Short communication

PROPOSAL FOR POPULATION STUDIES IN NUMERICAL QUADRATURE

Hans J. Stetter, Christoph W. Ueberhuber (*)

1. INTRODUCTION

As John Rice pointed out in his paper [1] , there is an urgent need for a systematic study of the population of integrands that arise in numerical quadrature in "real life". Such a study should be the basis for a realistic evaluation of existing and proposed quadrature routines.

Another important question has hardly been investigated : The interaction between calling sequences and user habits (e.g. is a quadrature routine with a short parameter list preferred to one with a longer parameter list; what parameters do the user prefer; etc.).

These questions are without answer not only for quadrature, but practically for all fields of applied numerical analysis.

We have started a project concerning these problems in May 1975 at the Technical University of Vienna. Because of the long-range character of a study of thise type some time will elapse until definite results are available.

The intention of this note is to stimulate further investigations in this subject area : For this purpose, we present a list of proposals, together with our own realisation.

Our present project is a pilot project in the context of the activities of Working Group 2.5 (Numerical Software) of the International Federation of Information Processing (IFIP).

2. GENERAL STRATEGY

2.1. Domain of the study

At first the domain of the study has to be fixed. Our present study has been restricted to "general purpose" quadrature routines for one-dimensional problems. "Special purpose" routines (e.g. routines for highly oscillating integrands, tabulated integrands. indefinite integrals,...) and routines for multidimensional problems may be considered in a later project' The programming language for our study is FORTRAN IV; single precision on a CDC Cyber 74(= 48 bit mantissa) is used.

Our survey is (at present) restricted to the community of users of the CDC Cyber 74 of the Computer Center at the Technical University of Vienna. We hope that it may be extended to other Computer Centers and other user groups.

2.2. Integrand populations

There are basically two possibilities to obtain statistical data on integrand populations. On the one hand, one can ask computer users what sort of functions they have to integrate (if they need numerical quadrature at all). On the other hand, one can register automatically each call to quadrature routines in the library and try to classify the integrand according to information which becomes available during the execution of the quadrature routine . This information can be stored away for later scrutiny. We have chosen an automatic monitoring system following the second of these possibilities.

A disadvantage of this approach in the bias according to the restricted user population that employs routines from a certain subprogram library (no random sample from the whole user community).

Therefore an additional random sample seems to be necessary in a later stage of such a project to eliminate the effect of this sort of nonresponse, introduced by users who write their own quadrature routines or utilize other sources of software.

The effect of "overweighting,, single users with excessive numbers of calls (e.g. Monte Carlo studies) is easily corrected if a user identification is registered too.

(*) Institute for Numerical Mathematics, Technical University of Vienna, Gusshausstrasse 27-29, A-1040 Wien, Oesterreich

2.3. User habits

The users may be supplied with different possibilities to call essentially the same quadrature routine (cf. section 4.1.).
Any call to one of these subprograms may be subject to automatic monitoring, whereby data are collected which should allow a certain characterization of user habits.

3. MONITORING OF INTEGRAND POPULATIONS

3.1. Choice of quadrature routine

Automatic monitoring as described in 2.2. must be based on a high quality quadrature routine which produces data usable for a classification of the integrand.
Concerning the quality of the routine we considered mainly two programs as being attractive :

CADRE of Carl de Boor [2] and
AIND of Robert Piessens [3], [4].

Although AIND has produced better test results (with the set of 50 test-integrands of Casaletto, Picket and Rice [5] as well as with Kahaner's 21 test integrands [6]). CADRE has been chosen as the basic routine for our study because of its ability to recognize peculiarities of the integrand (e.g. integrable singularities, jumps).

3.2. Choice of monitor variables

The following list contains a proposed set of quantities which should permit a classification of the integrand in a quadrature subroutine call :

a) Scaling of the problem :
Length of the integration interval;
Minimum and maximum function value;
Computed value of the integral.

b) Necessity of the adaptive facilities :
Minimum and maximum steplength used;
Minimum and maximum order of the "cautious", Romberg extrapolations.

c) Smoothness of the integrand :
Minimum and maximum order of the "cautious" Romberg extrapolations (see above) together with the required accuracy; Number of recognized linear sections;
Singularities :
Only the number of recognized jumps and singularities of the type x^{α} is counted. If the singularities are endpoint-singularities the respective values of α are registered, too. The number of jumps with a jumpheight greater than (required accuracy)/abs(b-a) is counted separately (to eliminate some of the erroneously registered jumps).

d) Difficulties :
Number of sections, where CADRE could not classify the behaviour of the integrand (e.g. because of noisy integrands,...); Error diagnostics (e.g. too small storage arrays,...).
In our own project, the values of all these quantities are registered automatically with each call of our special routines (see next section).

4. MONITORING OF USER HABITS

4.1. Interface programs

The implementation of one and the same quadrature routine with different calling sequences and different levels of user convenience was achieved by the use of *interface programs* : The user does not call CADRE but a routine which calls CADRE for him and at the same time sets parameters which he did not want (or did not know how) to set.
The interface program with the greatest simplicity (and the least possibilities) for the user is AQUA1 which is called as follows :

CALL AQUAI1 (F,A,B. RESULT,IER)

Here the user has to supply only the integrand FUNCTION F and the limits A and B to obtain a RESULT as well as an indicator IER that something went wrong (IER = 1), or that no error condition occured (IER=0). The remaining parameters of CADRE are set inside AQUA1 : a relative accuracy of 1E-6 is requested and at most 5000 function evaluations are permitted.
The other three interface programs AQUA2, AQUA3 and AQUA4 have a built-in optional printout facility that may be switched off; this facility could be particularly useful in the testing phase. (Our strategy was to eliminate printing from CADRE - this modified version was called QUADRA - and put it into the interface programs).
AQUA2 has the same calling sequence as AQUA1 plus a print-control-parameter IPRINT.
In AQUA3, the user may specify a required (relative) accuracy in decimal digits.
AQUA4 provides the full parameter set of QUADRA (i.e. CADRE without printing), with an additional printing facility.
Furthermore , the user may choose to call QUADRA immediately. The interface programs were designed to be user modifiable, whereas QUADRA (or CADRE) itself cannot be considered as easily user modifiable.

4.2. Choice of monitor variables

First of all, we register which interface program was called .Then the parameter values of the QUADRA-call are registered in full extent, regardless of whether they were set in an interface program or by the user himself. The values of the indicator-variables IER and IFLAG are recorded as well.

5. CONCLUSION

Since May 1975 the whole set of programs described above has been available for the user community at the Technical University of Vienna. Analyses of the recorded data will be published as soon as they become available.
One has to bear in mind, that our study is restricted to a special group of computer users (users of a large scale digital computer at a technical university) and to a rather special choice of monitor-variables. (There would also be other monitor-variables of interest, e.g. the computing time used for one call. This would help, to decide the question if it is better to minimize the number of function evaluations or the overhead of a routine). It would be very important to start similar projects at other places to gain a sound basis for the development of generally applicable test- and evaluation-criteria for quadrature routines and other areas of numerical software.
For persons interested in this topic, our whole material can be made available. Comments and advice are welcome.

LITERATURE

1. John R. Rice, "A metalgorithm for adaptive quadrature", J. ACM Vol. 22 (1975), pp. 61-82.
2. Carl de Boor, "CADRE : An algorithm for numerical quadrature", in John R. Rice (Ed.). "Mathematical software", Academic Press, New York, 1971, pp. 417-449.
3. Robert Piessens, "An algorithm for automatic integration", Angewandte Informatik Vol. 9 (1973), pp. 399-401.
4. Robert Piessens, "A quadrature routine with round-off error guard", Report TW 17, Appl. Math. and Programming Division, Katholieke Universiteit Leuven, 1974.
5. J. Casaletto, M. Picket, J. Rice, "A comparison of some numerical integration programs", SIGNUM Newsletter Vol. 3 (1969), pp. 30-40.
6. David K. Kahaner, "Comparison of numerical quadrature formulas", in John R. Rice (Ed.). "Mathematical software", Academic Press, New York, 1971, pp. 229-259.