

# NAG Library Routine Document

## D01EAF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

D01EAF computes approximations to the integrals of a vector of similar functions, each defined over the same multi-dimensional hyper-rectangular region. The routine uses an adaptive subdivision strategy, and also computes absolute error estimates.

### 2 Specification

```

SUBROUTINE D01EAF (NDIM, A, B, MINCLS, MAXCLS, NFUN, FUNSUB, ABSREQ,
1                RELREQ, LENWRK, WRKSTR, FINEST, ABSEST, IFAIL)
    INTEGER      NDIM, MINCLS, MAXCLS, NFUN, LENWRK, IFAIL
    double precision A(NDIM), B(NDIM), ABSREQ, RELREQ, WRKSTR(LENWRK),
1                FINEST(NFUN), ABSEST(NFUN)
    EXTERNAL    FUNSUB

```

### 3 Description

D01EAF uses a globally adaptive method based on the algorithm described by van Dooren and de Ridder (1976) and Genz and Malik (1980). It is implemented for integrals in the form:

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_n}^{b_n} (f_1, f_2, \dots, f_m) dx_n \dots dx_2 dx_1,$$

where  $f_i = f_i(x_1, x_2, \dots, x_n)$ , for  $i = 1, 2, \dots, m$ .

Upon entry, unless MINCLS has been set to a value less than or equal to 0, D01EAF divides the integration region into a number of subregions with randomly selected volumes. Inside each subregion the integrals and their errors are estimated. The initial number of subregions is chosen to be as large as possible without using more than MINCLS calls to FUNSUB. The results are stored in a partially ordered list (a heap). The routine then proceeds in stages. At each stage the subregion with the largest error (measured using the maximum norm) is halved along the co-ordinate axis where the integrands have largest absolute fourth differences. The basic rule is applied to each half of this subregion and the results are stored in the list. The results from the two halves are used to update the global integral and error estimates (FINEST and ABSEST) and the routine continues unless  $\|ABSEST\| \leq \max(ABSREQ, \|FINEST\| \times RELREQ)$  where the norm  $\|\cdot\|$  is the maximum norm, or further subdivision would use more than MAXCLS calls to FUNSUB. If at some stage there is insufficient working storage to keep the results for the next subdivision, the routine switches to a less efficient mode; only if this mode of operation breaks down is insufficient storage reported.

### 4 References

Genz A C and Malik A A (1980) An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region *J. Comput. Appl. Math.* **6** 295–302

van Dooren P and de Ridder L (1976) An adaptive algorithm for numerical integration over an N-dimensional cube *J. Comput. Appl. Math.* **2** 207–217

## 5 Parameters

- 1: NDIM – INTEGER *Input*  
*On entry:*  $n$ , the number of dimensions of the integrals.  
*Constraint:*  $\text{NDIM} \geq 1$ .
- 2: A(NDIM) – *double precision* array *Input*  
*On entry:* the lower limits of integration,  $a_i$ , for  $i = 1, 2, \dots, n$ .
- 3: B(NDIM) – *double precision* array *Input*  
*On entry:* the upper limits of integration,  $b_i$ , for  $i = 1, 2, \dots, n$ .
- 4: MINCLS – INTEGER *Input/Output*  
*On entry:* must be set either to the minimum number of FUNSUB calls to be allowed, in which case  $\text{MINCLS} \geq 0$  or to a negative value. In this case, the routine continues the calculation started in a previous call with the same integrands and integration limits: no parameters other than MINCLS, MAXCLS, ABSREQ, RELREQ or IFAIL must be changed between the calls.  
*On exit:* gives the number of FUNSUB calls actually used by D01EAF. For the continuation case ( $\text{MINCLS} < 0$  on entry) this is the number of new FUNSUB calls on the current call to D01EAF.
- 5: MAXCLS – INTEGER *Input*  
*On entry:* the maximum number of FUNSUB calls to be allowed. In the continuation case this is the number of new FUNSUB calls to be allowed.  
*Constraints:*  
 $\text{MAXCLS} \geq \text{MINCLS};$   
 $\text{MAXCLS} \geq r;$   
 where  $r = 2^n + 2n^2 + 2n + 1$ , if  $n < 11$ , or  $r = 1 + n(4n^2 - 6n + 14)/3$ , if  $n \geq 11$ .
- 6: NFUN – INTEGER *Input*  
*On entry:*  $m$ , the number of integrands.  
*Constraint:*  $\text{NFUN} \geq 1$ .
- 7: FUNSUB – SUBROUTINE, supplied by the user. *External Procedure*  
 FUNSUB must evaluate the integrands  $f_i$  at a given point.

The specification of FUNSUB is:

```
SUBROUTINE FUNSUB (NDIM, Z, NFUN, F)
  INTEGER          NDIM, NFUN
  double precision Z (NDIM), F (NFUN)
```

- 1: NDIM – INTEGER *Input*  
*On entry:*  $n$ , the number of dimensions of the integrals.
- 2: Z(NDIM) – *double precision* array *Input*  
*On entry:* the co-ordinates of the point at which the integrands must be evaluated.
- 3: NFUN – INTEGER *Input*  
*On entry:*  $m$ , the number of integrands.

|    |                                                                                                                        |               |
|----|------------------------------------------------------------------------------------------------------------------------|---------------|
| 4: | F(NFUN) – <i>double precision</i> array<br><i>On exit</i> : the value of the <i>i</i> th integrand at the given point. | <i>Output</i> |
|----|------------------------------------------------------------------------------------------------------------------------|---------------|

FUNSUB must be declared as EXTERNAL in the (sub)program from which D01EAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

- 8: ABSREQ – *double precision* *Input*  
*On entry*: the absolute accuracy required by you.  
*Constraint*:  $\text{ABSREQ} \geq 0.0$ .
- 9: RELREQ – *double precision* *Input*  
*On entry*: the relative accuracy required by you.  
*Constraint*:  $\text{RELREQ} \geq 0.0$ .
- 10: LENWRK – INTEGER *Input*  
*On entry*: the dimension of the array WRKSTR as declared in the (sub)program from which D01EAF is called.  
*Suggested value*:  $\text{LENWRK} \geq 6n + 9m + (n + m + 2)(1 + p/r)$ , where *p* is the value of MAXCLS and *r* is defined under MAXCLS. If LENWRK is significantly smaller than this, the routine will not work as efficiently and may even fail.  
*Constraint*:  $\text{LENWRK} \geq 8 \times \text{NDIM} + 11 \times \text{NFUN} + 3$ .
- 11: WRKSTR(LENWRK) – *double precision* array *Input/Output*  
*On entry*: if MINCLS < 0, WRKSTR must be unchanged from the previous call of D01EAF.  
*On exit*: contains information about the current subdivision which could be used in a continuation call.
- 12: FINEST(NFUN) – *double precision* array *Output*  
*On exit*: FINEST(*i*) specifies the best estimate obtained from the *i*th integral, for  $i = 1, 2, \dots, m$ .
- 13: ABSEST(NFUN) – *double precision* array *Output*  
*On exit*: ABSEST(*i*) specifies the estimated absolute accuracy of FINEST(*i*), for  $i = 1, 2, \dots, m$ .
- 14: IFAIL – INTEGER *Input/Output*  
*On entry*: IFAIL must be set to 0, –1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.  
*On exit*: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value –1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output parameters may be useful even if IFAIL  $\neq$  0 on exit, the recommended value is –1. **When the value –1 or 1 is used it is essential to test the value of IFAIL on exit.**

## 6 Error Indicators and Warnings

If on entry  $IFAIL = 0$  or  $-1$ , explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$IFAIL = 1$

MAXCLS was too small for D01EAF to obtain the required accuracy. The arrays FINEST and ABSEST respectively contain current estimates for the integrals and errors.

$IFAIL = 2$

LENWRK is too small for the routine to continue. The arrays FINEST and ABSEST respectively contain current estimates for the integrals and errors.

$IFAIL = 3$

On a continuation call, MAXCLS was set too small to make any progress. Increase MAXCLS before calling D01EAF again.

$IFAIL = 4$

On entry,  $NDIM < 1$ ,  
 or  $NFUN < 1$ ,  
 or  $MAXCLS < MINCLS$ ,  
 or  $MAXCLS < r$  (see MAXCLS),  
 or  $ABSREQ < 0.0$ ,  
 or  $RELREQ < 0.0$ ,  
 or  $LENWRK < 8 \times NDIM + 11 \times NFUN + 3$ .

## 7 Accuracy

An absolute error estimate for each integrand is output in the array ABSEST. The routine exits with  $IFAIL = 0$  if

$$\max_i(\text{ABSEST}(i)) \leq \max\left(\text{ABSREQ}, \text{RELREQ} \times \max_i|\text{FINEST}(i)|\right).$$

## 8 Further Comments

Usually the running time for D01EAF will be dominated by the time in FUNSUB, so the maximum time that could be used by D01EAF will be proportional to MAXCLS multiplied by the cost of a call to FUNSUB.

On a normal call, you should set  $MINCLS = 0$  on entry.

For some integrands, particularly those that are poorly behaved in a small part of the integration region, D01EAF may terminate prematurely with values of ABSEST that are significantly smaller than the actual absolute errors. This behaviour should be suspected if the returned value of MINCLS is small relative to the expected difficulty of the integrals. When this occurs D01EAF should be called again, but with an entry value of  $MINCLS \geq 2r$ , (see specification of MAXCLS) and the results compared with those from the previous call.

If the routine is called with  $MINCLS \geq 2r$ , the exact values of FINEST and ABSEST on return will depend (within statistical limits) on the sequence of random numbers generated internally within D01EAF by calls to G05SAF. Separate runs will produce identical answers unless the part of the program executed prior to calling D01EAF also calls (directly or indirectly) routines from Chapter G05, and, in addition, the series of such calls differs between runs.

Because of moderate instability in the application of the basic integration rule, approximately the last  $1 + \log_{10}(n^3)$  decimal digits may be inaccurate when using D01EAF for large values of  $n$ .

## 9 Example

This example computes

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 (f_1, f_2, \dots, f_{10}) dx_4 dx_3 dx_2 dx_1,$$

where, for  $j = 1, 2, \dots, 10$ ,  $f_j = \ln(x_1 + 2x_2 + 3x_3 + 4x_4) \sin(j + x_1 + 2x_2 + 3x_3 + 4x_4)$ . The program is intended to show how to exploit the continuation facility provided with D01EAF: the routine exits with IFAIL = 1 (printing an explanatory error message) and is re-entered with MAXCLS reset to a larger value. The program can be used with any values of NDIM and NFUN, except that the expression for  $r$  must be changed if NDIM > 10 (see specification of MAXCLS).

### 9.1 Program Text

```

*      D01EAF Example Program Text
*      Mark 14 Revised. NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          NDIM, NFUN, IRCLS, MXCLS, LENWRK
      PARAMETER       (NDIM=4, NFUN=10,
+                    IRCLS=2**NDIM+2*NDIM*NDIM+2*NDIM+1, MXCLS=IRCLS,
+                    LENWRK=6*NDIM+9*NFUN+(NDIM+NFUN+2)
+                    *(1+MXCLS/IRCLS))
      INTEGER          NOUT
      PARAMETER       (NOUT=6)
*      .. Local Scalars ..
      DOUBLE PRECISION ABSREQ, RELREQ
      INTEGER          I, IFAIL, MAXCLS, MINCLS, MULFAC, N
*      .. Local Arrays ..
      DOUBLE PRECISION A(NDIM), ABSEST(NFUN), B(NDIM), FINEST(NFUN),
+                    WRKSTR(LENWRK)
*      .. External Subroutines ..
      EXTERNAL         D01EAF, FUNSUB
*      .. Executable Statements ..
      WRITE (NOUT,*) 'D01EAF Example Program Results'
      DO 20 N = 1, NDIM
          A(N) = 0.0D0
          B(N) = 1.0D0
20    CONTINUE
      MINCLS = 0
      MAXCLS = MXCLS
      ABSREQ = 0.0D0
      RELREQ = 1.0D-3
      IF (NDIM.LE.10) THEN
          MULFAC = 2**NDIM
      ELSE
          MULFAC = 2*NDIM**3
      END IF
40    IFAIL = 1
*
      CALL D01EAF(NDIM,A,B,MINCLS,MAXCLS,NFUN,FUNSUB,ABSREQ,RELREQ,
+              LENWRK,WRKSTR,FINEST,ABSEST,IFAIL)
*
      WRITE (NOUT,*)
      IF (IFAIL.GT.0) THEN
          IF (IFAIL.EQ.1 .OR. IFAIL.EQ.3) THEN
              WRITE (NOUT,*)
+              ' ** MAXCLS too small to obtain required accuracy.'
              WRITE (NOUT,*) '      MAXCLS =', MAXCLS, '      IFAIL =',
+              IFAIL
              WRITE (NOUT,*)
              WRITE (NOUT,99999) 'Results so far (' , MINCLS,
+              ' FUNSUB calls in last call of D01EAF)'
              WRITE (NOUT,*)
              WRITE (NOUT,*) '      I      Integral      Estimated error'
              DO 60 I = 1, NFUN
                  WRITE (NOUT,99998) I, FINEST(I), ABSEST(I)
60          CONTINUE
              WRITE (NOUT,*)

```

```

        MINCLS = -1
        MAXCLS = MAXCLS*MULFAC
        GO TO 40
    END IF
ELSE IF (IFAIL.EQ.0) THEN
    WRITE (NOUT,99999) 'Final results (', MINCLS,
+   ' FUNSUB calls in last call of D01EAF)'
    WRITE (NOUT,*)
    WRITE (NOUT,*) '   I       Integral   Estimated error'
    DO 80 I = 1, NFUN
        WRITE (NOUT,99998) I, FINEST(I), ABSEST(I)
80    CONTINUE
ELSE
    WRITE (NOUT,99997) ' ** D01EAF returned with IFAIL = ', IFAIL
END IF

*
99999 FORMAT (1X,A,I7,A)
99998 FORMAT (1X,I4,2F14.4)
99997 FORMAT (1X,A,I5)
END

*
SUBROUTINE FUNSUB(NDIM,Z,NFUN,F)
*   .. Scalar Arguments ..
    INTEGER    NDIM, NFUN
*   .. Array Arguments ..
    DOUBLE PRECISION F(NFUN), Z(NDIM)
*   .. Local Scalars ..
    DOUBLE PRECISION SUM
    INTEGER    I, N
*   .. Intrinsic Functions ..
    INTRINSIC    DBLE, LOG, SIN
*   .. Executable Statements ..
    SUM = 0.0D0
    DO 20 N = 1, NDIM
        SUM = SUM + DBLE(N)*Z(N)
20    CONTINUE
    DO 40 I = 1, NFUN
        F(I) = LOG(SUM)*SIN(DBLE(I)+SUM)
40    CONTINUE
    RETURN
END

```

## 9.2 Program Data

None.

## 9.3 Program Results

D01EAF Example Program Results

```

** MAXCLS too small to obtain required accuracy.
   MAXCLS =          57      IFAIL =          1

```

Results so far ( 57 FUNSUB calls in last call of D01EAF)

| I  | Integral | Estimated error |
|----|----------|-----------------|
| 1  | 0.0422   | 0.0086          |
| 2  | 0.3998   | 0.0038          |
| 3  | 0.3898   | 0.0127          |
| 4  | 0.0214   | 0.0099          |
| 5  | -0.3666  | 0.0020          |
| 6  | -0.4176  | 0.0120          |
| 7  | -0.0846  | 0.0110          |
| 8  | 0.3261   | 0.0001          |
| 9  | 0.4371   | 0.0112          |
| 10 | 0.1461   | 0.0119          |

```

** MAXCLS too small to obtain required accuracy.
   MAXCLS =          912      IFAIL =          1

```

Results so far ( 798 FUNSUB calls in last call of D01EAF)

| I  | Integral | Estimated error |
|----|----------|-----------------|
| 1  | 0.0384   | 0.0006          |
| 2  | 0.4012   | 0.0006          |
| 3  | 0.3952   | 0.0006          |
| 4  | 0.0258   | 0.0006          |
| 5  | -0.3673  | 0.0006          |
| 6  | -0.4227  | 0.0006          |
| 7  | -0.0895  | 0.0006          |
| 8  | 0.3260   | 0.0006          |
| 9  | 0.4417   | 0.0006          |
| 10 | 0.1514   | 0.0006          |

Final results ( 912 FUNSUB calls in last call of D01EAF)

| I  | Integral | Estimated error |
|----|----------|-----------------|
| 1  | 0.0384   | 0.0004          |
| 2  | 0.4012   | 0.0003          |
| 3  | 0.3952   | 0.0003          |
| 4  | 0.0258   | 0.0003          |
| 5  | -0.3672  | 0.0003          |
| 6  | -0.4227  | 0.0003          |
| 7  | -0.0895  | 0.0003          |
| 8  | 0.3260   | 0.0003          |
| 9  | 0.4417   | 0.0003          |
| 10 | 0.1514   | 0.0003          |

**Example Program**  
Multi-dimensional Integrals of 10 Integrands with Various Numbers of Function Calls



