C06LBF – 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

C06LBF computes the inverse Laplace transform $f(t)$ of a user-supplied function $F(s)$, defined for complex s. The routine uses a modification of Weeks' method which is suitable when $f(t)$ has continuous derivatives of all orders. The routine returns the coefficients of an expansion which approximates $f(t)$ and can be evaluated for given values of t by subsequent calls of C06LCF.

2 Specification

```
      SUBROUTINE C06LBF(F, SIGMA0, SIGMA, B, EPSTOL, MMAX, M, ACOEF,
     1                  ERRVEC, IFAIL)
      INTEGER           MMAX, M, IFAIL
      real              SIGMA0, SIGMA, B, EPSTOL, ACOEF(MMAX), ERRVEC(8)
      complex           F
      EXTERNAL          F
```

3 Description

Given a function $f(t)$ of a real variable t, its Laplace transform $F(s)$ is a function of a complex variable s, defined by:

$$F(s) = \int_0^\infty e^{-st} f(t)\,dt, \qquad \mathrm{Re}\,s > \sigma_0.$$

Then $f(t)$ is the inverse Laplace transform of $F(s)$. The value σ_0 is referred to as the abscissa of convergence of the Laplace transform; it is the rightmost real part of the singularities of $F(s)$.

This routine, along with its companion C06LCF, attempts to solve the following problem:

given a function $F(s)$, compute values of its inverse Laplace transform $f(t)$ for specified values of t.

The method is a modification of Weeks' method (see Garbow *et al.*[1]), which approximates $f(t)$ by a truncated Laguerre expansion:

$$\tilde{f}(t) = e^{\sigma t} \sum_{i=0}^{m-1} a_i\, e^{-bt/2} L_i(bt), \qquad \sigma > \sigma_0, \quad b > 0$$

where $L_i(x)$ is the Laguerre polynomial of degree i. This routine computes the coefficients a_i of the above Laguerre expansion; the expansion can then be evaluated for specified t by calling C06LCF. The user must supply the value of σ_0, and also suitable values for σ and b: see Section 8 for guidance.

The method is only suitable when $f(t)$ has continuous derivatives of all orders. For such functions the approximation $\tilde{f}(t)$ is usually good and inexpensive. The routine will fail with an error exit if the method is not suitable for the supplied function $F(s)$.

The routine is designed to satisfy an accuracy criterion of the form:

$$\left| \frac{f(t) - \tilde{f}(t)}{e^{\sigma t}} \right| < \epsilon_{tol}, \qquad \text{for all } t$$

where ϵ_{tol} is a user-supplied bound. The error measure on the left-hand side is referred to as the **pseudo-relative error**, or **pseudo-error** for short. Note that if $\sigma > 0$ and t is large, the absolute error in $\tilde{f}(t)$ may be very large.

C06LBF is derived from the subroutine MODUL1 in [2].

4 References

[1] Garbow B S, Giunta G, Lyness J N and Murli A (1988) Software for an implementation of Weeks' method for the inverse laplace transform problem *ACM Trans. Math. Software* **14** 163–170

[2] Garbow B S, Giunta G, Lyness J N and Murli A (1988) Algorithm 662: A Fortran software package for the numerical inversion of the Laplace transform based on Weeks' method *ACM Trans. Math. Software* **14** 171–176

5 Parameters

1: F — ***complex*** FUNCTION, supplied by the user. *External Procedure*

F must return the value of the Laplace transform function $F(s)$ for a given complex value of s.

Its specification is:

```
complex FUNCTION F(S)
complex          S
```

1: S — ***complex*** *Input*

On entry: the value of s for which $F(s)$ must be evaluated. The real part of S is greater than σ_0.

F must be declared as EXTERNAL in the (sub)program from which C06LBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: SIGMA0 — ***real*** *Input*

On entry: the abscissa of convergence of the Laplace transform, σ_0.

3: SIGMA — ***real*** *Input/Output*

On entry: the parameter σ of the Laguerre expansion. If on entry SIGMA $\leq \sigma_0$, SIGMA is reset to $\sigma_0 + 0.7$.

On exit: the value actually used for σ, as just described.

4: B — ***real*** *Input/Output*

On entry: the parameter b of the Laguerre expansion. If on entry B $< 2(\sigma - \sigma_0)$, B is reset to $2.5(\sigma - \sigma_0)$.

On exit: the value actually used for b, as just described.

5: EPSTOL — ***real*** *Input*

On entry: the required relative pseudo-accuracy, that is, an upper bound on $|f(t) - \tilde{f}(t)|e^{-\sigma t}$.

6: MMAX — INTEGER *Input*

On entry: an upper bound on the number of Laguerre expansion coefficients to be computed. The number of coefficients actually computed is always a power of 2, so MMAX should be a power of 2; if MMAX is not a power of 2 then the maximum number of coefficients calculated will be the largest power of 2 less than MMAX.

Suggested value: MMAX = 1024 is sufficient for all but a few exceptional cases.

Constraint: MMAX ≥ 8.

7: M — INTEGER *Output*

On exit: the number of Laguerre expansion coefficients actually computed. The number of calls to F is M/2 + 2.

8: ACOEF(MMAX) — ***real*** array *Output*

On exit: the first M elements contain the computed Laguerre expansion coefficients, a_i.

9: ERRVEC(8) — ***real*** array *Output*

On exit: an 8-component vector of diagnostic information:

ERRVEC(1)	=	overall estimate of the pseudo-error $\lvert f(t) - \tilde{f}(t)\rvert e^{-\sigma t}$;
	=	ERRVEC(2) + ERRVEC(3) + ERRVEC(4);
ERRVEC(2)	=	estimate of the discretisation pseudo-error;
ERRVEC(3)	=	estimate of the truncation pseudo-error;
ERRVEC(4)	=	estimate of the condition pseudo-error on the basis of minimal noise levels in function values;
ERRVEC(5)	=	K, coefficient of a heuristic decay function for the expansion coefficients;
ERRVEC(6)	=	R, base of the decay function for the expansion coefficients;
ERRVEC(7)	=	logarithm of the largest expansion coefficient; and
ERRVEC(8)	=	logarithm of the smallest non-zero expansion coefficient.

The values K and R returned in ERRVEC(5) and ERRVEC (6) define a decay function KR^{-i} constructed by the routine for the purposes of error estimation. It satisfies

$$|a_i| < KR^{-i}, \quad \text{for } i = 1, 2, \ldots, m.$$

10: 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, MMAX < 8.

IFAIL = 2

The estimated pseudo-error bounds are slightly larger than EPSTOL. Note, however, that the actual errors in the final results may be smaller than EPSTOL as bounds independent of the value of t are pessimistic.

IFAIL = 3

Computation was terminated early because the estimate of rounding error was greater than EPSTOL. Increasing EPSTOL may help.

IFAIL = 4

The decay rate of the coefficients is too small. Increasing MMAX may help.

IFAIL = 5

The decay rate of the coefficients is too small. In addition the rounding error is such that the required accuracy cannot be obtained. Increasing MMAX or EPSTOL may help.

IFAIL = 6

The behaviour of the coefficients does not enable reasonable prediction of error bounds. Check the value of SIGMA0. In this case, ERRVEC(i) is set to -1.0, for $i = 1$ to 5.

When IFAIL ≥ 3, changing SIGMA or B may help. If not, the method should be abandoned.

7 Accuracy

The error estimate returned in ERRVEC(1) has been found in practice to be a highly reliable bound on the pseudo-error $|f(t) - \tilde{f}(t)|e^{-\sigma t}$.

8 Further Comments

8.1 The Role of σ_0

Nearly all techniques for inversion of the Laplace transform require the user to supply the value of σ_0, the convergence abscissa, or else an upper bound on σ_0. For this routine, one of the reasons for having to supply σ_0 is that the parameter σ must be greater than σ_0; otherwise the series for $\tilde{f}(t)$ will not converge.

If you do not know the value of σ_0, you must be prepared for significant preliminary effort, either in experimenting with the method and obtaining chaotic results, or in attempting to locate the rightmost singularity of $F(s)$.

The value of σ_0 is also relevant in defining a natural accuracy criterion. For large t, $f(t)$ is of uniform numerical order $ke^{\sigma_0 t}$, so a **natural** measure of relative accuracy of the approximation $\tilde{f}(t)$ is:

$$\epsilon_{nat}(t) = (\tilde{f}(t) - f(t))/e^{\sigma_0 t}.$$

The routine uses the supplied value of σ_0 only in determining the values of σ and b (see below); thereafter it bases its computation entirely on σ and b.

8.2 Choice of σ

Even when the value of σ_0 is known, choosing a value for σ is not easy. Briefly, the series for $\tilde{f}(t)$ converges slowly when $\sigma - \sigma_0$ is small, and faster when $\sigma - \sigma_0$ is larger. However the natural accuracy measure satisfies

$$|\epsilon_{nat}(t)| < \epsilon_{tol}\ e^{(\sigma-\sigma_0)t}$$

and this degrades exponentially with t, the exponential constant being $\sigma - \sigma_0$.

Hence, if you require meaningful results over a large range of values of t, you should choose $\sigma - \sigma_0$ small, in which case the series for $\tilde{f}(t)$ converges slowly; while for a smaller range of values of t, you can allow $\sigma - \sigma_0$ to be larger and obtain faster convergence.

The default value for σ used by the routine is $\sigma_0 + 0.7$. There is no theoretical justification for this.

8.3 Choice of b

The simplest advice for choosing b is to set $b/2 \geq \sigma - \sigma_0$. The default value used by the routine is $2.5(\sigma - \sigma_0)$.

A more refined choice is to set

$$b/2 \geq \min_j |\sigma - s_j|$$

where s_j are the singularities of $F(s)$.

9 Example

To compute values of the inverse Laplace transform of the function

$$F(s) = \frac{3}{s^2 - 9}.$$

The exact answer is

$$f(t) = \sinh 3t.$$

The program first calls C06LBF to compute the coefficients of the Laguerre expansion, and then calls C06LCF to evaluate the expansion at $t = 1, 2, 3, 4, 5$.

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.

```
*     C06LBF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX
      PARAMETER        (MMAX=512)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             B, EPSTOL, EXACT, FINV, PSERR, SIGMA, SIGMA0, T
      INTEGER          IFAIL, J, M
*     .. Local Arrays ..
      real             ACOEF(MMAX), ERRVEC(8)
*     .. External Subroutines ..
      EXTERNAL         C06LBF, C06LCF
*     .. External Functions ..
      complex          F
      EXTERNAL         F
*     .. Intrinsic Functions ..
      INTRINSIC        ABS, EXP, real, SINH
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06LBF Example Program Results'
      SIGMA0 = 3.0e0
      EPSTOL = 0.00001e0
      SIGMA = 0.0e0
      B = 0.0e0
      IFAIL = 0
*
*     Compute inverse transform
      CALL C06LBF(F,SIGMA0,SIGMA,B,EPSTOL,MMAX,M,ACOEF,ERRVEC,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,99999) 'No. of coefficients returned by C06LBF =', M
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +  '                    Computed          Exact      Pseudo'
      WRITE (NOUT,*)
     +  '        t           f(t)           f(t)       error'
      WRITE (NOUT,*)
*
*     Evaluate inverse transform for different values of t
      DO 20 J = 0, 5
         T = real(J)
```

```
*
         CALL C06LCF(T,SIGMA,B,M,ACOEF,ERRVEC,FINV,IFAIL)
*
         EXACT = SINH(3.0e0*T)
         PSERR = ABS(FINV-EXACT)/EXP(SIGMA*T)
         WRITE (NOUT,99998) T, FINV, EXACT, PSERR
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,A,I6)
99998 FORMAT (1X,1P,e10.2,2e15.4,e12.1)
      END
*
      complex FUNCTION F(S)
*     .. Scalar Arguments ..
      complex           S
*     .. Executable Statements ..
      F = 3.0e0/(S**2-9.0e0)
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
C06LBF Example Program Results

No. of coefficients returned by C06LBF =    64

                  Computed          Exact      Pseudo
         t            f(t)           f(t)       error

  0.00E+00     1.5129E-09     0.0000E+00     1.5E-09
  1.00E+00     1.0018E+01     1.0018E+01     1.7E-09
  2.00E+00     2.0171E+02     2.0171E+02     1.2E-10
  3.00E+00     4.0515E+03     4.0515E+03     9.8E-10
  4.00E+00     8.1377E+04     8.1377E+04     3.0E-10
  5.00E+00     1.6345E+06     1.6345E+06     1.7E-09
```