Notes on Numerical Laplace Inversion

Kathrin Spendier

April 12, 2010

1 Introduction

The main idea behind the Laplace transformation is that we can solve an equation (or system of equations) containing differential and integral terms by transforming the equation in time (t) domain into Laplace (ϵ) domain. For example, we can use Laplace transforms to turn an initial value problem into an algebraic problem which is easier to solve. After we solved the problem in Laplace domain we find the inverse transform of the solution and hence solved the initial value problem. The Laplace transform of $f(t)$ is:

$$\tilde{f}\left(\epsilon\right)=\int_{0}^{\infty}e^{-\epsilon t}f\left(t\right)dt, \tag{1}$$

where ϵ is a complex variable known as the Laplace variable. The inverse integral is defined as the Bromwich contour integral ($\epsilon\rightarrow\gamma+i\infty$) as:

$$f\left(t\right)=\frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty}e^{\epsilon t}\tilde{f}\left(\epsilon\right)d\epsilon. \tag{2}$$

γ is chosen so the all singular points of $\tilde{f}\left(\epsilon\right)$ lie to the left of the line $\mathcal{R}e(\epsilon)=\gamma$ in the complex ϵ-plane.

In simple cases the inverse transform can be found via analytical methods or with the help of tables. You can also compute the Laplace transform by evaluation of the complex integral of inverse transformation. Unfortunately, it is not always easy to find the inverts. One possible reason is that the inverse is not a named function or can not be represented by a "simple" formula. Moreover, if the Laplace transform is computable or measurable on the real and positive axis only the problem is ill-posed. Two time domain functions which differ at a single point in time for example will have the same transform. This case is very complicated simply because of absence of the exact inversion formula. In these cases a numerical method must be used. There are several numerical algorithms in literature and each individual method has its own applications and is suitable for a particular type of functions [1]. In this report I will briefly introduce the Fourier Series expansion and the Gaver-Stehfest method.

1.1 What is the Challenge of Numerical Laplace Inversion?

The numerical inversion of $f(t)$ depends on the sensitivity of the inversion procedure. This is clear when we consider the need to multiply by a potentially increasing large exponent $e^{\epsilon t}$ in Eq.2. Algorithmic and finite precision errors (i.e. increasing round off error for large numbers) can lead to exponential divergence of numerical solutions.

2 Fourier Series Expansion

This method is based on the Bromwich contour inversion integral, which can be expressed as the integral of a real valued function of a real variable by choosing a specific contour. One first converts the inversion integral into the Fourier transform and then approximates the transform by a Fourier series (use trapezoidal rule) with a specific discretization error.

As mentioned, the method [2] utilizes the standard Bromwich contour $\epsilon \rightarrow \gamma + i\omega$ (choose a specific contour) to rewrite the Laplace transform integral given in Eq.2 as:

$$f\left(t\right) = \frac{e^{\gamma t}}{2\pi} \int\limits_{\infty}^{\infty} e^{i\omega t} \tilde{f}\left(\gamma + i\omega\right) d\omega, \tag{3}$$

which is in the form of a Fourier transform. We can even go further and rewrite $e^{i\omega t}$ in Eq.3 to obtain:

$$f(t) = \frac{e^{\gamma t}}{2\pi} \int\limits_{\infty}^{\infty} \left[\cos\left(\omega t\right) + i \sin\left(\omega t\right)\right] \tilde{f}\left(\gamma + i\omega\right) d\omega. \tag{4}$$

Equation 4 can then be rewritten in real and imaginary parts

$$f(t) = \frac{e^{\gamma t}}{2\pi} \int\limits_{\infty}^{\infty} \Big[\mathcal{R}\mathrm{e}\left\{\tilde{f}\left(\gamma + i\omega\right)\right\} \cos\left(\omega t\right) - \mathcal{I}\mathrm{m}\left\{\tilde{f}\left(\gamma + i\omega\right)\right\} \sin\left(\omega t\right)\Big] d\omega. \tag{5}$$

With the assumption that $f(t)$ is non-negaive, $f(-t) = 0$ for $t > 0$, and $\mathcal{R}e(\tilde{f})$ is even and $\mathcal{I}m(\tilde{f})$ is odd we are only interested in the real part and we may write:

$$f(t) = \frac{2e^{\gamma t}}{\pi} \int\limits_{0}^{\infty} \mathrm{Re}\left\{\tilde{f}\left(\gamma + i\omega\right)\right\} \cos\left(\omega t\right) d\omega. \tag{6}$$

Equation 6 can now be approximated by the trapezoidal rule

$$\int\limits_{a}^{b} f\left(x\right) dx \approx \frac{b-a}{n} \left(\frac{f\left(a\right) + f\left(b\right)}{2}\right) + \sum_{k=1}^{n-1} f\left(a + k\frac{b-a}{n}\right)$$

with a step size $h = \frac{b-a}{n}$:

$$f(t) \approx \frac{he^{\gamma t}}{\pi} \mathrm{Re}\left\{\tilde{f}\left(\gamma + i\omega\right)\right\} + \frac{2he^{\gamma t}}{\pi} \sum_{k=1}^{\infty} \mathrm{Re}\left\{\tilde{f}\left(\gamma + ikh\right)\right\} \cos\left(kht\right), \tag{7}$$

and can be further simplified if $h = \frac{\pi}{2t}$ and $\gamma = \frac{A}{2t}$ and written as a nearly alternating series:

$$\boxed{f(t) \approx \frac{e^{A/2}}{2t}\mathrm{Re}\left\{\tilde{f}\left(\frac{A}{2t}\right)\right\} + \frac{e^{A/2}}{t}\sum_{k=1}^{\infty}(-1)^k\,\mathrm{Re}\left\{\tilde{f}\left(\frac{A+2k\pi i}{2t}\right)\right\}.} \tag{8}$$

Equation 8 can now be computed numerically by summing over a finite number of k. Hence, A and k are parameters which must be optimized for increasing accuracy. This method turns out to be very accurate. The disadvantage of this method is that it is difficult to implement and requires a large computation time [3] (i.e. sum over many terms).

3 Gaver-Stehfest Method

This method requires sampling of the Laplace space function $\tilde{f}(\epsilon)$ only on the real line and is explained in detail in Refs [5] and [6]. Similarly to the Fourier Method, we make the following transformation in Eq.2. We define a new complex variable $z = \epsilon t$ and rewrite Eq.2 as:

$$f(t) = \frac{1}{2\pi i t}\int_{C'}\tilde{f}\left(\frac{z}{t}\right)e^z dz, \tag{9}$$

where C' is the same contour as in Eq.2 as a function of z. Next we approximate e^z by a rational function

$$e^z \approx \sum_{k=0}^{n}\frac{\omega_k}{\alpha_k - z},$$

where ω_k and α_k are complex numbers and called weights and nodes respectively. Using this approximation and applying the Cauchy integral formula we obtain

$$\boxed{f(t) \approx \frac{1}{t}\sum_{k=0}^{n}\omega_k\tilde{f}\left(\frac{\alpha_k}{t}\right).} \tag{10}$$

Equation 10 approximates the inverse Laplace transform by a linear combination of transform values. The nodes and weights are complex numbers but do not depend on $\tilde{f}$ of the function argument t, but typically depend upon n. Since the weights are initially left unspecified it is typically called a framework rather than algorithm. Hence, Eq. 10 is referred to as *Unified Framework for Numerical Laplace Inversion* [4] since many different algorithms can be but into this framework, even the Fourier method (or Euler) for which the nodes are complex values, Eq.8 [4].

The Gaver-Stehfest method considers the case in which $f(t)$ is real-valued and the weights and nodes are real which leads to very accurate result for functions of type $e^{-\alpha t}$. The Stehfest's algorithm is based on a probabilistic derivation [5] and approximates the time domain solution using the following equation [1,4]:

$$\boxed{f(t) \approx \frac{\ln(2)}{t}\sum_{k=1}^{2M}\omega_k\tilde{f}\left(\frac{k\ln(2)}{t}\right)}, \tag{11}$$

with the weights

$$\omega_k = (-1)^{M+k} \sum_{j=\lfloor (k+1)/2 \rfloor}^{min(k,M)} \frac{j^{M+1}}{M!} \begin{pmatrix} M \\ j \end{pmatrix} \begin{pmatrix} 2j \\ j \end{pmatrix} \begin{pmatrix} j \\ k-j \end{pmatrix}$$

where $\lfloor x \rfloor$ being the greatest integer or less than or equal to x. I implemented Eq. 11 and 12 in Matlab using double precision, which restricts M to be less than 7. Hence, the Gaver-Stehfest method only evaluates the function at real and positive values of the Laplace variable ϵ and sums a total of 14 $(2M)$ terms. This method is easy to implement and very accurate for functions of type $e^{-\alpha t}$. For functions with oscillatory behavior in time domain, the Gaver-Stehfest algorithm fails.

3.1 Some Numerical Inversion Examples implemented in Matlab

In Fig.1, I inverted $\tilde{f}(\epsilon) = 1/(\epsilon + 1)$ numerically and compared it to the known analytic solution in time domain $f(t) = e^{-t}$. As expected the Graver-Stehfest method approximates the solution very well.

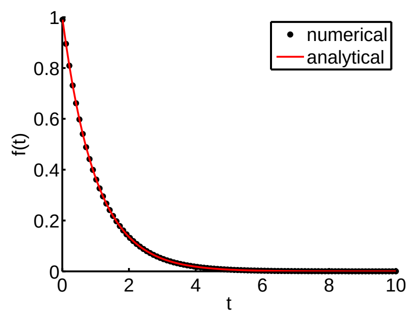

Figure 1: Numerical Inversion (black dots) compared to analytic solution (red line) for $f(t) = e^{-t}$.

In Fig.2, I inverted $\tilde{f}(\epsilon) = 1/(\epsilon^2 + 1)$ numerically and compared it to the known analytic solution in time domain $f(t) = sin(t)$. This is an example for which the Graver-Stehfest method fails. Only for short time we obtain a good approximation.

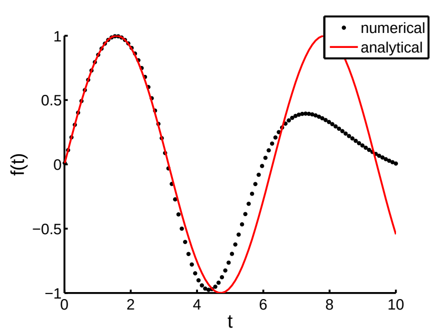

Figure 2: Numerical Inversion (black dots) compared to analytic solution (red line) for $f(t) = sin(t)$.

4 References

[1] Hassan Hassanzadeh, and Mehran Pooladi-Darvish (2007), *Comparison of different numerical Laplace inversion methods for engineering applications*, Applied Mathematics and Computation, 189:1966-1981.
[2] Joseph Abate (1995), *Numerical Inversion of Laplace Transforms of Probability Distributions*, ORSA Journal on Computing, 7:36-43.
[3] Kenny S. Crump (1976), *Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation*, Journal of the Association for Computing Machinery, 23:89-96.
[4] Joseph Abate and Ward Whitt (2006), *Unified Framework for Numerically Inverting Laplace Transforms*, INFORMS Journal on Computing, 18:408-421.
[5] D. P. Gaver Jr. (1966), *Observing Stochastic Processes and Approximate Transform Inversion*, Operations Research, 14:444-459.