NRL Memorandum Report 2231

A Modified Simpson's Rule and Fortran Subroutine for Cumulative Numerical Integration of a Function Defined by Data Points

L. V. BLAKE

Radar Geophysics Branch
Radar Division

April 1971

NAVAL RESEARCH LABORATORY
Washington, D.C.

20

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Research Laboratory Washington, D. C. 20390	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A MODIFIED SIMPSON'S RULE AND FORTRAN SUBROUTINE FOR CUMULATIVE NUMERICAL INTEGRATION OF A FUNCTION DEFINED BY DATA POINTS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

A final report on one phase of the problem; work is continuing.

5. AUTHOR(S) (First name, middle initial, last name)

Lamont V. Blake

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1971	20	None

8a. CONTRACT OR GRANT NO. NRL Problem 53R02-55 b. PROJECT NO. RF-151-402-4011 c. d.	NRL Memorandum Report 2231 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Department of the Navy (Naval Ship Systems Command) Washington, D. C. 20360

13. ABSTRACT

Formulas are derived for finding the areas between each pair of points under the second-degree-polynomial curve defined by three equispaced points in an x-y (cartesian) coordinate system. These formulas are a modification of Simpson's numerical-integration rule which gives only the total area lying under the curve between the initial and final points. The formulas, implemented by a Fortran computer subroutine named SIMCUM, are useful in problems where it is necessary to find integrals under a curve defined by a limited number of data points, and the cumulative integral is desired at each data point rather than at every second data point as would be possible with the ordinary form of Simpson's rule. With a fixed number of data points, the method gives improved accuracy, compared with the alternative of using the trapezoidal rule, when the "true" curve is continuous, not a straight line, and is reasonably well defined by the data points. For a specified integration accuracy, a considerable cost saving can often be effected by using this method, instead of the trapezoidal rule with a considerably greater number of data points.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Numerical integration Simpson's rule Fortran integration subroutine						

CONTENTS

ABSTRACT

Formulas are derived for finding the areas between each pair of points under the second-degree-polynomial curve defined by three equispaced points in an x-y (cartesian) coordinate system. These formulas are a modification of Simpson's numerical-integration rule which gives only the total area lying under the curve between the initial and final points. The formulas, implemented by a Fortran computer subroutine named SIMCUM, are useful in problems where it is necessary to find integrals under a curve defined by a limited number of data points, and the cumulative integral is desired at each data point rather than at every second data point as would be possible with the ordinary form of Simpson's rule. With a fixed number of data points, the method gives improved accuracy, compared with the alternative of using the trapezoidal rule, when the "true" curve is continuous, not a straight line, and is reasonably well defined by the data points. For a specified integration accuracy, a considerable cost saving can often be effected by using this method, instead of the trapezoidal rule with a considerably greater number of data points.

PROBLEM STATUS

This is a final report on one phase of the problem; work is continuing.

AUTHORIZATION

NRL Problem R02-55
Project RF-151-402-4011

A MODIFIED SIMPSON'S RULE AND FORTRAN SUBROUTINE FOR CUMULATIVE NUMERICAL INTEGRATION OF A FUNCTION DEFINED BY DATA POINTS

INTRODUCTION

In many scientific and engineering problems, it is desired to perform cumulative numerical integration of a function defined by a limited number of data points. In optical ray tracing, for example, the data points might be the index of refraction at various levels in a layered medium, or other quantities from which the refractive index could be computed. A similar problem is to compute the cumulative power absorption along a ray path. (This was in fact the problem that led the author to develop the method to be described.)* The limitation on the number of points

*It is not known whether the method is new, but it is not described in any texts readily available to the author. Jon Wilson, of the NRL Radar Division Analysis Staff, has pointed out that formulas given in the book "Numerical Integration" by P. J. Davis and P. Rabinowitz (Blaisdell; Waltham, Mass., 1967; pp. 22-23) are similar in principle to the formulas derived here, but they are not there developed to yield the explicit formulas given here. Their formulas are also generalized to apply to nonequally spaced abscissa values, which are discussed later in this report.

may be due to the cost of obtaining experimental data at smaller intervals, or it may equally well be that the points are computed by a lengthy procedure, so that obtaining additional points would require an excessive amount of computer time.

The general nature of the problem in abstract mathematical terms is: Given a function $y(x)$ defined by n data points $y(x_1)$, $y(x_2)$, ... $y(x_n)$ with equally spaced abscissa (x) values, evaluate:

$$I_i = \int_{x_1}^{x_i} y(x)\,dx, \quad i = 2, 3, 4, \ldots n. \tag{1}$$

The simple way to do this is to use the trapezoidal rule of numerical integration and apply the recursion formula:

$$\int_{x_1}^{x_i} y(x)\,dx = \int_{x_1}^{x_{i-1}} y(x)\,dx + \left(\frac{x_i - x_{i-1}}{2}\right)\left[y(x_i) + y(x_{i-1})\right] \tag{2}$$

As is well known, Simpson's rule gives a much more accurate value of the integral than does the trapezoidal rule if a continuous smooth curve through the data points represents the "true" function and if this curve is not a straight line. The trouble with Simpson's rule in this application is that since it requires triplets of points rather than pairs of points for each cumulative step in the integration, values of $\int_{x_1}^{x_i} y(x)\,dx$ can be obtained only for $i = 3, 5, 7, 9, \ldots$ etc. This limitation is an objection to the use of Simpson's rule when the available number of data points is so limited that obtaining the cumulative integral only for odd-numbered data points is unacceptable.

The purpose of this report is to describe a modification of Simpson's rule which overcomes this objection and permits the cumulative integral to be found at each data point.

METHOD

Simpson's rule is basically a formula for numerically finding the area under the parabolic (second-degree polynomial) curve defined by just three points in a plane, the points being separated by equal intervals along the x-axis of an x-y (cartesian) coordinate system. Such a set of points and the corresponding curve are depicted in Fig. 1. The area under this curve, by the

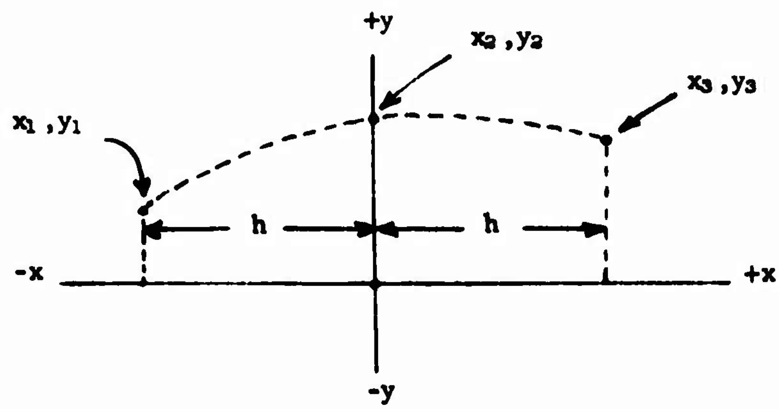

Fig. 1. Triplet of equispaced points in xy coordinate system; second-degree-polynomial curve shown dotted.

ordinary Simpson's rule, is:

$$A = \int_{x_1}^{x_3} y(x)\ dx = \frac{h}{3}\ (y_1 + 4y_2 + y_3) \qquad (3)$$

where $y(x) = ax^2 + bx + c$, with a, b, and c chosen to make y(x) fit the given data points. The quantity h is the separation of successive points on the x-axis; i.e., $h = x_3 - x_2 = x_2 - x_1$.

Since x_1, x_2, and x_3 do not appear explicitly on the right-hand side of Eq. (3), it is permissible in analysis of the matter to set $x_1 = -h$, $x_2 = 0$, and $x_3 = +h$. In these terms, the problem that has been posed is to find the formulas, in terms of h, y_1, y_2, and y_3, that give

$$A_1 = \int_{-h}^{0} y(x)\ dx \tag{4}$$

and

$$A_2 = \int_{0}^{h} y(x)\ dx. \tag{5}$$

The formulas are found to be:

$$A_1 = \frac{h}{3}\left(\frac{5}{4}y_1 + 2y_2 - \frac{1}{4}y_3\right) \tag{6}$$

$$A_2 = \frac{h}{3}\left(-\frac{1}{4}y_1 + 2y_2 + \frac{5}{4}y_3\right) \tag{7}$$

Of course, $A_1 + A_2 = A$, and it is evident that adding the right-hand sides of Eqs. (6) and (7) gives the usual Simpson's rule, Eq. (3).

DERIVATION OF THE FORMULAS

If the curve in Fig. 1 is represented by

$$y(x) = ax^2 + bx + c \tag{8}$$

and (x_1, y_1), (x_2, y_2), and (x_3, y_3) are points on the curve, then

$$a = (y_1 - 2y_2 + y_3)/2h^2 \tag{9}$$

$$b = (y_3 - y_1)/2h \tag{10}$$

$$c = y_2 \tag{11}$$

Eq. (4) in these terms becomes:

$$A_1 = \int_{-h}^{0} y(x)\,dx = \left[\frac{ax^3}{3} + \frac{bx^2}{2} + cx\right]_{-h}^{0} = \frac{ah^3}{3} - \frac{bh^2}{2} + ch \tag{12}$$

Substituting the results of Eqs. (9), (10), and (11) in Eq. (12) gives Eq. (6). A similar procedure starting with Eq. (5) instead of Eq. (4) yields Eq. (7).

FORTRAN IMPLEMENTATION OF THE METHOD

Using these results, the recursion formulas now applicable instead of the trapezoidal formula, Eq. (2), are

$$\int_{x_1}^{x_i} y(x)\,dx = \int_{x_1}^{x_{i-1}} y(x)\,dx + \frac{h}{3}\left[\frac{5}{4}y_{i-1} + 2y_i - \frac{1}{4}y_{i+1}\right] \tag{13}$$

and

$$\int_{x_1}^{x_{i+1}} y(x)\,dx = \int_{x_1}^{x_i} y(x)\,dx + \frac{h}{3}\left[-\frac{1}{4}y_{i-1} + 2y_i + \frac{5}{4}y_{i+1}\right] \tag{14}$$

in which i is an even integer and $h = x_i - x_{i-1} = x_{i+1} - x_i$

The initial (least value) of x is denoted by x_1. Given a set of points (x_i, y_i) defined by Fortran arrays X(I) and Y(I), these recursion formulas may be incorporated into a "modified Simpson's rule" cumulative integration subroutine as follows.

```
      SUBRØUTINE SIMCUM (Y,H,N,AREA)
      DIMENSIØN Y(N),AREA(N)
      AREA(1)=0.
      FAC=H/3.
      TERM1=1.25*Y(1)
      TERM2=-0.25*Y(1)
      K=1
      DØ 1 I=3,N,2
      J=I-1
      TERM3 = 2.*Y(J)
      TERM4=1.25*Y(I)
      TERM5=-0.25*Y(I)
      AREA(J)= AREA(K) * FAC*(TERM1 + TERM3 + TERM5)
      AREA(I) = AREA(J) + FAC*(TERM2 + TERM3 + TERM4)
      TERM1=TERM4
      TERM2=TERM5
      K=I
    1 CØNTINUE
      IF (2*(N/2).EQ.N) 3,4
    3 AREA(N)=AREA(K) + FAC*(-0.25*Y(J) + 2.*Y(K) + 1.25*Y(N))
    4 RETURN
      END
```

The parameter Y is the array of values of y, H is the interval between successive values of X, AREA is the computed array of successive (cumulative) values of the integral, and N is the integer number of values of y in the array. Of course, Y and AREA must be properly dimensioned in the program or subroutine from which SIMCUM is called. Statement 3 provides for calculation of the integral through the leftover last interval if N happens to be an even number.

UNEQUALLY SPACED DATA POINTS

These formulas and the subroutine are based on the assumption that the data points are equally spaced along the independent-variable (x) axis. A virtue of the trapezoidal rule is that it can also accommodate unequally spaced points. Actually, a further modification of Simpson's rule can be made to apply to this case also. The formulas are quite cumbersome, and the author has not derived them in detail. They can be derived by the procedure illustrated by Eq. (12), except that the limits of integration are now x_1 and x_2; an equation analogous to Eq. (6) is thus found. Then the limits are changed to x_2 and x_3, to obtain an equation analogous to Eq. (7). The cumbersomeness arises from the fact that the expressions for a, b, and c of Eq. (8) are much more complicated when the intervals are unequal. In most practical cases, data points are obtained at equal intervals along the x-axis, so that it did not seem worthwhile to derive the complicated formulas for unequal intervals.

It may sometimes happen that a set of data points has spacings that are constant over subintervals but not over the entire interval of the x-axis. In such a case, of course, Subroutine SIMCUM can be successively applied to each subinterval.

A TEST OF THE FORTRAN SUBROUTINE

Subroutine SIMCUM was tested by first defining an array of values of $y(x) = \sin (x)$ for 10 equal subintervals of x over the total interval $\pi/2$. The "correct" result was found by using the analytic integration formula $\int_a^b \sin (x)\, dx = \cos (a) - \cos (b)$.

The cumulative integral was found by using SIMCUM and the result was compared with that of trapezoidal-rule integration (Eq. (2)), for each subinterval. The computations were done using the NRL CDC-3800 computer. The results are shown in the following table:

Subinterval	Correct Result	SIMCUM Result	Trapezoidal Rule Result
1	0.012312	0.012337	0.012286
2	0.048944	0.048944	0.048843
3	0.108994	0.109016	0.108769
4	0.190983	0.190984	0.190590
5	0.292893	0.292912	0.292291
6	0.412215	0.412216	0.411367
7	0.546010	0.546023	0.544886
8	0.690983	0.690985	0.689562
9	0.843566	0.843572	0.841830
10	1.000000	1.000003	0.997943

Thus the cumulative error using SIMCUM is seen to be 3×10^{-6} compared to approximately 2×10^{-3} using trapezoidal-rule integration with the same data points. Of course, this is a special case, but it illustrates the superiority of Simpson's rule when the number of data points is limited, if the "true" function is a smooth curve. A test was also made using the same function divided into 9 subintervals, so that there were 10 data points (an even number). This causes Statement 3 of Subroutine SIMCUM to be utilized. The results were comparable in accuracy to those tabulated above.

A subroutine named SIMRUN is available in the CDC CO-OP Program Library which does a Simpson's Rule integration similar to that of SIMCUM except that it finds the cumulative integral from Eq. (3), and is therefore limited to obtaining values at only the odd-numbered points. In terms of the test of SIMCUM tabulated above, this corresponds to the even-numbered intervals. The test described was also done using SIMRUN, and (as expected) the results were identical to those obtained with SIMCUM at the even-numbered intervals.

APPLICABILITY OF THE METHOD

The method of cumulative integration described is applicable when the following criteria are met:

(1) The function y(x) to be integrated is defined by a discrete set of data points (equally spaced along the x-axis of the coordinate system) rather than by a continuous function; or, it is defined by a "function" that is so complicated that a

lengthy computation is required to calculate a single numerical value of y.

(2) The function represented by the data points, though not necessarily defined analytically, is known to be a function that varies slowly relative to the x-axis separation of the independent-variable values, i.e., the data points define the function fairly well in the sense that a second-degree polynomial is a reasonable interpolating function for any three successive points.

(3) The function is not a straight line, and the desired accuracy is better than that obtained by straight-line interconnection of the successive points (trapezoidal rule).

(4) The cost of the increase in computing time resulting from the use of Subroutine SIMCUM instead of a trapezoidal-rule subroutine is probably small compared to the cost of obtaining a greatly increased density of data points (and also therefore performing a greater number of trapezoidal-rule computations). A means of evaluating this criterion is difficult to specify in general terms, but in most specific cases the answer is likely to be obvious.

The method definitely should not be used when only the integral over the total interval, rather than the cumulative set of integrals, is needed. In that case, the "ordinary" form of Simpson's rule should be used, although a problem is created when the number of data points is even instead of odd. This problem can be solved by using the three last statements of Subroutine SIMCUM at the end of the subroutine. A Simpson's rule subroutine incorporating this feature is given in Appendix A. When the data points are known to define a succession of straight-line segments, or if it is desired to interpret the data as if they do, the trapezoidal rule, Eq. (2), should be used.

Formal documentation of Subroutine SIMCUM has been submitted to the NRL Research Computation Center for possible issuance as a Computer Note or Computer Bulletin.

APPENDIX A

FORTRAN SUBROUTINES FOR TRAPEZOIDAL-RULE AND "ORDINARY" SIMPSON'S RULE INTEGRATION

For convenient reference, a listing of the trapezoidal-rule subroutine written by the author for use in comparing that rule with the modified Simpson's rule is given below. The parameters are identical to those of Subroutine SIMCUM.

```
      SUBRØUTINE TRAPZØID (Y,H,N,AREA)
      DIMENSIØN Y(N),AREA(N)
      AREA(1) = 0.
      H2 = H*.5
      DØ 1 I=2,N
      J = I-1
    1 AREA(I) = AREA(J) + H2*(Y(I) + Y(J))
      END
```

Also given below for convenience is a subroutine using the conventional Simpson's rule in a form suitable for finding the overall integral of a set of data points, rather than the set of cumulative integrals. AREA in this subroutine is not dimensioned. Provision is made for inclusion of the left-over last subinterval when the total number of data points is an even number by using the formula of Eq. (14).

```
      SUBRØUTINE SIMSØN (Y,H,N,AREA)
      DIMENSIØN Y(N)
      SUM = 0.
      J=1
      DØ 1 I=3,N,2
      SUM=SUM + Y(J) + 4.*Y(I-1) + Y(I)
      J=I
    1 CØNTINUE
      IF (2*(N/2).EQ.N) 3,4
    3 SUM = SUM - 0.25*Y(N-2) + 2.*Y(J) + 1.25*Y(N)
    4 AREA = SUM * H/3.
      END
```

This subroutine was tested by integrating $\int_0^{\pi/2} \sin(x)\,dx$, with N = 10 (so that Statement 3 was utilized). The result obtained was AREA = 0.999998460. (As noted previously, the exactly correct result is 1.0.)

AD-723583

NRL Memorandum Report 2231
Change 1, May 25, 1971

From: Naval Research Laboratory
To: Distribution list for NRL Memorandum Report 2231, "A Modified Simpson's Rule and Fortran Subroutine for Cumulative Numerical Integration of a Function Defined by Data Points," by L.V. Blake, April 1971

Please make the following correction:

1. Page 7, in the Fortran listing of subroutine SIMCUM, 13th line, the first asterisk should be a + sign. The statement should read:

```
AREA(J) = AREA(K) + FAC*(TERM1 + TERM3 + TERM5)
```