A PACKAGE FOR TESTING MULTIPLE INTEGRATION SUBROUTINES

Alan Genz
Computer Science Department
Washington State University
Pullman, WA 99164-1210
U. S. A.

1. INTRODUCTION

During the years 1960-1980 a significant number of methods were proposed for the approximate calculation of multiple integrals (see the books by Stroud [11] and Davis and Rabinowitz [4]). Usually the publication of the description of one of these methods was accompanied by results from a few selected test problems. Users of these methods discovered that although good results could occasionally be obtained in a short time with a particular method, it was often the case that it took hours of computer time to produce inaccurate results. The original description of the methods usually provided only limited information about the practical range of application of the method, and it was not clear if there could ever be any practical means of testing the different methods.

During the same period, standards were developed for the testing of one dimensional integration software and highly robust and reliable FORTRAN subroutines were produced. One of the key aspects of the careful testing of one dimensional integration software was the use of the performance profile method. This was developed in a series of papers by Lyness [7] and Lyness and Kaganove [8,9]. An important feature of the performance profile method is the use of families of test integrands. The test families replace the single integrals that had been previously used with battery tests. A series of sample integrands are drawn from these families, and the results obtained when an integration subroutine is applied to the sample integrands are used to produce performance statistics for a specific family. The statistics can be plotted in various ways to produce different kinds of performance profiles, and the profiles for different integration routines can be used to compare the software.

A paper by the present author [5] contained a description of an extension of the performance profile method to the testing of multidimensional integration subroutines. In the rest of this paper, we briefly describe the software package that was produced for that paper, summarize some of the results of that paper and report on some recent uses of the package.

2. THE TEST PACKAGE

The test package was designed for testing subroutines that have a FORTRAN calling sequence in the form

```
CALL INTGRL(N, REGION, FUNCTN, EPS, ACCUR, VALUE, MAXVLS, ...),
```

P. Keast and G. Fairweather (eds.), Numerical Integration, 337–340.
© 1987 by D. Reidel Publishing Company.

for an N dimensional integral with integrand FUNCTN over a region defined by REGION (which might be a set of parameters or a subroutine). For subroutines of this type the user supplies a requested accuracy EPS and a maximum allowed number of integrand evaluations MAXVLS, and the subroutine should return an integral value VALUE with estimated accuracy ACCUR. The test package that was developed assumed that the region was a hyper-rectangular region and the errors were relative ones. Minor changes to the package would allow it to be used for other regions and/or integration subroutines that worked with absolute errors.

A necessary part of a software testing package that is based on the performance profile method is a collection of test problem families. In order to help distinguish between strengths and weaknesses of the tested software each family should have a specific attribute that distinguishes the family from the other families. The families used in the package were as follows:

Integrand Family	Attribute
$f_1(\mathbf{x}) = cos(2\pi u_1 + \sum_{i=1}^{n} a_i x_i)$	*Oscillatory*
$f_2(\mathbf{x}) = \prod_{i=1}^{n} (a_i^{-2} + (x_i - u_i)^2)^{-1}$	*Product Peak*
$f_3(\mathbf{x}) = (1 + \sum_{i=1}^{n} a_i x_i)^{-(n+1)}$	*Corner Peak*
$f_4(\mathbf{x}) = exp(-\sum_{i=1}^{n} a_i^2 (x_i - u_i)^2)$	*Gaussian*
$f_5(\mathbf{x}) = exp(-\sum_{i=1}^{n} a_i \lvert x_i - u_i \rvert)$	*C^0 Function*
$f_6(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 > u_1 \text{ or } x_2 > u_2 \\ exp(\sum_{i=1}^{n} a_i x_i) & \text{otherwise} \end{cases}$	*Discontinuous*

The integration region for these test families is the unit n-cube. The **u** parameters are parameters that should not affect the difficulty of the integration problem as long as $0 \le u_i \le 1$, so these parameters can be varied randomly in order to generate sample test integrals. The **a** parameters are assumed to be positive numbers and their overall size should affect the degree of difficulty of integrals in a particular problem family. Increasing $\| \mathbf{a} \|$ will in general make the integrands more difficult. In order to fix the difficulty level for a series of tests, the package uses a scaling which is given by

$$n^{e_j} \sum_{i=1}^{n} a_i = h_j .$$

Here j indexes the integrand family, and the numbers n, e_j and h_j are fixed for a series of tests. For each test in the series an n-vector $\mathbf{a}'$ is computed with random components chosen from [0,1]. It is then scaled by a constant c so that $\mathbf{a} = c\,\mathbf{a}'$ satisfies the scaling equation. The integrand families were chosen so that they could be easily evaluated analytically. This is important for testing multiple integration algorithms, which can often take hours of computer time to produce results with unreliable error estimates.

The test package was written in Standard FORTRAN and includes a main subroutine MULTST which calls various supporting subroutines, including the subroutine that is being tested. MULTST takes as input the name of the subroutine to be tested, a set of j values to select test families, **h** and **e** vectors associated with the families, a set of values of n, a sample size S, a limit on the integrand calls allowed for each call of the tested subroutine and a requested relative accuracy for the tested subroutine. For each j and value of n, MULTST makes S calls of the tested subroutine. For each of these calls the **u** and **a** parameters are randomly selected and the correct value for the sample integral is computed. While the S tests are being carried out MULTST accumulates results of the actual and estimated errors for each test, and the actual number of integrand calls required by the tested subroutine for each test. When all of the tests have been completed for each test family and each value of n, MULTST prints a table of results for that series of tests. For each family and value of n the table gives confidence intervals for the median of the estimated errors, actual errors and wrong digits. The number of wrong digits is defined as the difference between the number of estimated correct decimal digits and the actual correct decimal digits when this difference is positive for a particular test. If the difference is negative then zero is used. The test result table also includes a reliability ratio and a median for the actual number of integrand calls for each test family and value of n. The reliability ratio is the the number of times the tested routine had zero wrong digits divided by S.

3. USE OF THE TESTING PACKAGE

For the original report the package was used for a series of tests with three automatic integration methods, a globally adaptive method [6,12], a number theoretic method based on Korobov rules (see Stroud [11]) and a simple Monte-Carlo method. All six test families were used with n = 2, 3, 4, 6, 8. Other input parameters were S = 20, requested accuracy EPS = 10^{-14}, **h** = (110, 600, 600, 100, 150, 100) and **e** = (1.5, 2, 2, 1, 2, 2). The limit on the number of integrand calls was set at 5000, 10000, 20000 and 40000, so that four tables of results were produced for each integration method. In general, the results showed the Monte-Carlo method to be the most reliable but the least accurate of the methods. The adaptive method was reasonably reliable for the first four integrand families and more efficient than the other two methods for n = 2, 3, 4. For n = 6 the number theoretic method reliably produced results that were as good as the adaptive method and for n = 8 the number theoretic method usually produced the best results. While the results were somewhat limited, it seemed clear that a globally adaptive method like the one tested could be reasonably reliable and very much more efficient than the other two methods for moderate size n, as long as the integrands were smooth. For larger values of n the number theoretic method could be much better than a simple Monte-Carlo method.

Since the original tests were carried out the package has been used by the present author to test modifications to the error estimating procedure in the globally adaptive method. Bernsten [1,3] and Bernsten and Espelid [2] have used the package to test different combinations of basic rules for the globally adaptive method for n = 2, 3. They have shown that the use of certain good rule pairs can substantially increase the reliability of this method. A modified version of the package has also been used by the NAG [10] Library integration committee to compare different subroutines for integration over the n-sphere.

4. REFERENCES

[1] J. Bernsten, A Test of the NAG Software for Automatic Numerical Integration over the Cube, Reports in Informatics No. 15, University of Bergen, 1985.

[2] J. Bernsten and T. Espelid, On the Construction of Minimum Point Symmetric Three Dimensional Embedded Integration Rules for Adaptive Integration Schemes, Reports in Informatics No. 16, University of Bergen, 1985.

[3] J. Bernsten, Cautious Adaptive Numerical Integration Over the Cube, Reports in Informatics No. 17, University of Bergen, 1985.

[4] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration,* Academic Press, New York, 1984.

[5] A. C. Genz, Testing Multiple Integration Software, in *Tools, Methods and Languages for Scientific and Engineering Computation,* B.Ford, J-C. Rault and F. Thomasset (Eds.), North Holland, New York, 1984, pp. 208-217.

[6] A. C. Genz and A. A. Malik, An Adaptive Algorithm for Numerical Integration over an N-Dimensional Rectangular Region, J. Comp. Appl. Math., 6(1980), pp. 295-302.

[7] J. N. Lyness, Performance Profiles and Software Evaluation, Argonne National Laboratory Applied Mathematics Division Technical Memorandum 343 (1979).

[8] J. N. Lyness and J. J. Kaganove, Comments on the Nature of Automatic Quadrature Routines, ACM Trans. Math. Software, 2 (1976), pp. 65-81.

[9] J. N. Lyness and J. J. Kaganove, A Technique for Comparing Automatic Quadrature Routines, Comp. J., 20 (1977), pp. 170-177.

[10] Numerical Algorithms Group Limited, Mayfield House, 256 Banbury Road, Oxford OX2 7DE, United Kingdom.

[11] A. H. Stroud, *Approximate Calculation of Multiple Integrals,* Prentice-Hall, New Jersey, 1971.

[12] P. van Dooren and L. de Ridder An Adaptive Algorithm for Numerical Integration over an N-Dimensional Rectangular Region, J. Comp. Appl. Math. 2(1976), pp. 207-217.