C06LAF – NAG Fortran Library Routine Document

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

C06LAF estimates values of the inverse Laplace transform of a given function using a Fourier series approximation. Real and imaginary parts of the function, and a bound on the exponential order of the inverse, are required.

2 Specification

```
      SUBROUTINE C06LAF(FUN, N, T, VALINV, ERREST, RELERR, ALPHAB, TFAC,
     1                  MXTERM, NTERMS, NA, ALOW, AHIGH, NFEVAL, WORK,
     2                  IFAIL)
      INTEGER           N, MXTERM, NTERMS, NA, NFEVAL, IFAIL
      real              T(N), VALINV(N), ERREST(N), RELERR, ALPHAB,
     1                  TFAC, ALOW, AHIGH, WORK(4*MXTERM+2)
      EXTERNAL          FUN
```

3 Description

Given a function $F(p)$ defined for complex values of p, this routine estimates values of its inverse Laplace transform by Crump's method [2]. (For a definition of the Laplace transform and its inverse, see the Chapter Introduction.)

Crump's method applies the epsilon algorithm (Wynn [3]) to the summation in Durbin's Fourier series approximation [1]

$$f(t_j) \simeq \frac{e^{at_j}}{\tau}\left[\frac{1}{2}F(a) - \sum_{k=1}^{\infty}\left\{\mathrm{Re}(F(a+\frac{k\pi i}{\tau}))\cos\frac{k\pi t_j}{\tau} - \mathrm{Im}(F(a+\frac{k\pi i}{\tau}))\sin\frac{k\pi t_j}{\tau}\right\}\right],$$

for $j = 1, 2, \ldots, n$, by choosing a such that a prescribed relative error should be achieved. The method is modified slightly if $t = 0.0$ so that an estimate of $f(0.0)$ can be obtained when it has a finite value. τ is calculated as $t_{fac} \times \max(0.01, t_j)$, where $t_{fac} > 0.5$. The user specifies t_{fac} and α_b, an upper bound on the exponential order α of the inverse function $f(t)$. α has two alternative interpretations:

(i) α is the smallest number such that

$$|f(t)| \leq m \times \exp(\alpha t)$$

for large t,

(ii) α is the real part of the singularity of $F(p)$ with largest real part.

The method depends critically on the value of α. See Section 8 for further details. The routine calculates at least two different values of the parameter a, such that $a > \alpha_b$, in an attempt to achieve the requested relative error and provide error estimates. The values of t_j, for $j = 1, 2, \ldots, n$, must be supplied in monotonically increasing order. The routine calculates the values of the inverse function $f(t_j)$ in decreasing order of j.

4 References

[1] Durbin F (1974) Numerical inversion of Laplace transforms: An efficient improvement to Dubner and Abate's method *Comput. J.* **17** 371–376

[2] Crump K S (1976) Numerical inversion of Laplace transforms using a Fourier series approximation *J. Assoc. Comput. Mach.* **23** 89–96

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

5 Parameters

1: FUN — SUBROUTINE, supplied by the user. *External Procedure*

FUN must evaluate the real and imaginary parts of the function $F(p)$ for a given value of p.

Its specification is:

```
SUBROUTINE FUN(PR, PI, FR, FI)
real           PR, PI, FR, FI
```

1: PR — ***real*** *Input*
2: PI — ***real*** *Input*

On entry: the real and imaginary parts of the argument p.

3: FR — ***real*** *Output*
4: FI — ***real*** *Output*

On exit: the real and imaginary parts of the value $F(p)$.

FUN must be declared as EXTERNAL in the (sub)program from which C06LAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: N — INTEGER *Input*

On entry: the number of points, n, at which the value of the inverse Laplace transform is required.

Constraint: N ≥ 1.

3: T(N) — ***real*** array *Input*

On entry: each T(j) must specify a point at which the inverse Laplace transform is required, for $j = 1, 2, \ldots, n$.

Constraint: $0.0 \leq$ T(1) $<$ T(2) $< \ldots <$ T(n).

4: VALINV(N) — ***real*** array *Output*

On exit: an estimate of the value of the inverse Laplace transform at $t =$ T(j), for $j = 1, 2, \ldots, n$.

5: ERREST(N) — ***real*** array *Output*

On exit: an estimate of the error in VALINV(j). This is usually an estimate of relative error but, if VALINV(j) $<$ RELERR, ERREST(j) estimates the absolute error. ERREST(j) is unreliable when VALINV(j) is small but slightly greater than RELERR.

6: RELERR — ***real*** *Input*

On entry: the required relative error in the values of the inverse Laplace transform. If the absolute value of the inverse is less than RELERR, then absolute accuracy is used instead. RELERR must be in the range $0.0 \leq$ RELERR < 1.0. If RELERR is set too small or to 0.0, then the routine uses a value sufficiently larger than ***machine precision***.

7: ALPHAB — ***real*** *Input*

On entry: α_b, an upper bound for α (see Section 3). Usually, α_b should be specified equal to, or slightly larger than, the value of α. If $\alpha_b < \alpha$ then the prescribed accuracy may not be achieved or completely incorrect results may be obtained. If α_b is too large the routine will be inefficient and convergence may not be achieved.

Note. It is as important to specify α_b correctly as it is to specify the correct function for inversion.

8: TFAC — ***real*** *Input*

On entry: t_{fac}, a factor to be used in calculating the parameter τ. Larger values (e.g., 5.0) may be specified for difficult problems, but these may require very large values of MXTERM.

Suggested value: TFAC = 0.8.

Constraint: TFAC > 0.5.

9: MXTERM — INTEGER *Input*

On entry: the maximum number of (complex) terms to be used in the evaluation of the Fourier series.

Suggested value: MXTERM $\geq$ 100, except for very simple problems.

Constraint: MXTERM $\geq$ 1.

10: NTERMS — INTEGER *Output*

On exit: the number of (complex) terms actually used.

11: NA — INTEGER *Output*

On exit: the number of values of a used by the routine. See Section 8.

12: ALOW — ***real*** *Output*

On exit: the smallest value of a used in the algorithm. This may be used for checking the value of ALPHAB − see Section 8.

13: AHIGH — ***real*** *Output*

On exit: the largest value of a used in the algorithm. This may be used for checking the value of ALPHAB − see Section 8.

14: NFEVAL — INTEGER *Output*

On exit: the number of calls to FUN made by the routine.

15: WORK(4*MXTERM+2) — ***real*** array *Workspace*

16: 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 or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL $\neq$ 0 on exit, users are recommended to set IFAIL to −1 before entry. **It is then 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 specified by the routine:

IFAIL = 1

On entry, N < 1,
or MXTERM < 1,
or RELERR < 0.0,
or RELERR $\geq$ 1.0,
or TFAC $\leq$ 0.5.

IFAIL = 2

On entry, T(1) < 0.0,
or T(1),T(2),...,T(N) are not in strictly increasing order.

IFAIL = 3

T(N) is too large for this value of ALPHAB. If necessary, scale the problem as described in Section 8.

IFAIL = 4

The required accuracy cannot be obtained. It is possible that ALPHAB is less than α. Alternatively, the problem may be especially difficult. Try increasing TFAC, ALPHAB or both.

IFAIL = 5

Convergence failure in the epsilon algorithm. Some values of VALINV(j) may be calculated to the desired accuracy; this may be determined by examining the values of ERREST(j). Try reducing the range of T or increasing MXTERM. If IFAIL = 5 still results, try reducing TFAC.

IFAIL = 6

All values of VALINV(j) have been calculated but not all are to the requested accuracy; the values of ERREST(j) should be examined carefully. Try reducing the range of t, or increasing TFAC, ALPHAB or both.

7 Accuracy

The error estimates are often very close to the true error but, because the error control depends on an asymptotic formula, the required error may not always be met. There are two principal causes of this: Gibbs' phenomena, and zero or small values of the inverse Laplace transform.

Gibbs' phenomena (see the Chapter Introduction) are exhibited near $t = 0.0$ (due to the method) and around discontinuities in the inverse Laplace transform $f(t)$. If there is a discontinuity at $t = c$ then the method converges such that $f(c) \to (f(c-) + f(c+))/2$.

Apparent loss of accuracy, when $f(t)$ is small, may not be serious. Crump's method keeps control of relative error so that good approximations to small function values may appear to be very inaccurate. If $|f(t)|$ is estimated to be less than RELERR then this routine switches to absolute error estimation. However, when $|f(t)|$ is slightly larger than RELERR the relative error estimates are likely to cause IFAIL = 6. If this is found inconvenient it can sometimes be avoided by adding k/p to the function $F(p)$, which shifts the inverse to $k + f(t)$.

Loss of accuracy may also occur for highly oscillatory functions.

More serious loss of accuracy can occur if α is unknown and is incorrectly estimated. See Section 8.

8 Further Comments

8.1 Timing

The value of n is less important in general than the value of NTERMS. Unless the subroutine FUN is very inexpensive to compute, the timing is proportional to NA × NTERMS. For simple problems NA = 2 but in difficult problems NA may be somewhat larger.

8.2 Precautions

The user is referred to the Chapter Introduction for advice on simplifying problems with particular difficulties, e.g., where the inverse is known to be a step function. The method does not work well for large values of t when α is positive. It is advisable, especially if IFAIL = 3 is obtained, to scale the problem if $|\alpha|$ is much greater than 1.0. See the Chapter Introduction. The range of values of t specified for a particular call should not be greater than about 10 units. This is because the method uses parameters based on the value T(n) and these tend to be less appropriate as t becomes smaller. However, as the timing of the routine is not especially dependent on n, it is usually far more efficient to evaluate the inverse for ranges of t than to make separate calls to the routine for each value of t. The most important parameter to specify correctly is ALPHAB, an upper bound for α. If, on entry, ALPHAB is sufficently smaller than α then completely incorrect results will be obtained with IFAIL = 0. Unless α is known theoretically it is strongly advised that the user should test any estimated value used. This may be done by specifying a single value of t (i.e T(n), n = 1) with two sets of suitable values of TFAC, RELERR and MXTERM, and examining the resulting values of ALOW and AHIGH. The value of T(1) should be chosen very carefully and the following points should be borne in mind:

(i) T(1) should be small but not too close to 0.0 because of Gibbs' phenomenon (see Section 7),

(ii) the larger the value of T(1), the smaller the range of values of a that will be used in the algorithm,

(iii) T(1) should ideally not be chosen such that $f(\mathrm{T}(1)) = 0.0$ or a very small value. For suitable problems T(1) might be chosen as, say, 0.1 or 1.0 depending on these factors. The routine calculates ALOW from the formula

$$\mathrm{ALOW} = \mathrm{ALPHAB} - \frac{\ln(0.1 \times \mathrm{RELERR})}{2 \times \tau}.$$

Additional values of a are computed by adding $1/\tau$ to the previous value. As $\tau = \mathrm{TFAC} \times \mathrm{T}(n)$, it will be seen that large values of TFAC and RELERR will test for a close to ALPHAB. Small values of TFAC and RELERR will test for a large. If the result of both tests is IFAIL = 0, with comparable values for the inverse, then this gives some credibility to the chosen value of ALPHAB. The user should note that this test could be more computationally expensive than the calculation of the inverse itself. The example program (see Section 9) illustrates how such a test may be performed.

9 Example

The example program estimates the inverse Laplace transform of the function $F(p) = 1/(p + 1/2)$. The true inverse of $F(p)$ is $\exp(-t/2)$. Two preliminary calls to the routine are made to verify that the chosen value of ALPHAB is suitable. For these tests the single value T(1) = 1.0 is used. To test values of a close to ALPHAB, the values TFAC = 5.0 and RELERR = 0.01 are chosen. To test larger a, the values TFAC = 0.8 and RELERR = 1.0E−3 are used. Because the values of the computed inverse are similar and IFAIL = 0 in each case, these tests show that there is unlikely to be a singularity of $F(p)$ in the region $-0.04 \leq \mathrm{Re}\, p \leq 6.51$.

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.

```
*     C06LAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX, MXTERM
      PARAMETER        (NMAX=20,MXTERM=200)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             AHIGH, ALOW, ALPHAB, RELERR, TFAC
      INTEGER          I, IFAIL, N, NA, NFEVAL, NTERMS
*     .. Local Arrays ..
      real             ERREST(NMAX), T(NMAX), TRUREL(NMAX),
     +                 TRURES(NMAX), VALINV(NMAX), WORK(4*MXTERM+2)
*     .. External Subroutines ..
      EXTERNAL         C06LAF, FUN
*     .. Intrinsic Functions ..
      INTRINSIC        ABS, EXP, real
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06LAF Example Program Results'
      WRITE (NOUT,*)
      WRITE (NOUT,*) '(results may be machine-dependent)'
      ALPHAB = -0.5e0
      T(1) = 1.0e0
*
*     Test for values of a close to ALPHAB.
*
      RELERR = 0.01e0
      TFAC = 7.5e0
      WRITE (NOUT,*)
      WRITE (NOUT,99997) 'Test with T(1) =', T(1)
      WRITE (NOUT,*)
      WRITE (NOUT,99999) '  MXTERM =', MXTERM, '  TFAC =', TFAC,
     +  '  ALPHAB =', ALPHAB, '  RELERR =', RELERR
      IFAIL = -1
*
      CALL C06LAF(FUN,1,T,VALINV,ERREST,RELERR,ALPHAB,TFAC,MXTERM,
     +            NTERMS,NA,ALOW,AHIGH,NFEVAL,WORK,IFAIL)
*
      IF (IFAIL.GT.0 .AND. IFAIL.LT.5) GO TO 60
      WRITE (NOUT,*)
      WRITE (NOUT,*) '   T        Result        exp(-T/2)   ',
     +  'Relative error  Error estimate'
      TRURES(1) = EXP(-T(1)/2.0e0)
      TRUREL(1) = ABS((VALINV(1)-TRURES(1))/TRURES(1))
      WRITE (NOUT,99998) T(1), VALINV(1), TRURES(1), TRUREL(1),
     +  ERREST(1)
      WRITE (NOUT,*)
      WRITE (NOUT,99996) ' NTERMS =', NTERMS, '  NFEVAL =', NFEVAL,
     +  '  ALOW =', ALOW, '  AHIGH =', AHIGH, '  IFAIL =', IFAIL
*
*     Test for larger values of a.
*
      RELERR = 1.0e-3
      TFAC = 0.8e0
      WRITE (NOUT,*)
      WRITE (NOUT,99997) 'Test with T(1) =', T(1)
      WRITE (NOUT,*)
      WRITE (NOUT,99999) '  MXTERM =', MXTERM, '  TFAC =', TFAC,
     +  '  ALPHAB =', ALPHAB, '  RELERR =', RELERR
      IFAIL = -1
*
      CALL C06LAF(FUN,1,T,VALINV,ERREST,RELERR,ALPHAB,TFAC,MXTERM,
     +            NTERMS,NA,ALOW,AHIGH,NFEVAL,WORK,IFAIL)
*
      IF (IFAIL.GT.0 .AND. IFAIL.LT.5) GO TO 60
      WRITE (NOUT,*)
      WRITE (NOUT,*) '   T        Result        exp(-T/2)   ',
     +  'Relative error  Error estimate'
      TRURES(1) = EXP(-T(1)/2.0e0)
      TRUREL(1) = ABS((VALINV(1)-TRURES(1))/TRURES(1))
      WRITE (NOUT,99998) T(1), VALINV(1), TRURES(1), TRUREL(1),
     +  ERREST(1)
      WRITE (NOUT,*)
      WRITE (NOUT,99996) ' NTERMS =', NTERMS, '  NFEVAL =', NFEVAL,
```

```
     +   '  ALOW =', ALOW, '  AHIGH =', AHIGH, '  IFAIL =', IFAIL
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Compute inverse'
      WRITE (NOUT,*)
      WRITE (NOUT,99999) '  MXTERM =', MXTERM, '  TFAC =', TFAC,
     +   '  ALPHAB =', ALPHAB, '  RELERR =', RELERR
      WRITE (NOUT,*)
      WRITE (NOUT,*) '   T        Result        exp(-T/2)    ',
     +   'Relative error  Error estimate'
      N = 5
      DO 20 I = 1, N
         T(I) = real(I)
   20 CONTINUE
      IFAIL = -1
*
      CALL C06LAF(FUN,N,T,VALINV,ERREST,RELERR,ALPHAB,TFAC,MXTERM,
     +            NTERMS,NA,ALOW,AHIGH,NFEVAL,WORK,IFAIL)
*
      IF (IFAIL.GT.0 .AND. IFAIL.LT.5) GO TO 60
      DO 40 I = 1, N
         TRURES(I) = EXP(-T(I)/2.0e0)
         TRUREL(I) = ABS((VALINV(I)-TRURES(I))/TRURES(I))
   40 CONTINUE
      WRITE (NOUT,99998) (T(I),VALINV(I),TRURES(I),TRUREL(I),ERREST(I),
     +  I=1,N)
   60 WRITE (NOUT,*)
      WRITE (NOUT,99996) ' NTERMS =', NTERMS, '  NFEVAL =', NFEVAL,
     +   '  ALOW =', ALOW, '  AHIGH =', AHIGH, '  IFAIL =', IFAIL
*
99999 FORMAT (1X,A,I4,A,F6.2,A,F6.2,A,1P,e8.1)
99998 FORMAT (1X,F4.1,7X,F6.3,9X,F6.3,8X,e8.1,8X,e8.1)
99997 FORMAT (1X,A,F4.1)
99996 FORMAT (1X,A,I4,A,I4,A,F7.2,A,F7.2,A,I2)
      END
*
      SUBROUTINE FUN(PR,PI,FR,FI)
*     Function to be inverted
*     .. Scalar Arguments ..
      real              FI, FR, PI, PR
*     .. External Subroutines ..
      EXTERNAL          A02ACF
*     .. Executable Statements ..
      CALL A02ACF(1.0e0,0.0e0,PR+0.5e0,PI,FR,FI)
*
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
C06LAF Example Program Results

(results may be machine-dependent)

Test with T(1) = 1.0

  MXTERM = 200  TFAC =  7.50  ALPHAB = -0.50  RELERR = 1.0E-02

   T        Result        exp(-T/2)   Relative error  Error estimate
 1.0        0.607          0.607         0.1E-02         0.4E-02

 NTERMS =  18  NFEVAL =  36  ALOW =  -0.04  AHIGH =   0.09  IFAIL = 0

Test with T(1) = 1.0

  MXTERM = 200  TFAC =  0.80  ALPHAB = -0.50  RELERR = 1.0E-03

   T        Result        exp(-T/2)   Relative error  Error estimate
 1.0        0.607          0.607         0.2E-04         0.8E-04

 NTERMS =  13  NFEVAL =  28  ALOW =   5.26  AHIGH =   6.51  IFAIL = 0

Compute inverse

  MXTERM = 200  TFAC =  0.80  ALPHAB = -0.50  RELERR = 1.0E-03

   T        Result        exp(-T/2)   Relative error  Error estimate
 1.0        0.607          0.607         0.5E-04         0.3E-03
 2.0        0.368          0.368         0.7E-05         0.9E-04
 3.0        0.223          0.223         0.2E-04         0.8E-04
 4.0        0.135          0.135         0.1E-04         0.8E-04
 5.0        0.082          0.082         0.2E-04         0.8E-04

 NTERMS =  23  NFEVAL =  43  ALOW =   0.65  AHIGH =   0.90  IFAIL = 0
```
