NAG Fortran Library Routine Document

C06BAF

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

C06BAF accelerates the convergence of a given convergent sequence to its limit.

2 Specification

```
SUBROUTINE C06BAF(SEQN, NCALL, RESULT, ABSERR, WORK, IWORK, IFAIL)
INTEGER           NCALL, IWORK, IFAIL
real              SEQN, RESULT, ABSERR, WORK(IWORK)
```

3 Description

The routine performs Shanks' transformation on a given sequence of real values by means of the Epsilon algorithm of Wynn (1956). A (possibly unreliable) estimate of the absolute error is also given.

The routine must be called repetitively, once for each new term in the sequence.

4 References

Shanks D (1955) Nonlinear transformations of divergent and slowly convergent sequences *J. Math. Phys.* **34** 1–42

Wynn P (1956) On a device for computing the $e_m(S_n)$ transformation *Math. Tables Aids Comput.* **10** 91–96

5 Parameters

1: SEQN – ***real*** *Input*

On entry: the next term of the sequence to be considered.

2: NCALL – INTEGER *Input/Output*

On entry: on the first call NCALL must be set to 0. Thereafter NCALL **must not** be changed between calls.

On exit: the number of terms in the sequence that have been considered.

3: RESULT – ***real*** *Output*

On exit: the estimate of the limit of the sequence. For the first two calls, RESULT = SEQN.

4: ABSERR – ***real*** *Output*

On exit: an estimate of the absolute error in RESULT. For the first three calls, ABSERR is set to a large machine-dependent constant.

5: WORK(IWORK) – ***real*** array *Workspace*

Used as workspace, but **must not** be changed between calls.

6: IWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which C06BAF is called.

Suggested value: (maximum number of terms in the sequence) $+6$. See Section 8.2.

Constraint: IWORK ≥ 7.

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 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, for users not familiar with this parameter the recommended value is 0. **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 $= 1$

On entry, NCALL < 0.

IFAIL $= 2$

On entry, IWORK < 7.

7 Accuracy

The accuracy of the absolute error estimate ABSERR varies considerably with the type of sequence to which the routine is applied. In general it is better when applied to oscillating sequences than to monotonic sequences where it may be a severe underestimate.

8 Further Comments

8.1 Timing

The time taken by the routine is approximately proportional to the final value of NCALL.

8.2 Choice of IWORK

For long sequences, a 'window' of the last n values can be used instead of all the terms of the sequence. Tests on a variety of problems indicate that a suitable value is $n = 50$; this implies a value for IWORK of 56. Users are advised to experiment with other values for their own specific problems.

8.3 Convergence

The routine will induce convergence in some divergent sequences. See Shanks (1955) for more details.

9 Example

The example program attempts to sum the infinite series

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2} = \frac{\pi^2}{12}$$

by considering the sequence of partial sums

$$\sum_{n=1}^{1}, \sum_{n=1}^{2}, \sum_{n=1}^{3}, \ldots, \sum_{n=1}^{10}$$

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     C06BAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          IWORK
      PARAMETER        (IWORK=16)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             ABSERR, ANS, ERROR, PI, R, RESULT, SEQN, SIG
      INTEGER          I, IFAIL, NCALL
*     .. Local Arrays ..
      real             WORK(IWORK)
*     .. External Functions ..
      real             X01AAF
      EXTERNAL         X01AAF
*     .. External Subroutines ..
      EXTERNAL         C06BAF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06BAF Example Program Results'
      WRITE (NOUT,*)
      PI = X01AAF(0.0e0)
      ANS = PI**2/12.0e0
      NCALL = 0
      SIG = 1.0e0
      SEQN = 0.0e0
      WRITE (NOUT,*)
     +  '                                      Estimated       Actual'
      WRITE (NOUT,*)
     +  '   I       SEQN       RESULT        abs error        error'
      WRITE (NOUT,*)
      DO 20 I = 1, 10
         R = real(I)
         SEQN = SEQN + SIG/(R**2)
         IFAIL = 1
*
         CALL C06BAF(SEQN,NCALL,RESULT,ABSERR,WORK,IWORK,IFAIL)
*
         IF (IFAIL.NE.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'C06BAF fails. IFAIL=', IFAIL
            STOP
         END IF
         ERROR = RESULT - ANS
         SIG = -SIG
         WRITE (NOUT,99998) I, SEQN, RESULT, ABSERR, ERROR
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,A,I2)
```

```
99998 FORMAT (1X,I4,2F12.4,3X,2e14.2)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 C06BAF Example Program Results

                                    Estimated       Actual
    I       SEQN       RESULT       abs error        error

    1     1.0000      1.0000         0.13+155     0.18E+00
    2     0.7500      0.7500         0.13+155    -0.72E-01
    3     0.8611      0.8269         0.13+155     0.45E-02
    4     0.7986      0.8211         0.26E+00    -0.14E-02
    5     0.8386      0.8226         0.78E-01     0.12E-03
    6     0.8108      0.8224         0.60E-02    -0.33E-04
    7     0.8312      0.8225         0.15E-02     0.35E-05
    8     0.8156      0.8225         0.16E-03    -0.85E-06
    9     0.8280      0.8225         0.37E-04     0.10E-06
   10     0.8180      0.8225         0.45E-05    -0.23E-07
```