AUTOMATIC EVALUATION OF AN ABSCISSA OF CONVERGENCE FOR INVERSE LAPLACE TRANSFORM

Vu Kim Tuan
Department of Mathematics and Computer Science
Faculty of Science, Kuwait University
P.O. Box 5969, Safat, Kuwait

and

Dinh Thanh Duc
Department of Mathematics
Quynhon Teacher's Training College
Quynhon, Vietnam

Abstract. An automatic algorithm evaluating numerically an abscissa of convergence of the inverse Laplace transform is introduced.

Keywords: Numerical inverse Laplace transform, Abscissa of convergence.

AMS Mathematics Subject Classification: 44A10, 65R10.

1. Introduction

Let f be a function defined on the interval $(0, \infty)$. The integral

$$F(p) = \int_0^\infty e^{-pt} f(t) dt, \tag{1}$$

if it exists, is the Laplace transform of f. If f is locally integrable and does not grow faster than an exponential function at infinity, then there exists a number d_0, $-\infty \leq d_0 < \infty$, such that the integral (1) converges for all p with $\Re e\, p > d_0$, and diverges for all p with $\Re e\, p < d_0$. The number d_0 is called the minimal abscissa of convergence. The function $F(p)$ is an analytic function in the right half-plane $\Re e\, p > d_0$, of the order $o(\Im m\, p)$ as $p \to \infty$ in any half-plane $\Re p \geq d > d_0$, and has at least one singular point on the line $\Re e\, p = d_0$. The original function f can be recovered from the image function F by the Bromwich contour integral [12]

$$f(t) = \frac{1}{2\pi i} \int_{d-i\infty}^{d+i\infty} F(p) e^{pt} dp =: H(d), \tag{2}$$

where d is any real number such that $d > d_0$. Such a d is called an abscissa of convergence.

The problem of evaluating the inverse Laplace transform of a function is of fundamental importance in pure and applied mathematics. Since the function e^{pt} is oscillatory on the contour $(d - i\infty, d + i\infty)$, special numerical methods are required to compute the integral (2). Many computer programs for approximate evaluation of integral (2) have been developed [1, 2, 5, 6, 7, 8, 10, 11]. For a comparison of many of these methods see [3, 4]. However, none of these programs is suitable for automatic inversion. Indeed, an automatic program is intended to be used "blindly", without any analysis, but in all of these programs an abscissa of convergence d is supplied by users *a priori*, and this in general is not simple. Take, for example,

$$f(t) = \frac{1}{t}\cosh\left(\frac{ap}{2}\right)\sin\left(\frac{ap}{2}\right).$$

Then [9],

$$F(p) = \arctan\left(\frac{a}{p}+1\right) + \arctan\left(\frac{a}{p}-1\right),$$

and

$$d_0 = (|\Re e\, a| + |\Im m\, a|)/2.$$

As one can see in this example d_0 cannot be easily predicted by users without *a priori* analysis.

In practice many users have no idea about the importance of abscissae of convergence d and d_0. If f is absolutely integrable, or square integrable on $(0,\infty)$, then $d_0 \le 0$, and therefore, d can be chosen as any small positive number. But if f grows exponentially at infinity, then $d_0 > 0$, and *the problem of automatic determination of a suitable d has, as far as we know, not been investigated until now.*

If we take $d < d_0$, then the integral (2) differs from the original $f(t)$, roughly speaking, by the sum of the residues of the function $F(p)e^{pt}$ at the poles in the strip $d < \Re e\, p \le d_0$, if there is no singular point of other kind there. However, not every abscissa of convergence $d > d_0$ is suitable for numerical purpose. If $d_2 > d_1 > d_0$, then although both of the integrals $H(d_1)$ and $H(d_2)$ give the exact value of $f(t)$, the numerical evaluation of the integral $H(d_2)$ would give a relative error of magnitude $e^{(d_2-d_1)t}$ larger than the relative error of the numerical evaluation of the integral $H(d_1)$, when the same numerical scheme is used. Therefore, in order to evaluate $f(t)$ for $t \in (0,T)$ from (2) numerically with high accuracy, an abscissa of convergence d should be chosen so that $(d-d_0)T$ is relatively small.

In this paper an algorithm to find an abscissa of convergence $d > d_0$, but close to d_0, is proposed. Once an abscissa of convergence d that is a good approximation to the exact abscissa of convergence d_0 is found, one can apply one of the programs described in the cited papers to evaluate the inverse Laplace transform (2).

2. Mathematical Background

Our algorithm is based on the following observation:

If F is the Laplace transform of a function f, and d_0 is the minimal abscissa of convergence, then for any $d > d_0$ we have [9]

$$\frac{1}{2\pi i}\int_{d-i\infty}^{d+i\infty}\frac{F(p)}{p^k}e^{pt}dp = \frac{1}{\Gamma(k)}\int_0^t (t-y)^{k-1}f(y)dy, \quad k \ge 1. \tag{3}$$

Putting $t=0$ in (3), we get

$$I(d) := \frac{1}{2\pi i}\int_{d-i\infty}^{d+i\infty}\frac{F(p)}{p^k}dp = 0, \quad k \ge 1. \tag{4}$$

Hence, if $d > d_0$ the integral (4) equals 0. If $d \leq d_0$, then in the strip $d \leq \Re e\, p \leq d_0$ there should be some singular points of $F(p)$, otherwise d_0 is not the minimal abscissa of convergence. Therefore, the integral (4) in this case is in general not equal to 0. Thus, equality of the integral in (4) to zero is necessary for d to be an abscissa of convergence. But it is not a sufficient condition. If d is not an abscissa of convergence, it may still happen that the integral (4) vanishes. It is the case when the sum of the residues of all the poles in the strip $d \leq \Re e\, p \leq d_0$ is zero. To get a numerically sufficient condition, we consider instead of the integral (4) the modified integral

$$J(d) := \int_{d-i\infty}^{d+i\infty} \frac{F(p)}{p^2(p+ih)F(d+ih)}\, dp, \tag{5}$$

where h is a real number, randomly selected by the program. The role of the divisor $F(d+ih)$ will be explained later. Since $\frac{F(p)}{p+ih}$ is the Laplace transform of the function $e^{-iht}\int_0^t f(y)e^{ihy}\, dy$ (see [9]), the integral (5) equals zero if d is an abscissa of convergence. Because h is chosen randomly, the probability that the sum of the residues of all the poles of the function $\frac{F(p)}{p^2(p+ih)}$ in the strip $d \leq \Re e\, p \leq d_0$ equals 0 is practically zero. Therefore, the equality of the integral (5) to zero practically guarantees that d is an abscissa of convergence.

The Bromwich countour integral (2) may converge slowly due to the oscillatory term e^{pt}, and therefore requires special numerical integration techniques. The integral (5) does not have an oscillatory term e^{pt}, and since the function $F(p)$, as an image of the Laplace transform of a function, is of the order $o(\Im m\, p)$ as $\Im m\, p$ tends to $\pm\infty$, the integral (5) converges fast. Hence, numerically, the integral (5) is very easily computed (for example, by a Gaussian quadrature).

Observe that $J(d)$ is a piecewise constant function, with $J(d) \neq 0$ for $d < d_0$, and $J(d) = 0$ for $d > d_0$. To find d close to d_0, say $d - d_0 < q$ for some tolerance error q, we find d_1 with $0 < d - d_1 < q$, such that $J(d) = 0$, but $J(d_1) \neq 0$. In that case $d_1 \leq d_0 < d$, and therefore, $d - d_0 < q$. It can be done by using, say, interval bisection. Because of floating-point arithmetic, in order to compare $J(d)$ with zero, the normalizing factor $F^{-1}(d+ih)$ is introduced, and some error tolerance ϵ should be specified: $J(d)$ is assumed to be zero if $|J(d)| \leq \epsilon$. This tolerance error ϵ depends on the computer and the numerical integration program in use.

3. Algorithm

Input parameters: The user provides parameters $Bound, q, \epsilon > 0$. An abscissa of convergence d is sought in the interval $(0, Bound)$.

Bound – the maximum range of abscissa of convergence allowed: $d \leq Bound$.

q – the tolerance error for d: $d - d_0 < q$ if $d_0 \geq 0$, and $d < q$ if $d_0 < 0$. For most of applications we can select $q = 1$.

ϵ – the tolerance error for numerical integration.

The algorithm consists of the following steps :

Step 1. Select $h \in (1, 2)$ randomly. Put $d = Bound$ and $d1 = 0$.

Step 2. If $|J(Bound)| > \epsilon$, exit: either the abscissa of convergence exceeds the maximum range *Bound* allowed, or F is not a Laplace transform of any function.

Step 3. Put $d2 = (d + d1)/2$.
If $|J(d2)| \leq \epsilon$, put $d = d2$.
Otherwise, put $d1 = d2$.
Step 4. If $d - d1 \geq q$, go to Step 3.
Otherwise, exit: d is an abscissa of convergence in the interval $(d_0, d_0 + q)$, if $d_0 \geq 0$, and in the interval $(0, q)$, if $d_0 < 0$.

As it can be seen, if d_0 is large, the integral (5) has to be evaluated for several d, and the algorithm becomes expensive. Still, the main cost of the computation is due to the evaluation of the inverse Laplace integral (2), since integral (5) can be computed with machine accuracy, while integral (2) for large d_0 should be computed with much higher accuracy (for example, with double precision), and several times for different $t \in [0, T]$.

4. Numerical Experiment

The following code was written and tested on MATHEMATICA 3.0. Numerical integration is carried out in 16-digit floating-point arithmetic and returns roughly 6 accurate decimal digits. Thus, a tolerance error $\epsilon = 5.10^{-7}$ can be assumed, and $J(d) = 0$ if the magnitude of $J(d)$ less than or equal to 5.10^{-7}.

(* Begin Package *)

Clear[AbscissaInverseLaplaceTransform];

nilt::singularity = " Warning: The function is suspected not to be a Laplace transform of any function or the function may have a singular point with the real part larger than '1'. Change the upper bound '1'."; (* This message will be displayed in case the minimal abscissa of convergence exceeds the *Bound* provided by the user.*)

AbscissaInverseLaplaceTransform[$F\_, p\_, Bound\_, q\_$] :=

Module[{ d, d1, d2, f0, f1, tmp },

$$
\begin{aligned}
d1 &= 0; \\
d &= N[Bound]; \\
h &= \text{Random[Real}, \{1, 2\}]; \\
&\quad (*\,\text{h is selected randomly by MATHEMATICA 3.0 from the interval } (1, 2).*) \\
f0 &= N[(F)/.p- > d + Ih, 6]; \\
f1 &= f0\hat{}(-1)(F/(p * p * (p + Ih)))/.\, p- > d + Ix; \\
tmp &= \text{NIntegrate}[f1, \{x, -\text{Infinity}, \text{Infinity}\}];
\end{aligned}
$$

If[Abs[tmp] > 5 * 10^(−7),
Message[nilt :: singularity, Bound];
Return[$Failed]
];
While[$d - d1 > q$,

$d2 = (d + d1)/2;$
$f0 = N[(F)/.\, p- > d2 + Ih, 6];$
$f1 = f0\hat{}(-1)(F/(p * p * (p + Ih)))/.\, p- > d2 + Ix;$
$tmp = \text{NIntegrate}[f1, \{x, -\text{Infinity}, \text{Infinity}\}];$
$\text{If}[\text{Abs}[tmp] > 5.10\hat{}(-7),$
$d1 = d2,$
$d = d2$
]
];
d
]

(* End Package *)

We experiment the code with $q = 0.5, 1$ and $Bound = 20, 50, 100$, for functions

$$F(p) = \frac{1}{(p-a)^b}, \qquad d_0 = a,$$

$$F(p) = \arctan\left(\frac{a}{p} + 1\right) + \arctan\left(\frac{a}{p} - 1\right), \qquad d_0 = (|\Re a| + |\Im a|)/2,$$

with several values of a and b. All abscissae of convergence d obtained belong to the interval $(0, q)$, if $d_0 < 0$, and to the interval $(d_0, d_0 + q)$, if $d_0 \geq 0$.

The authors are grateful to Professor C. Grossmann (Dresden) and an anonymous referee, whose critical remarks were essential in improving the presentation of the paper.

References

[1] J. Abate and W. Whitt, The Fourier-series method for inverting transforms of probability distributions, *Queueing Systems* 10(1992), 5-88.

[2] A.H-D. Cheng, P. Sidauruk, and Y. Abousleiman, Approximate inversion of the Laplace transform, *The Mathematica Journal,* 4(1994), no. 2, 76-82.

[3] B. Davies, B. Martin, Numerical inversion of the Laplace transform: a survey and comparison of methods, *J. Comput. Phys.* 33(1979), 1-32.

[4] D.G. Duffy, On the numerical inversion of Laplace transforms: Comparison of three new methods on characteristic problems from applications, *ACM Trans. Math. Softw.* 19(1993), no. 3, 333-359.

[5] G. Honig and U. Hirdes, Algorithm 27: A method for the numerical inversion of Laplace transforms, *J. Comput. Appl. Math.* 10(1984), 113-132.

[6] A. Murli and M. Rizzardi, Algorithm 682: Talbot's method for the Laplace inversion problem, *AMS Trans. Math. Softw.* 16(1990), no. 2, 158-168.

[7] R. Piessens, Algorithm 453: Gaussian quadrature formulas for Bromwich's integral [D1], *Comm. ACM* 6(1973), no. 8, 486-487.

[8] R. Piessens and R. Huysmans, Algorithm 619: Automatic numerical inversion of the Laplace transform [D5], *ACM Trans. Math. Softw.* 10(1984), 348-353.

[9] A.P. Prudnikov, Yu.A. Brychkov, and O.I. Marichev, *Integrals and Series. Volume 4: Direct Laplace Transforms*, Gordon and Breach, Reading, 1992.

[10] H. Stehfest, Algorithm 368: Numerical inversion of Laplace transforms [D5], *Comm. ACM* 13(1970), no. 1, 47-49.

[11] F. Veillon, Algorithm 486: Numerical inversion of Laplace transform [D5], *Comm. ACM* 17(1974), no. 10, 587-589.

[12] D.V. Widder, *The Laplace Transform*, Princeton, Princeton Univ. Press, 1972.