. R. Rice, Ed., Academic Press.

LYNESS, J. N., and KAGANOVE, J. J. (1976). Comments on the Nature of Automatic Quadrature Routines, *ACM Trans. on Math. Soft.*, Vol. 2, pp. 65-81.

LYNESS, J. N. (1969). Notes on the Adaptive Simpson Quadrature Routine, *JACM*, Vol. 15, pp. 483-495.

A historic computer film

In 1951 a film was made in the Mathematical Laboratory at Cambridge University illustrating the operation of the EDSAC which had then been working since May 1949. The film was originally shown at the First Joint Computer Conference held in Philadelphia in December 1951 and is believed to be the first film describing a stored program computer to be made. Among the computer pioneers who took part in it were A. S. Douglas, S. Gill, and E. N. Mutch.

The film has now been re-issued with an introduction and commentary recorded in 1976 by M. V. Wilkes. It is a 16 mm film, is in colour, and runs for approximately 10 minutes. Copies are obtainable at a cost of £95 + VAT from the Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge CB2 3QG. There is a special discount for British Universities and similar bodies.

European Federation for Medical Informatics—Cambridge Congress Call for Papers

The Medical Specialist Groups of The British Computer Society have linked with ten other European medical computing societies to form the European Federation for Medical Informatics. The objectives of the Federation are to promote, throughout Europe, research, international co-operation and information exchange, and high standards of education in the application of information processing theory to medicine and health care delivery.

The first congress of the Federation is to be held in Cambridge, England from 4-8 September 1978. Conference sessions, including practical demonstrations, lectures by industrial participants, and teach-ins, will be held in lecture theatres of the University, with accommodation provided in Churchill College.

The theme of the congress will be 'reporting on practical experience gained' and a 'Call for Papers' has been issued recently. If you feel that you have something to say, and you have missed the Call for Papers, then you can obtain copies from

Dr. B. Barber
Management Services Division,
North East Thames Regional Health Authority,
St. Faith's Hospital
London Road
Brentwood
Essex

The closing date for submissions is not until 1 September 1977.