NAG Toolbox

nag_sum_invlaplace_weeks (c06lb)

1 Purpose

nag_sum_invlaplace_weeks (c06lb) computes the inverse Laplace transform $f(t)$ of a user-supplied function $F(s)$, defined for complex s. The function uses a modification of Weeks' method which is suitable when $f(t)$ has continuous derivatives of all orders. The function returns the coefficients of an expansion which approximates $f(t)$ and can be evaluated for given values of t by subsequent calls of nag_sum_invlaplace_weeks_eval (c06lc).

2 Syntax

```
[sigma, b, m, acoef, errvec, ifail] = nag_sum_invlaplace_weeks(f, sigma0, sigma, b, epstol, 'mmax', mmax)
```

```
[sigma, b, m, acoef, errvec, ifail] = c06lb(f, sigma0, sigma, b, epstol, 'mmax', mmax)
```

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 $F(s)$.

nag_sum_invlaplace_weeks (c06lb), along with its companion nag_sum_invlaplace_weeks_eval (c06lc), 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.* (1988a)), which approximates $f(t)$ by a truncated Laguerre expansion:

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

where $L_i(x)$ is the Laguerre polynomial of degree i. This function computes the coefficients a_i of the above Laguerre expansion; the expansion can then be evaluated for specified t by calling nag_sum_invlaplace_weeks_eval (c06lc). You must supply the value of σ_0, and also suitable values for σ and b: see Section 9 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 function will fail with an error exit if the method is not suitable for the supplied function $F(s)$.

The function 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.

nag_sum_invlaplace_weeks (c06lb) is derived from the function MODUL1 in Garbow *et al.* (1988a).

4 References

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

Garbow B S, Giunta G, Lyness J N and Murli A (1988b) 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

5.1 Compulsory Input Parameters

1: **f** – COMPLEX (KIND=nag_wp) FUNCTION, supplied by the user.

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

```
[result] = f(s)
```

Input Parameters

1: **s** – COMPLEX (KIND=nag_wp)

The value of s for which $F(s)$ must be evaluated. The real part of **s** is greater than σ_0.

Output Parameters

1: **result**

The value of the Laplace transform function $F(s)$ for the given complex value, s.

2: **sigma0** – REAL (KIND=nag_wp)

The abscissa of convergence of the Laplace transform, σ_0.

3: **sigma** – REAL (KIND=nag_wp)

The parameter σ of the Laguerre expansion. If on entry **sigma** $\leq \sigma_0$, **sigma** is reset to $\sigma_0 + 0.7$.

4: **b** – REAL (KIND=nag_wp)

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

5: **epstol** – REAL (KIND=nag_wp)

The required relative pseudo-accuracy, that is, an upper bound on $\left|f(t) - \tilde{f}(t)\right|e^{-\sigma t}$.

5.2 Optional Input Parameters

1: **mmax** – INTEGER

Suggested value: **mmax** $= 1024$ is sufficient for all but a few exceptional cases.

Default: 1024

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**.

Constraint: **mmax** ≥ 8.

5.3 Output Parameters

1: **sigma** – REAL (KIND=nag_wp)

The value actually used for σ, as just described.

2: **b** – REAL (KIND=nag_wp)

The value actually used for b, as just described.

3: **m** – INTEGER

The number of Laguerre expansion coefficients actually computed. The number of calls to **f** is $\mathbf{m}/2 + 2$.

4: **acoef**(**mmax**) – REAL (KIND=nag_wp) array

The first **m** elements contain the computed Laguerre expansion coefficients, a_i.

5: **errvec**(**8**) – REAL (KIND=nag_wp) array

An 8-component vector of diagnostic information.

errvec(1)
Overall estimate of the pseudo-error
$|f(t) - \tilde{f}(t)|e^{-\sigma t} = \mathbf{errvec}(2) + \mathbf{errvec}(3) + \mathbf{errvec}(4)$.

errvec(2)
Estimate of the discretization 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.

errvec(8)
Logarithm of the smallest nonzero expansion coefficient.

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

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

6: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Note: nag_sum_invlaplace_weeks (c06lb) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail $= 1$

On entry, **mmax** < 8.

ifail $= 2$ (*warning*)

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 $= 3$ (*warning*)

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 $= 6$ (*warning*)

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 $i = 1, 2, \dots, 5$.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

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 $\left|f(t) - \tilde{f}(t)\right| e^{-\sigma t}$.

8 Further Comments

8.1 The Role of σ_0

Nearly all techniques for inversion of the Laplace transform require you to supply the value of σ_0, the convergence abscissa, or else an upper bound on σ_0. For this function, one of the reasons for having to supply σ_0 is that the argument 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:

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

nag_sum_invlaplace_weeks (c06lb) uses the supplied value of σ_0 only in determining the values of σ and b (see Section 9.2 and Section 9.3); 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

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

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 nag_sum_invlaplace_weeks (c06lb) 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 function 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

This example computes 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 nag_sum_invlaplace_weeks (c06lb) to compute the coefficients of the Laguerre expansion, and then calls nag_sum_invlaplace_weeks_eval (c06lc) to evaluate the expansion at $t = 0$, 1, 2, 3, 4, 5.

9.1 Program Text

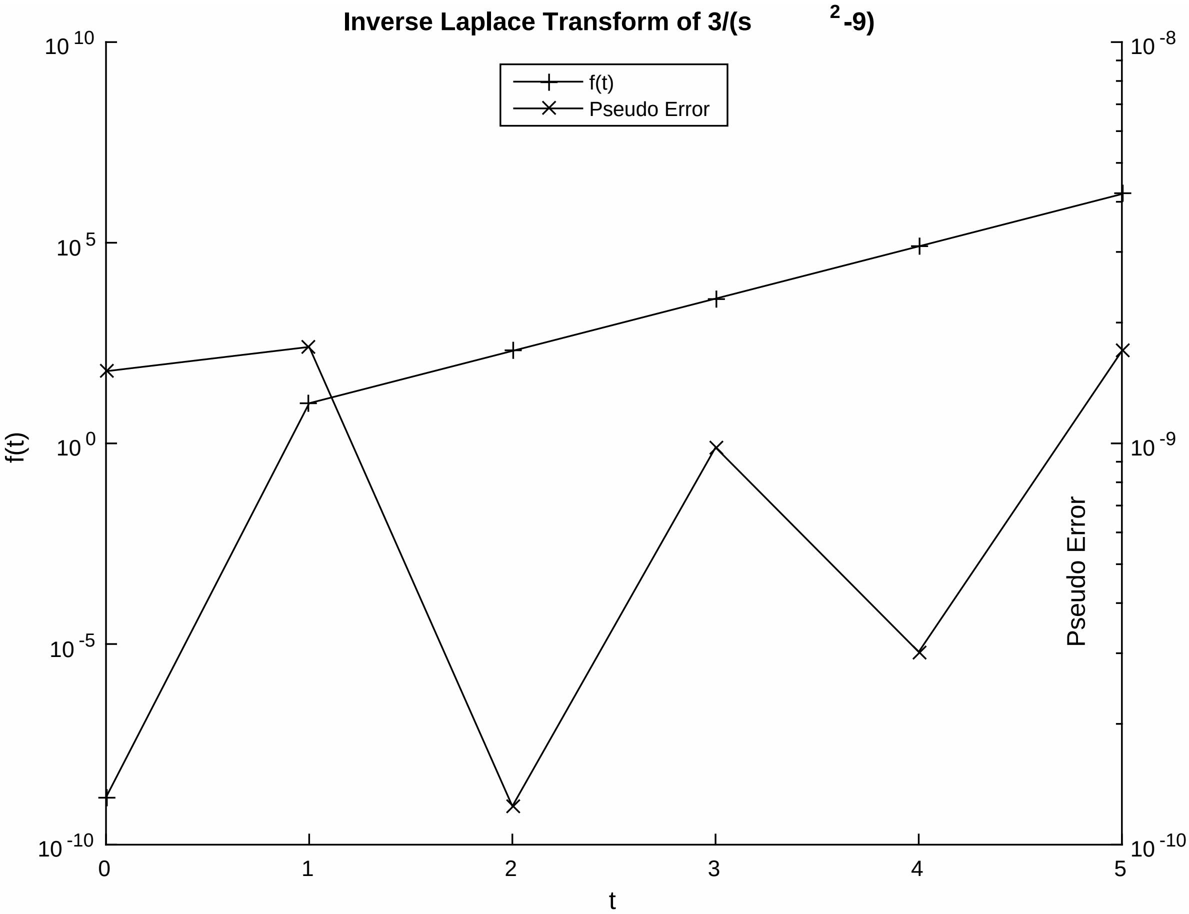
```
function c06lb_example

fprintf('c06lb example results\n\n');

% Initialize variables and arrays.
sigma0 = 3;
epstol = 1e-5;
b = 0;
sigma = 0;
n = 5;
earray = zeros(1, n+1);
jarray = zeros(1, n+1);
farray = zeros(1, n+1);
parray = zeros(1, n+1);
```

```
[sigmaOut, bOut, m, acoef, errvec, ifail] = ...
    c06lb(@f, sigma0, sigma, b, epstol);
if ifail ~= 0
    % Convergence problems.  Print message and exit.
    error('Warning: c06lb returned with ifail = %1d ',ifail);
end

% Prepare to output results.
disp(['No. of coefficients returned by c06lb = ',num2str(m)]);
disp('  ');
disp('      Computed       Exact         Pseudo');
disp('t       f(t)          f(t)          error');
% Evaluate inverse transform for different values of t.  We use c06lc,
% which calculates the transform from the coefficients given by c06lb.
for j = 0:5
    t = double(j);

    [finv, ifail] = c06lc(t, sigmaOut, bOut, acoef, errvec, 'm', m);
    if ifail ~= 0
        % Approximation is too large or too small.  Print message and exit.
        error('Warning: c06lc returned with ifail = %1d ',ifail);
    end
    exact = sinh(3.0*t);
    pserr = abs(finv-exact)/exp(sigmaOut*t);
    fprintf('%d   %10.4d   %11.4d     %8.4d\n', t, finv, exact, pserr);
    % Create arrays to be used for plotting.
    jarray(j+1) = t;
    farray(j+1) = finv;
    parray(j+1) = pserr;
end

% Plot results.
fig1 = figure;
display_plot(jarray, farray, parray)

function [f] = f(s)
% Evaluate the Laplace transform function.
f=3.0/(s^2-9.0);
if isreal(f)
    f=complex(f);
end

function plot(jarray, farray, parray)
% Use a log plot for both curves.
[haxes, hline1, hline2] = plotyy(jarray, farray, jarray, parray,...
    'semilogy','semilogy');
% Set the axis limits and the tick specifications to beautify the plot.
set(haxes(1), 'YLim', [1.0e-10 1.0e10]);
set(haxes(1), 'YMinorTick', 'on');
set(haxes(1), 'YTick', [1.0e-10 1.0e-5 1.0 1.0e5 1.0e10]);
set(haxes(2), 'YLim', [1.0e-10 1.0e-8]);
set(haxes(2), 'YMinorTick', 'on');
set(haxes(2), 'YTick', [1e-10 1e-9 1e-8]);
for iaxis = 1:2
    % These properties must be the same for both sets of axes.
    set(haxes(iaxis), 'XLim', [0 5]);
    set(haxes(iaxis), 'XTick', [0 1 2 3 4 5]);
end
set(gca, 'box', 'off'); % no ticks on opposite axes.
% Set the title.
title('Inverse Laplace Transform of 3/(s^2-9)');
% Label the x axis.
xlabel('t');
% Label the left & right y axes.
ylabel(haxes(1),'f(t)');
ylabel(haxes(2),'Pseudo Error');
% Label the curves.
legend('f(t)','Pseudo Error','location','North')
% Set some features of the three lines.
set(hline1, 'Linewidth', 0.5, 'Marker', '+', 'LineStyle', '-');
```

```
set(hline2, 'Linewidth', 0.5, 'Marker', 'x', 'LineStyle', '-');

function display_plot(jarray, farray, parray)
% Use a log plot for both curves.
[haxes, hline1, hline2] = plotyy(jarray, farray, jarray, parray,...
    'semilogy','semilogy');
% Set the axis limits and the tick specifications to beautify the plot.
set(haxes(1), 'YLim', [1.0e-10 1.0e10]);
set(haxes(1), 'YMinorTick', 'on');
set(haxes(1), 'YTick', [1.0e-10 1.0e-5 1.0 1.0e5 1.0e10]);
set(haxes(2), 'YLim', [1.0e-10 1.0e-8]);
set(haxes(2), 'YMinorTick', 'on');
set(haxes(2), 'YTick', [1e-10 1e-9 1e-8]);
for iaxis = 1:2
    % These properties must be the same for both sets of axes.
    set(haxes(iaxis), 'XLim', [0 5]);
    set(haxes(iaxis), 'XTick', [0 1 2 3 4 5]);
end
set(gca, 'box', 'off'); % so ticks aren't shown on opposite axes.
% Set the title.
title('Inverse Laplace Transform of 3/(s^2-9)');
% Label the x axis.
xlabel('t');
% Label the left & right y axes.
ylabel(haxes(1),'f(t)');
yh=ylabel(haxes(2),'Pseudo Error');
set(yh, 'position',[4.7,5e-10]);
% Label the curves.
legend('f(t)','Pseudo Error','location','North')
% Set some features of the three lines.
set(hline1, 'Linewidth', 0.5, 'Marker', '+', 'LineStyle', '-');
set(hline2, 'Linewidth', 0.5, 'Marker', 'x', 'LineStyle', '-');
```

9.2 Program Results

```
     c06lb example results

No. of coefficients returned by c06lb = 64

      Computed       Exact         Pseudo
t       f(t)          f(t)          error
0   1.5129e-09          0000     1.5129e-09
1   1.0018e+01    1.0018e+01     1.7394e-09
2   2.0171e+02    2.0171e+02     1.2471e-10
3   4.0515e+03    4.0515e+03     9.7722e-10
4   8.1377e+04    8.1377e+04     3.0221e-10
5   1.6345e+06    1.6345e+06     1.6991e-09
```


Inverse Laplace Transform of 3/(s^2-9)
f(t)
Pseudo Error
10^10
10^5
10^0
10^-5
10^-10
10^-8
10^-9
10^-10
0
1
2
3
4
5
t
f(t)
Pseudo Error