

Romberg tables for singular integrands

J. A. Shanks

Oxford University Computing Laboratory, 19 Parks Road, Oxford OX1 3PL

This paper modifies a device of Fox (1967) for constructing the Romberg tables for numerical quadrature for certain classes of singular integrands, and produces the same result with a simpler and more economical technique.

(Received January 1972)

1. Introduction

Fox (1967) introduced an adapted Romberg scheme for accelerating the convergence of simple composite quadrature formulae for the evaluation of integrals with various types of singular integrands. When the error for the quadrature formula $G(h)$, used at interval h to approximate the integral I , is given by

$$G(h) - I = Ah^\alpha + Bh^\beta + Ch^\gamma + \dots, \quad \alpha < \beta < \gamma < \dots, \quad (1)$$

the first column of the Romberg table contains the sequence of approximations $G(h), G(\frac{1}{2}h), G(\frac{1}{4}h), \dots$, each with error $O(h^\alpha)$. Successive columns are found as follows: in the new column corresponding to the elimination of an error term $O(h^\eta)$ the elements have the form

$$\frac{2^\eta F(\frac{1}{2}h) - F(h)}{2^\eta - 1}, \quad (2)$$

where $F(h), F(\frac{1}{2}h), \dots$, are values in the previous column; thus the second column has elements typified by

$$G(h, \frac{1}{2}h) = \frac{2^\alpha G(\frac{1}{2}h) - G(h)}{2^\alpha - 1} \quad (3)$$

with errors $O(h^\beta)$.

If the error is of the form

$$G(h) - I = Ah^\alpha \ln h + Bh^\alpha + Ch^\beta + Dh^\gamma + \dots,$$

Fox remarked that the elimination of the term with constant A from $G(h)$ and $G(\frac{1}{2}h)$ involves the value of h itself, the formula for $\alpha = 2$ being

$$G(h, \frac{1}{2}h) = \frac{(4 \ln h) G(\frac{1}{2}h) - (\ln h - \ln 2) G(h)}{3 \ln h + \ln 2}.$$

He therefore proposed to bypass the second column of the Romberg table, producing the third column directly by simultaneous elimination of the terms with constants A and B by means of a formula, for $\alpha = 2$, given by

$$G(h, \frac{1}{2}h, \frac{1}{4}h) = \frac{1}{9} \left\{ 16G(\frac{1}{4}h) - 8G(\frac{1}{2}h) + G(h) \right\}. \quad (4)$$

This process is inconvenient and it can be avoided if we eliminate the Bh^α term before the term $Ah^\alpha \ln h$. For this the scheme (2) with $\eta = \alpha$ gives

$$G(h, \frac{1}{2}h) - I = Ah^\alpha \left\{ \frac{\ln(\frac{1}{2}h) \times \ln(h)}{2^\alpha - 1} \right\} + C'h^\beta + D'h^\gamma + \dots \\ = B'h^\alpha + C'h^\beta + D'h^\gamma + \dots$$

The troublesome $\ln h$ term has disappeared and the constants B', C', \dots are still independent of h . A second application of (2) with $\eta = \alpha$ then eliminates the $B'h^\alpha$ term and we may proceed in the normal way to eliminate the other terms by using (2) with $\eta = \beta, \gamma, \dots$, to produce successive columns in the table. It is easy to verify that formula (4) is equivalent to two applications of (2) with $\eta = 2$, but the latter fits more easily and conveniently into the standard Romberg scheme and needs no 'special' treatment.

2. Further error terms

This idea is equally applicable for more complicated error terms with repeated occurrence of the $\ln h$ term. For example, with

$$G(h) - I = Ah + Bh \ln h + Ch^2 + Dh^2 \ln h + \dots \quad (5)$$

two applications of (2) with $\eta = 1$, followed by two applications with $\eta = 2$ eliminates all the terms given explicitly in (5).

With

$$G(h) - I = Ah^\alpha + Bh^\alpha \ln h + Ch^\alpha \ln^2 h + Dh^\alpha \ln^3 h \\ + Eh^\beta + Fh^\gamma + \dots,$$

we apply (2) four times with $\eta = \alpha$ to eliminate the first four terms and then proceed in the usual way with $\eta = \beta, \gamma, \dots$.

3. Numerical examples

Two numerical examples illustrate the simplicity of this technique. For

$$I = - \int_0^1 x^{\frac{1}{2}} \ln x \, dx = \frac{4}{9} = 0.444 \dots$$

the repeated trapezoidal rule $T(h)$ has the error expression

$$T(h) - I = Ah^{\frac{3}{2}} + Bh^{\frac{3}{2}} \ln h + Ch^2 + Dh^4 + \dots$$

Successive columns of the Romberg table, after the first, are obtained using (2) with $\eta = \frac{3}{2}, \frac{3}{2}, 2, 4, \dots$, and produce the results of Table 1. From the second column onwards of course

Table 1

$T(1)$	= 0.0000000		
		0.3790948	
$T(\frac{1}{2})$	= 0.2450645	0.4422595	
		0.4199274	0.4444548
$T(\frac{1}{4})$	= 0.3581041	0.4439060	0.4444448
		0.4354283	0.4444454
$T(\frac{1}{8})$	= 0.4080900	0.4443105	
		0.4411702	
$T(\frac{1}{16})$	= 0.4294746		

Table 2

$T(1)$	= 0.0000000		
		0.1110082	
$T(\frac{1}{2})$	= 0.0832562	0.3039582	
		0.2557207	0.3687263
$T(\frac{1}{4})$	= 0.2126046	0.3525343	0.3750253
		0.3283309	0.3734505
$T(\frac{1}{8})$	= 0.2993993	0.3682215	
		0.3582488	
$T(\frac{1}{16})$	= 0.3435364		

Fox's method would produce the same results apart from possibly different rounding errors.

For

$$I = - \int_0^1 x \ln^3 x \, dx = \frac{3}{8} = 0.375,$$

we find

$$T(h) - I = Ah^2 + Bh^2 \ln h + Ch^2 \ln^2 h + Dh^2 \ln^3 h + Eh^4 + Fh^6 + \dots$$

The Romberg columns are obtained from (2) with $\eta = 2, 2, 2, 2, 4, 6, \dots$, and the results are shown in **Table 2**.

Reference

Fox, L. (1967). Romberg integration for a class of singular integrands, *The Computer Journal*, Vol. 10, No. 1, pp. 87-93.

Correspondence

To the Editor
The Computer Journal

Sir

'Packing' in FORTRAN

A commonly occurring problem in FORTRAN mathematical programming is the indexing (or subscripting) of 'sparse' multiple subscripted variables where the use of arrays of three, four or more dimensions is inefficient and prohibitively space consuming. One of the 'traditional' solutions used by quantum chemists and physicists¹—who are often dealing with sparse four subscript quantities—has been to take a leaf from the commercial programmer's book and 'pack' the least significant bits of four or more integers into one real variable, and use this variable as a label. This has always been done via an Assembler routine CALLED from a FORTRAN program. However, modern FORTRAN compilers allow the use of single characters as logical variables and so this packing (and of course 'unpacking') can now be performed entirely in FORTRAN using appropriate EQUIVALENCES. I give below for your amusement annotated program fragments to pack and unpack four integers (0-255) into one real variable. Other character manipulations can be performed in the same way.

```
LOGICAL*1 LOG1(4), LOG2(8)
INTEGER*2 ID(4), I, J, K, L
EQUIVALENCE (WORD, LOG1(1)), (ID(1), LOG2(1))
EQUIVALENCE (ID(1), I), (ID(2), J) (ID(3), K), (ID(4), L)
```

C THESE EQUIVALENCES SET UP THE CHARACTERS

C THE FOLLOWING FOUR STATEMENTS PACK I, J, K, L INTO WORD

```
LOG1(1) = LOG2(2)
LOG1(2) = LOG2(4)
LOG1(3) = LOG2(6)
LOG1(4) = LOG2(8)
```

C THE FOLLOWING FOUR STATEMENTS UNPACK WORD INTO I, J, K, L

```
LOG2(2) = LOG1(1)
LOG2(4) = LOG1(2)
LOG2(6) = LOG1(3)
LOG2(8) = LOG1(4)
```

Yours faithfully,

D. B. COOK

Department of Chemistry
The University
Sheffield S3 7HF
7 June 1972

¹See, for example, 'The POLYATOM System' by I. G. Csizmadia, M. C. Harrison, J. W. Moskowitz, S. Seung, B. T. Sutcliffe, and M. P. Barnett, M.I.T.

To the Editor
The Computer Journal

Sir

The postage stamp problem

With reference to W. F. Lunnon's article (this *Journal*, Vol. 12, 377) I should like to report the solution $V(10, 3) = 155$. The original run using techniques developed from Lunnon's work took 250 Plessey XL4 processor hours, but with further sophistication a re-run was accomplished in only 99 hours. The calculations show a unique set of stamps, 1 2 6 8 19 28 40 43 91 103.

It has been reported to me by M. L. V. Pitteway that $V(8, 4) = 213$. Apparently this ran in approximately 100 low priority hours on a KDF 9; the stamp denominations themselves have unfortunately been mislaid.

Yours faithfully,

J. L. SELDON

Squadron Leader, Royal Air Force
7 June 1972

Downloaded from https://academic.oup.com/comjnl/article/15/4/360/3515061 by guest on 13 April 2024