13.1 Numerical Evaluation of Integrals Over One Dimension

A. Purpose

This collection of subprograms estimates the value of the integral

$$\int_a^b f(x)\,dx$$

where the integrand $f(x)$ and the limits a and b are supplied by the user.

B. Usage

Described below under B.1 through B.5 are:

B.1 Program Prototype, Single Precision.

B.1.a The Calling Routine

REAL A, B, ANSWER, WORK($\geq k_1$) [k_1 depends on options used (≥ 1).]

INTEGER IOPT($\geq k_2$) [k_2 depends on options used (≥ 2).]

Assign values to A and B.

Assign values to IOPT() and elements of WORK() referenced using options (see Section B.3). For simplest usage, set

IOPT(2) = 0

CALL SINT1(A,B,ANSWER,WORK,IOPT)

If the computation is successful (indicated by IOPT(1) < 0) the result is in ANSWER and the estimate of the error in the result is in WORK(1).

B.1.b Argument Definitions for SINT1

A [in] Lower limit of integration.

B [in] Upper limit of integration. The relative values of A and B are interpreted properly. Thus if one exchanges A and B, the sign of the answer is changed. When the integrand is positive, the sign of the result is the same as the sign of B − A.

ANSWER [out] Estimate of the integral.

WORK() [inout] On completion, WORK(1) contains an estimate of the upper bound of the magnitude of the difference between ANSWER and the true value of the integral. Other elements of WORK may be referenced by the option vector IOPT as described below and in Section B.3, and to pass parameters to SINTF as described in Section B.1.e.

IOPT() [inout] Used to return status information to the user, to allow the selection of options, and to pass parameters to SINTF as described in Section B.1.e. IOPT(1) returns a status indicator with the possible values:

−1	Normal termination with either the absolute or relative error tolerance criteria satisfied (see option 3 in Section B.3).
−2	Normal termination with neither the absolute nor relative error tolerance criteria satisfied, but the error tolerance based on the locally achievable precision is satisfied. (See option 3 in Section B.3). This is the normal status value when no tolerances are specified.
−3	Normal termination with none of the error tolerance criteria satisfied. (See option 3 in Section B.3).
+4	Bad value for an element of IOPT.
+5	Too many function values needed (see option 9 in Section B.3).
+6	The integrand apparently contains a nonintegrable singularity. The abscissa of the singularity is near ANSWER. WORK(1) contains a large number.

The remainder of IOPT may be used to select options and to pass parameters to SINTF. If no options are selected, set IOPT(2) = 0.

See Section B.3 for a detailed description of options and the following table for an overview.

Table of Options for SINT1

Option Number	Brief Description
0	No more options.
1	No effect. Reserved for future use. Do not use option 1.
2	Select level of diagnostic output.
3	Specify error tolerances.
4	Specify an a posteriori estimate of the absolute error in the calculated value of the integrand.
5	Specify an a priori estimate of relative error expected in the calculated values of integrands.
6	Use reverse communication.
7	Specify minimum index of quadrature formula.
8	No effect. The parameter may provide information to SINTF.
9	Specify maximum number of integrand evaluations allowed.
10	Return number of integrand evaluations used.
11	Specify location of singularity or discontinuity in integrand.
12	Specify absolute errors in the limits.
13	Used for multi-dimensional integration. See Chapter 13.2.

B.1.c The User-Supplied Integrand Subroutine SINTF

SINT1 requires that the user provide values of the integrand. In the case of simple usage these values are provided by a user-supplied subroutine of the form:

```
SUBROUTINE SINTF(ANSWER, X, IFLAG)
REAL  ANSWER, X
INTEGER  IFLAG
ANSWER = value of integrand at X
RETURN
END
```

B.1.d Argument Definitions for SINTF

ANSWER [out] set by SINTF to be the value of the integrand at X.

X [in] contains the abscissa at which the integrand is to be evaluated. X may also be used to pass extra information into SINTF, as described in Section B.1.e.

IFLAG [in] is not needed for simple usage, but may be used to pass extra information into SINTF, as described in Section B.1.e.

Most mathematical software that requires user defined function information allows the user to pass in the name of a subprogram for evaluating the function. The approach used here has the advantage of not requiring the user to declare his subprogram in an external statement (It's not unusual for this to be forgotten.), and of giving a meaningful diagnostic when no such routine is provided. If one is solving different problems with different programs, one can use different file names for the different function subprograms, all of which would use the same entry name. The linker must then be told which of these function subprograms should be used in forming the executable file. If one wants to solve several different problems in the same program, one should code the separate cases in one subprogram and select the one desired by passing a "case" variable into the routine. This can either be done as part of IFLAG as mentioned above, or can be done through the use of named common. One can also use reverse communication and call subprograms with different names for the different cases.

B.1.e Passing Extra Information into SINTF

The argument X of SINTF is the first element of the WORK vector passed from the user's calling routine to SINT1. Elements of WORK other than the first may be referenced by options, as described in Section B.3, or used to pass extra floating point information to SINTF. If X is used in this way it must be declared to be an array instead of a scalar.

The argument IFLAG of SINTF is the first element of the IOPT() vector passed from the users calling routine to SINT1. Elements of IOPT (after the first) that are not used for options as described in Section B.3 may be used to pass integer valued information into SINTF. In addition, the parameter of option 8 described in Section B.3 may be examined by SINTF. If IFLAG is used in this way it must be declared to be an array instead of a scalar.

B.2 Program Prototype, Single Precision, Reverse Communication.

B.2.a The Calling Routine

REAL A, B, ANSWER, WORK(≥ 1)

INTEGER IOPT(≥ 3)

Assign values to A, B, IOPT and elements of WORK referenced by options. Option 6 must be selected (see Section B.3). For simple usage, set

IOPT(2) = 6
IOPT(3) = 0

CALL SINT1 (A, B, ANSWER, WORK, IOPT)

```
DO
```

$\boxed{\textbf{CALL SINTA (ANSWER, WORK, IOPT)}}$

```
    IF (IOPT(1) .NE. 0) EXIT
    ANSWER = value of integrand at $x$, where
        $x$ = WORK(1)
END DO
{ Integration is complete. }
```

B.2.b Argument Definitions

A, B Used as described in Section B.1.b.

ANSWER [inout] During integration, ANSWER provides a value of the integrand to SINTA. On completion, ANSWER contains an estimate of the integral.

WORK() [inout] Used as described in Section B.1.b, except that during the integration WORK(1) is the abscissa at which the integrand is to be evaluated.

IOPT() [inout] Used as described in Section B.1.b, except IOPT(1) has additional uses. If IOPT(1) = 0, then evaluate the integrand at the abscissa specified in WORK(1) and put the result in ANSWER. If IOPT(1) $\neq$ 0, integration has been completed. Refer to Section B.1 for the meaning of nonzero values of IOPT(1).

B.3 Methods to Request Unusual Usage Through the Arguments IOPT and WORK

Options are identified by codes ranging from 0 to 12. Options 0 and 6 have no arguments, while the other options each require one integer argument. An option is selected by placing its code in the array IOPT() with the following element of the array containing its argument if it requires one. The option codes, with space for their arguments as required, must be stored as a contiguous sequence beginning in IOPT(2) and ending with the option code 0.

If a nonzero option code appears more than once, only the last occurrence will be effective.

Depending on the option, an argument may be a location where an integer value is provided or returned, or it may be an index into the array WORK() indicating the location of a floating point datum or the beginning of a sequence of floating point data.

If options involving indexes into WORK() are used, the index must exceed 1, and WORK() must be dimensioned sufficiently large.

All unselected options are assigned default values.

The options are defined as follows:

0 (No argument) Terminates the sequence of selected options.

1 (Argument K1) No effect. Reserved for future use. Do not use option 1.

2 (Argument K2) K2 selects the level of diagnostic output:

  0 No printing.
  1 Minimum printing — error messages (default value).
  2 Panel boundaries and error estimates in addition to (1).
  3 Error estimates for each quadrature formula in addition to (2).
  4 Maximum detail.

3 (Argument K3) K3 is an index into WORK(). The user provides the absolute error tolerance in WORK(K3), the error tolerance T based on the locally achievable precision in WORK(K3+1), and the error tolerance relative to the value of the integral in WORK(K3+2). The error tolerance T should be assigned a value ϵ^f, where ϵ is the smallest number such that computing $1.0 + \epsilon$ gives a number different from 1.0 (ϵ is the precision of the machine arithmetic) and f is the fraction (number of correct digits desired) / (digits of achievable precision). (This latter number is discussed in Section D where the method for computing the error estimate is described.) The internal value for T is bounded between ϵ and $\sqrt{\epsilon}$, that is $\frac{1}{2} \leq f \leq 1$. If this option is not selected, the absolute error tolerance and the tolerance relative to the value of the integral are both set to zero, and T is set using an f of 3/4, *i.e.* T = $\epsilon^{3/4}$. The default values usually provide all the accuracy that can be obtained efficiently.
The error tolerance relative to the value of the integral is applied globally (over the entire region of integration) rather than locally (one step at a time). This policy provides true control of error relative to the value of the integral when the integrand is not sign definite, as well as when the integrand is sign definite. To apply the criterion of error tolerance relative to the value of the integral, the value of the integral over the entire region, estimated without refinement of the region, is used to derive an absolute error tolerance that may be applied locally. If the preliminary estimate of the value of the integral is significantly in error, and the least restrictive error tolerance is relative to the value of the integral, the cost of computing the integral will be larger than the cost of computing the integral to the same degree of accuracy using appropriate values of either of the other error tolerance criteria. The preliminary estimate of the integral may be significantly in error if the integrand is not sign definite or has large variation.

4 (Argument K4) In WORK(K4) the user provides an a posteriori estimate of the absolute value of the error committed while evaluating the integrand. This value may be computed during evaluation of the integrand. The default value is zero.

5 (Argument K5) In WORK(K5) the user provides an a priori estimate of the relative error expected to be committed while evaluating integrands. Changes to this value are not detected during evaluation of the integral. The default value is the machine precision.

6 (No argument) Selects use of reverse communication. See Section B.2.

7 (Argument K7) K7 specifies the minimum index of quadrature formula used before SINTA does a more detailed analysis to determine whether a higher order formula should be used or the interval subdivided. The default value for this minimum index is 3. The number of integrand samples used for a formula of index k is $2^k - 1$, and the algebraic order is 3/2 the number of samples. If the user knows from analysis or experience with this program that an integral can only be computed to the desired accuracy by using a formula of higher index than 3, some efficiency may be gained by so notifying the program.

8 (Argument K8) No effect. K8 may be used to pass information to SINTF.

9 (Argument K9) K9 specifies the maximum number of function values to use to compute the integral. If this option is not selected, or if K9 ≤ 0, the number of function values is not bounded.

10 (Argument K10) On return the number of function values used to calculate the integral is stored in K10.

11 (Argument K11) K11 is a signed index providing a means for the user to give the subroutines information about the location and type of any known singularities of the integrand. When an integrand appears to have singular behavior at the end of the interval, a transformation of the variable of integration is applied to reduce the strength of the singularity. When an integrand appears to have singular behavior inside the interval, the abscissa of the singularity is determined as precisely as necessary, depending on the error tolerance, and the interval is subdivided.
The discovery of singular behavior and determination of the abscissa of singular behavior are expensive. If the user knows of the existence of a singularity, the efficiency of computation of the integral may be improved by requesting an immediate transformation of the independent variable or subdivision of the interval. In WORK(|K11|) the user provides the real part of the abscissa of a singularity or discontinuity in the integrand. It is recommended that the user select this option for all singularities, even those outside [A, B]. For singularities having a leading term of the form x^α where α is not an integer, if α is "large" or has the form $\alpha = (2n-1)/2$ where n is a nonnegative integer, or the singularity is well outside the interval, make K11 positive. Otherwise, make K11 negative. The meaning of "large" depends on the rest of the integrand and the length of the interval. "Large" is probably about 2 for the typical case. For a singularity of the form $x^\alpha \log x$ use the above rule, even if α is an integer. For other types of singularity make a reasonable guess based on the above. If several similar integrals are to be computed, some experimentation may be useful.
When K11 > 0, a transformation of the form T = TA + (X − TA)^2/ (TB − TA) is applied, where TA is the abscissa of the singularity and TB is the end of the interval. If TA is outside the interval, TB will be the end of the interval farthest from TA. If TA is inside the interval, the interval will immediately be subdivided at TA, and both parts will be separately integrated with TB equal to each end of the original interval, respectively. When K11 < 0, a transformation of the form T = TA + (X − TA)^4/(TB − TA)^3 is applied, with TA and TB as above.
If the integrand has singularities at more than one abscissa within the region, or more than one pole near the real axis such that the real parts are within the region of integration, the interval should be subdivided at the abscissae of the singularities or the real parts of the poles, the integrals should be computed as separate problems, and the answers summed.

12 (Argument K12) The user provides the error in the lower limit in WORK(K12), and the error in the upper limit in WORK(K12+1). Suppose the lower limit a is a function of several quantities y_j. Then WORK(K12) should be $\Sigma\ |\partial a/\partial y_j|\ E(y_j)$, where E($y_j$) is the error in y_j. The error in the integral due to error in the limit is the error in the limit times the integrand evaluated near the limit. WORK(K12+1) should be evaluated similarly for the upper limit. If this option is not selected, or K12 = 0, then the limits will be assumed to be exact.

B.4 Changing the Selection of Some Options During the Computation

It is possible to change the selection of options 1, 2, 6, 7, and 9 during the integration. To do this, construct an option vector exactly as for the initial call to SINT1. Then

CALL SINTOP (IOPT, WORK)

IF (IOPT(1) .GT. 0) GO TO [Bad IOPT value routine]

This call does not result in any option selections being set to their default values. Otherwise, the action of each option is as described in Section B.3.

B.5 Modifications for Double Precision Usage

For double precision usage, change all REAL type statements to DOUBLE PRECISION and change the prefix of all subroutine names from SINT to DINT.

C. Examples and Remarks

See DRSINT1F and ODSINT1F for an example of the use of SINT1 to compute

$$\int_0^\pi \sin x \, dx - 2 = 0$$

D. Functional Description

The integral is estimated using quadrature formulae due to T. N. L. Patterson [1]. Patterson described a family of formulae in which the k^{th} formula used all the integrand values used in the $k-1^{st}$ formula, and added 2^{k-1} new integrand values in an optimal way. The first formula is the midpoint rule, the second is the three point Gauss formula, and the third is the seven point Kronrod formula. Formulae of this family of higher degree had not previously been described. This program uses formulae up to $k=8$.

An error estimate is obtained by comparing the values of the integral estimated by two adjacent formulae, examining differences up to the fifteenth order, integrating round-off error, integrating error declared to have been committed during computation of the integrand, integrating a first order estimate of the effect round-off error in the abscissae has on integrand values, and including errors in the limits. The latter four methods are also used to derive a bound on the achievable precision.

If the integral over an interval cannot be estimated with sufficient accuracy the interval is subdivided. The difference table is used to discover whether the integral is difficult to compute because the integrand is too complex or has singular behavior. In the former case, the estimated error, requested error tolerance, and difference table are used to choose a step size.

In the latter case, the difference table is used in a search algorithm to find the abscissa of the singular behavior. If the singular behavior is discovered on the end of an interval, a change of independent variable is applied to reduce the strength of the singularity.

The program also uses the difference table to detect non-integrable singularities, jump discontinuities, and computational noise.

The extensive test results given in [2] and [3] suggest that SINT1 is more reliable and requires fewer function evaluations than seven other generally available single precision quadrature subprograms, and DINT1 is more reliable and requires fewer function evaluations than three other generally available double precision quadrature subprograms.

References

1. T. N. L. Patterson, *The optimum addition of points to quadrature formulae*, **Math. of Comp. 22** (1968) 847–856.

2. Fred T. Krogh and W. Van Snyder, **Preliminary Test Results with a New Quadrature Subroutine**. Internal Computing Memorandum 363, Jet Propulsion Laboratory (July 1974). Revised April 29, 1975.

3. Fred T. Krogh and W. Van Snyder, **Test Results With Quadrature Software**. Internal Computing Memorandum 502, Jet Propulsion Laboratory (Oct. 1977).

E. Error Procedures and Restrictions

Error messages are printed using the extended error message processor described in Chapter 19.3. If an error does not result in a "stop," error signals are returned to the user by values of the status flag in IOPT(1). Printing, when enabled by Option 2, is executed in subroutines SINTO for single precision and DINTO for double precision. Both of these programs also use the message processor described in Chapter 19.3. One can change the action on errors and parameters affecting the output of messages, by calling the message/error routine MESS before calling this routine.

F. Supporting Information

Entry	Required Files
DINT1	AMACH, DINT1, DINTA, DINTDL, DINTDU, DINTF, DINTNS, DINTO, DINTOP, DINTSM, DMESS, MESS
DINTA	AMACH, DINTA, DINTDL, DINTDU, DINTF, DINTNS, DINTO, DINTSM, DMESS, MESS
DINTOP	AMACH, DINTOP, MESS
SINT1	AMACH, MESS, SINT1, SINTA, SINTDL, SINTDU, SINTF, SINTNS,SINTO, SINTOP, SINTSM, SMESS
SINTA	AMACH, MESS, SINTA, SINTDL, SINTDU, SINTF, SINTNS, SINTO, SINTSM, SMESS
SINTOP	AMACH, MESS, SINTOP

The source language in ANSI Fortran 77. Common blocks referenced: SINTC and SINTEC in the single precision versions, and DINTC and DINTEC in the double precision versions. SINTC and DINTC are written using several COMMON statements, for maintenance purposes. This usage conforms to the ANSI Fortran-77 standard, but at least one compiler interprets it improperly. If you experience inscrutable errors, try re-writing the COMMON statements for SINTC (or DINTC) into a single statement.

Designed by Fred T. Krogh and W. Van Snyder, JPL, 1977. Programmed by W. Van Snyder, 1977.
Revised by W. Van Snyder, 1986, 1988.
Error/Message handling revised by F. T. Krogh, March 1992.

Two demonstration drivers are shown below, with the output they produced when run on an IBM PC/AT with an 80287 floating point coprocessor. The first demonstrates forward communication usage; the second demonstrates reverse communication usage.

DRSINT1F

```
c       DRSINT1F
c>> 2000-12-03 DRSINT1F Krogh   Declared WORK as WORK(1) in SINTF.
c>> 1994-11-02 DRSINT1F Krogh   Changes to use M77CON
c>> 1994-08-08 DRSINT1F Snyder  Took '0' out of formats for C conversion
c>> 1993-05-05 DRSINT1F Krogh   Adjusted to simplify conversion to C.
c>> 1987-12-09 DRSINT1F Snyder  Initial Code.
c--S replaces "?": DR?INT1f, ?INT1, ?INTF
c
c      DEMO DRIVER for 1 dimensional quadrature subprogram SINT1.
c
c      Compute the integral for X = 0.0 to X = PI of SIN(X), then
c      subtract 2.0 from the ANSWER.  The result should be zero.
c
c      The integrand is evaluated by the subprogram SINTF.
c
c
      real             A, B, ANSWER, WORK(1)
      integer IOPT(10)
10    format (/
     1' DRSINT1F:'/
     2' Compute the integral for X = 0.0 to X = PI of SIN(X), then'/
     3' subtract 2.0 from the ANSWER.  The result should be zero.')
20    format (/' ANSWER          =',G15.8/
     1         ' ERROR ESTIMATE  =',G15.8/
     2         ' STATUS FLAG     =',I6/
     3         ' FUNCTION VALUES =',I6)
c
      print 10
      A = 0.0E0
      B = 4.0E0 * ATAN(1.0E0)
      IOPT(2) = 10
      IOPT(3) = 0
      IOPT(4) = 0
c
      call SINT1 (A, B, ANSWER, WORK, IOPT)
      ANSWER = ANSWER - 2.0e0
      print 20, ANSWER, WORK(1), IOPT(1), IOPT(3)
      stop
      end
c   End of DRSINT1F

      subroutine SINTF (ANSWER, WORK, IFLAG)
c
c      Subroutine to provide integrand for SINT1.
```

```
c
      real              ANSWER, WORK(1)
      integer IFLAG
c
c     IFLAG is not used in this example.
c
      answer = sin(work(1))
      return
c
      end
```

ODSINT1F

```
DRSINT1F:
Compute the integral for X = 0.0 to X = PI of SIN(X), then
subtract 2.0 from the ANSWER.  The result should be zero.

ANSWER          =   0.0000000
ERROR ESTIMATE  =  0.65351918E-06
STATUS FLAG     =      -2
FUNCTION VALUES =      15
```

DRSINT1R

```
c      program DRSINT1R
c>> 1994-10-19 DRSINT1R Krogh  Changes to use M77CON
c>> 1994-08-08 DRSINT1R Snyder  Took '0' out of formats for C conversion
c>> 1991-11-20 CLL Edited for Fortran 90.
c>> 1987-12-09 DRSINT1R Snyder  Initial Code.
c--S replaces "?": DR?INT1R, ?INT1, ?INTA
c
c      DEMO DRIVER for 1 dimensional quadrature subprogram SINT1.
c
c      Compute the integral for X = 0.0 to X = PI of SIN(X), then
c      subtract 2.0 from the ANSWER.  The result should be zero.
c
c      The integrand is evaluated using reverse communication.
c
      real             A, B, ANSWER, WORK(1)
      integer IOPT(10)
c
10    format (/' DRSINT1R:'/
     1' Compute the integral for X = 0.0 to X = PI of SIN(X), then'/
     2' subtract 2.0 from the ANSWER.  The result should be zero.')
20    format (/' ANSWER          =',G15.8/
     1         ' ERROR ESTIMATE  =',G15.8/
     2         ' STATUS FLAG     =',I6/
     3         ' FUNCTION VALUES =',I6)
c
      print 10
      A = 0.0E0
      B = 4.0E0 * ATAN(1.0E0)
      IOPT(2) = 10
      IOPT(3) = 0
      IOPT(4) = 6
      IOPT(5) = 0
      call SINT1 (A, B, ANSWER, WORK, IOPT)
 30   continue
      call SINTA (ANSWER, WORK, IOPT)
      if (IOPT(1) .eq. 0) then
         ANSWER = SIN(WORK(1))
         go to 30
      end if
      print 20, ANSWER, WORK(1), IOPT(1), IOPT(3)
      stop
      end
```

ODSINT1R

```
DRSINT1R:
Compute the integral for X = 0.0 to X = PI of SIN(X), then
subtract 2.0 from the ANSWER.  The result should be zero.

ANSWER          =   2.0000000
ERROR ESTIMATE  = 0.65351918E-06
STATUS FLAG     =     -2
FUNCTION VALUES =     15
```