

The CME method: Efficient numerical inverse Laplace transformation with Concentrated Matrix Exponential distribution

Salah Al-Deen Almousa*, Gábor Horváth*, Illés Horváth†, András Mészáros*, Miklós Telek*†

* Budapest University of Technology and Economics, Hungary. E-mail: {almousa,ghorvath,maszerosa,telek}@hit.bme.hu

† MTA-BME Information Systems Research Group, Budapest, Hungary. E-mail: horvath.illes.antal@gmail.com

ABSTRACT

Numerical inverse Laplace transformation (NILT) is an important tool in the field of system modelling and performance analysis. The recently introduced CME method has many important advantages over the alternative numerical inverse Laplace transformation (NILT) methods. It avoids Gibbs oscillation (i.e., does not generate overshoot and undershoot), preserves the monotonicity of functions, its accuracy is gradually improving with the order, and it is numerically more stable than the alternative methods. In this paper we demonstrate these advantages and introduce our tool which implements the CME method and other popular NILT methods.

1. INTRODUCTION

Due to their analytical simplicity, Laplace transforms are widely used in various scientific fields [26]. The main difficulty in working with Laplace transforms is associated with the difficulties in obtaining time domain results from the Laplace transform descriptions. In a small subset of practically interesting cases it is possible to symbolically inverse transform the Laplace domain description, but in the majority of the cases symbolic inverse Laplace transformation is not available and numerical inverse Laplace transformation (NILT) remains the only feasible way of the analysis.

For several decades, NILT procedures were assumed to be unstable, numerically sensitive and unreliable. One reason for this bad reputation comes from the fact that low order NILT methods were inaccurate and high order NILT methods were unstable (often with weird alternation in the results, which is often referred to as Gibbs oscillation) and in many cases, the range of orders with reasonably accurate results was missing between these two extreme behaviours.

In spite of the shortcomings of NILT methods, the benefits of Laplace domain analysis gave continuous impetus for improving NILT methods. Plenty of NILT methods were developed and published during the last 4 decades (for surveys, we refer to [19, 31], and to those referenced therein).

Apart of many other scientific fields where Laplace transforms are efficiently applied [26], the field where Laplace transforms are most often applied is the stochastic performance modelling and analysis of computer and communication systems. As some examples of the measures of interest that is computed based on Laplace transform domain

description we can note security [12], reliability [27, 9], dependability [22, 15], performance [11], temporal logic kind of measures, like “what is the probability that a random event occurs before time T ”, which can be obtained from the transient and temporal measures of such systems [3, 4, 5].

In this work we demonstrate that the CME method is applicable in a rather wide range of practically interesting cases, such that it eliminates the mentioned weaknesses of the previously applied methods. Additionally we present a tool for NILT that implements the CME method and two other popular NILT methods, the Euler and the Gaver–Stehfest method. The offline version of the tool is available in multiple popular scientific languages. The online version can be used through an intuitive GUI and can be used for demonstration purposes and for quick and simple calculations.

The rest of the paper is organized as follows. Section 2 gives an overview of the common features of the most popular NILT methods and Section 3 introduces the CME method. Section 4 provides the details of the NILT software tools. Section 5 demonstrates the properties of the CME method. Finally, Section 7 concludes the paper.

2. NILT METHODS

For a real or complex valued function $h(t)$, the Laplace transform is defined as

$$h^*(s) = \int_{t=0}^{\infty} e^{-st} h(t) dt. \quad (1)$$

The goal of NILT is to find an approximate value of h at point T (i.e., $h(T)$) based on $h^*(s)$.

Many, seemingly different NILT methods were unified in a general framework by Abate and Whitt in [2]. In the Abate–Whitt framework, a finite linear combination of the transform values approximate h , via

$$h(T) \approx \sum_{k=1}^N \frac{\eta_k}{T} h^* \left(\frac{\beta_k}{T} \right), \quad T > 0, \quad (2)$$

where the nodes β_k ($1 \leq k \leq N$) and weights η_k ($1 \leq k \leq N$) are real or complex numbers that depend on the N , but not on the transform function h^* or the time argument T .

In order to have a real approximation in (2), the list of nodes β_k and weights η_k must contain only real values and complex conjugate pairs. Denoting the set of real nodes, complex nodes with positive imaginary part and complex nodes with negative imaginary part by $R = \{k : \beta_k \in R\}$,

$C^+ = \{k : \text{Im}(\beta_k) > 0\}$, and $C^- = \{k : \text{Im}(\beta_k) < 0\}$, we have

$$h(T) \approx \sum_{k \in R} \frac{\eta_k}{T} h^* \left(\frac{\beta_k}{T} \right) + \sum_{k \in C^+} 2\text{Re} \left(\frac{\eta_k}{T} h^* \left(\frac{\beta_k}{T} \right) \right), \quad (3)$$

which means that that h^* needs to be evaluated only $n = |R| + |C^+|$ times instead of $N = |R| + |C^+| + |C^-|$ times as it is in (2). Therefore we call n (instead of N) the order of the approximation. For the sake of simplicity, we will assume that $R \cup C^+ = \{1, 2, \dots, n\}$ in the following.

NILT methods outside the Abate–Whitt framework approximate $h(T)$ based on different approaches, e.g., using the derivatives or the series expansion of h^* . NILT methods of the Abate–Whitt framework turned out to be more efficient than the methods outside the Abate–Whitt framework and have become predominant.

Indeed, (2) defines a rather simple NILT procedure composed of the following steps

1. compute the nodes β_k ($1 \leq k \leq n$) and the weights η_k ($1 \leq k \leq n$) according to the applied Abate–Whitt framework method,
2. compute h^* in points β_k/T ($1 \leq k \leq n$),
3. sum up the results according to (2).

Step 1: Before the introduction of the CME method, the widely used Abate–Whitt framework methods computed the nodes and the weights based on a relatively easy to compute explicit formula. In many of these methods the weights have extremely large absolute value.

Step 2: The complexity of this step depends on h^* . There are practically interesting cases, where h^* does not exhibit a closed form, thus numerical integrals (similar to (1)) need to be evaluated in order to obtain h^* in the required points.

Step 3: The summation in (2) can be numerically sensitive when the weights have extremely large absolute value.

For the most commonly used methods in the Abate–Whitt framework, such as Euler method [1], the Gaver–Stehfest method [13, 28], and the Talbot method [29, 30], $\max_{1 \leq k \leq n} |\eta_k|$ increases exponentially with the order of the method.

As an example, the nodes and weights of the Euler method are:

$$\beta_k = \frac{(n-1)\ln(10)}{6} + \pi i(k-1), \quad 1 \leq k \leq n,$$

$$\eta_k = 10^{(n-1)/6} (-1)^k \xi_k, \quad 1 \leq k \leq n,$$

where

$$\xi_1 = \frac{1}{2},$$

$$\xi_k = 1, \quad 2 \leq k \leq (n+1)/2,$$

$$\xi_n = \frac{1}{2^{(n-1)/2}},$$

$$\xi_{n-k} = \xi_{n-k+1} + 2^{-(n-1)/2} \binom{(n-1)/2}{k},$$

$$\text{for } 1 \leq k < (n-1)/2.$$

3. THE CME METHOD

The CME method [18] also belongs to the Abate–Whitt framework. The major difference between the CME method

Version	Online	Offline
Languages	JavaScript	Matlab, IPython Mathematica
Precision	Machine	Arbitrary, Machine
Output	Visual	Numerical, Visual
Method Selection	Interactive	Preselected
Multi-function Support	Yes	Yes
Complex No. Support	Yes	Yes
Higher-Dimension Support	No	Yes
Supported Order	Up to 400	Up to 1000
Targeted Users	General	Researchers
Needed Resources	Browser	Computer

Table 1: A comparison between the online and the offline implementations

and the previously introduced Abate–Whitt framework methods is in the computation of the nodes and the weights. The nodes and the weights are not computed at every application of the NILT procedure, but they are computed in advance for all required order n and the stored nodes and weights are used for the computation of (2).

The CME method is based on the trigonometric – exponential relation

$$f_N(t) = c e^{-\lambda t} \prod_{j=1}^n \cos^2 \left(\frac{\omega \lambda t - \phi_j}{2} \right) = \sum_{k=0}^{N-1} \eta_k e^{-\beta_k t}, \quad (4)$$

whose details are provided in [16]. By construction the $f_N(t)$ of the CME method is non-negative.

Based on this relation, the nodes and the weights are defined via the solution of the optimization problem

$$(\omega^*, \phi_1^*, \dots, \phi_n^*) = \arg \min_{\omega, \phi_1, \dots, \phi_n} \text{SCV} \left(e^{-t} \prod_{j=0}^n \cos^2(\omega t - \phi_j) \right) \quad (5)$$

where the squared coefficient of variation (SCV) is defined as

$$\text{SCV} := \frac{\int_{t=0}^{\infty} t^2 f_N(t) dt \int_{t=0}^{\infty} f_N(t) dt}{\left(\int_{t=0}^{\infty} t f_N(t) dt \right)^2} - 1 \quad (6)$$

The SCV is measure of the difference between the unit impulse and $f_N(t)$, and it is insensitive to parameters c and λ , that is why they are neglected in (5). For large orders, $100 < n < 2 \cdot 10^5$, quasi-optimal solutions of this optimization problem is proposed in [7].

4. SOFTWARE SUPPORT

To provide software support for NILT, we created a tool that implements the CME method along with the Euler and the Gaver–Stehfest methods. The tool is available in two versions at [17] working under the UIUC distribution license

Figure 1: Custom Laplace transform function definition with the NILT parameters setup

[25]: offline version and online version. Both are available through the webpage of the project [17]. The tool has the following input parameters:

1. Laplace transform function to invert
2. list of evaluation points (T)
3. maximum number of evaluations (the maximum order n to use)
4. NILT method to use (CME by default)
5. precision of calculations (machine precision by default)

The generated output is the value(s) of the numerical-based inverse Laplace transform at point(s) given by T , presented either as a list or in a visual format using a plot.

The offline version is directed more toward researchers; that is why it is made available in multiple popular scientific languages: Matlab, Mathematica, and IPython. This mode offers more control options for the user, like which calculation precision (machine, arbitrary) needs to be applied, integration with other ongoing research work, extracting the results in visual and numerical formats. Additionally, it offers the option of abscissa shifting, which can be used for more precise approximation of the tail behaviour using the recommendations in [18]. This version supports the CME method for orders up to $n = 1000$. Up to this value standard double precision (i.e., machine precision in most environments) calculation is generally sufficient.

Aside from the file containing the implementation of the NILT methods, the offline version requires that the "iltcme.json" file be placed into the same directory. This file stores the pre-computed parameters that are used to calculate the β_k nodes and the η_k weights for selected orders of the CME distributions in double precision.

Recently, more than 15 research papers in various fields like hydrologic processes [24], water resources [20], control theory [8], and especially in stochastic theory and modeling [10, 14], have accredited CME NILT results based on our tool. We also applied it successfully in the analysis of fluid queues [4, 6], QBDs [5], and vehicular ad-hoc networks [21].

The online version allows the user to directly demonstrate the CME NILT within the browser via a JavaScript GUI. This version offers to perform NILT for six pre-defined Laplace transform functions (exponential, sine, heavy-side, exponential heavy-side, periodic square wave, staircase) or

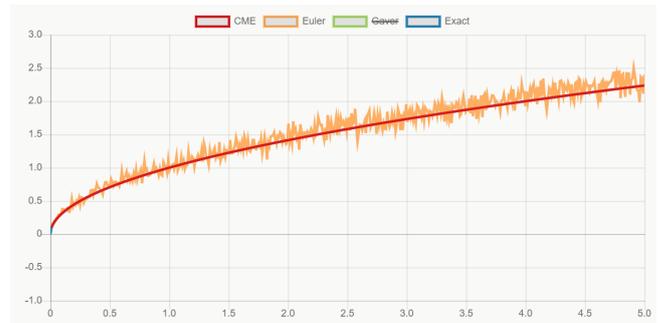


Figure 2: NILT of $\sqrt{\pi}/2s^{3/2}$ by the CME, Euler methods with $n = 100$

a custom Laplace transform function written in an appropriate syntax (matrices and complex numbers are also supported in the expression). The GUI comes with options to customize the axis boundaries and the number of steps for the x-axis. Moreover, the effect of changing the number of function evaluations (order n) on the NILT accuracy is illustrated interactively. During the demonstration, and simply with a few clicks on the legend bar, the user can also visually compare the CME method with the exact inverse Laplace transform (if known) as well as with the Gaver–Stehfest and the Euler method.

Table 1 concludes the features provided by the two versions.

5. APPLICATION EXAMPLES

In this section, we will demonstrate the advantages of the CME method over the classical methods under Abate–Whitt framework and display the features offered by the GUI interface. Numerical and visual examples for the offline version were discussed in detail in [18].

In the first example, shown in Figure 1, we use the custom function option to inverse transform $h^*(s) = \sqrt{\pi}/2s^{3/2}$, with $T = [0, 5]$ with 500 steps (i.e., 0.01 step size). The y-axis boundaries are set to be automatic. The tool provides the $h(t) = \sqrt{t}$ inverse transform pair of $h^*(s)$, since it can be obtained analytically. We set the NILT order (maximum number of function evaluation) to $n = 100$.

Figure 2 shows that the CME method generates smooth, stable, and accurate NILT results based on the comparison with the exact inverse, while the Euler method suffers from numerical instability, which manifests in significant noise. The Gaver method gives completely false results due to its even greater instability, therefore it was excluded from the figure by clicking on its legend.

To demonstrate the difference in Gibbs oscillation, in the second example we use the built-in exponential Heaviside function $h(t) = I_{t \geq 1}(t)e^{1-t}$ (in which $I_{t \geq 1}(t)$ is the indicator function of $t \geq 1$) and its Laplace transform $h^*(s) = \frac{e^{-s}}{1+s}$. Figure 3 shows the NILT results for the different methods. The CME method gives the most accurate result in practically all points, which is a common observation for functions with discontinuity. The Gaver–Stehfest method follows the jump quite slowly, while the Euler method follow it faster, but both result in significant under- or overshoot due to the Gibbs oscillation. This oscillation is inherent to the Gaver–Stehfest and Euler methods and has nothing to do

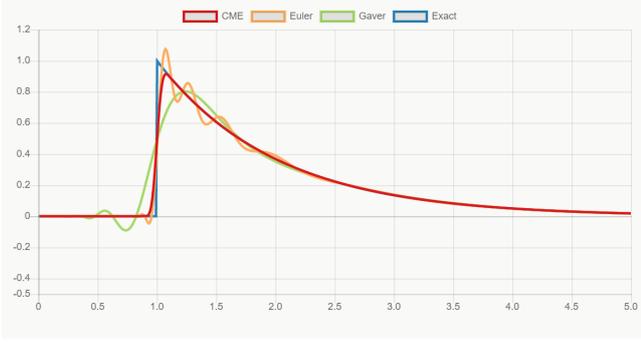


Figure 3: NILT of the Heaviside exponential function with $n = 20$

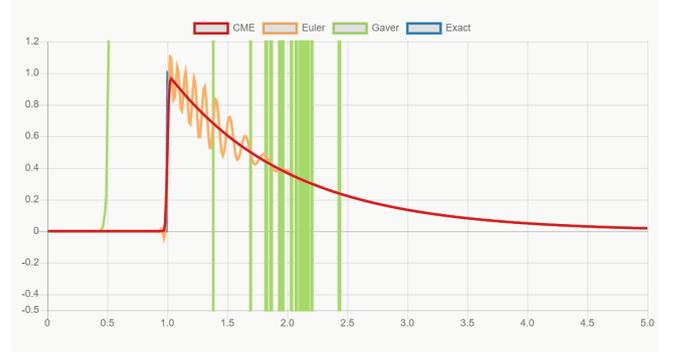


Figure 4: NILT of the Heaviside exponential function with $n = 52$

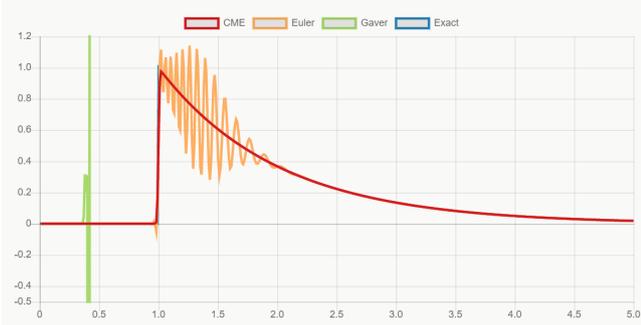


Figure 5: NILT of the Heaviside exponential function with $n = 72$

with the numerical precision used. It would seem natural that increasing the order of the methods should improve the accuracy, since it increases the number of terms used to approximate $h(t)$ according to (2). Figure 4 shows that this is not the case, however. By increasing the order to $n = 52$, only the CME method becomes more accurate. The Euler method follows the jump in the exact solution faster, but at the cost of significant Gibbs oscillation, while the Gaver–Stehfest method becomes unusable due to its poor numerical stability. Increasing the order to $n = 72$, as shown in Figure 5 improves the CME method further, while for the Euler method it only increases the Gibbs oscillation. This example shows that the CME method is not only Gibbs oscillation free and numerically stable, but also reliable in the sense that increasing its order also increases its accuracy.

6. APPLICATION OF NILT IN PERFORMANCE ANALYSIS

To demonstrate the application of NILT in performance analysis and the use of the offline tool in Mathematica environment we provide the response time analysis of a retrial system with unreliable servers from [23].

The considered queueing system behaves as follows. Incoming customers queue up in an infinite buffer when the server is busy. A single server serves the customers in FIFO order, which is subject to breakdown. In case of a server breakdown, the server gets back to operational after an independent, identically distributed (i.i.d.) setup time. If the server was busy at breakdown, it continues the service

of the interrupted customer when it gets back to operational according to one of the following three preemption policies: preemptive resume (PRS), preemptive repeat different (PRD), and preemptive repeat identical (PRI). With the PRS policy, the server continues the service of the interrupted customer from the point it was interrupted. With the PRD policy, after an interruption the server restarts the service of the interrupted customer with i.i.d. service time. With the PRI policy the server restarts the service of the interrupted customer, and the service time of the customer in the current operational period of the server is identical to the one of the previous operational period.

The performance measure of interest is the generalized service time, G , which is the time from the instant the server starts the service of a customer until it completes the service of that customer considering the potential breakdown and setup cycles of the server and the applied preemption policy.

The CDF, PDF, and the LT of the (breakdown free) service time, S , are denoted by $F(x) = Pr(S < x)$, $f(x) = dF(x)/dx$, and $f^*(s) = E(e^{-sS})$, respectively. Similarly, the CDF, PDF, and the LT of the setup time, R , and the generalized service time G are denoted by $R(x)$, $r(x)$, $r^*(s)$ and $G(x)$, $g(x)$, $g^*(s)$, respectively. Additionally, we are interested in the equilibrium distribution of the generalized service time, whose PDF is $g_e(x) = \frac{1-G(x)}{E(G)}$ and LT is $g_e^*(s) = \frac{1-g^*(s)}{sE(G)}$. The server breaks down with constant rate ν .

With the different preemption policies the LT of the generalized service time is

$$g_{\text{prd}}^*(s) = \frac{(s + \nu)f^*(s + \nu)}{(s + \nu) - \nu(1 - f^*(s + \nu))r^*(s)}, \quad (7)$$

$$g_{\text{prs}}^*(s) = f^*(s + \nu - \nu r^*(s)), \quad (8)$$

$$g_{\text{pri}}^*(s) = \frac{(s + \nu)}{(s + \nu) - \nu r^*(s)} \sum_{j=0}^{\infty} \left(\frac{-\nu r^*(s)}{(s + \nu) - \nu r^*(s)} \right)^j f^*((j + 1)(s + \nu)). \quad (9)$$

Figure 6 plots the density function of the generalized service with different Weibull distributed service time distributions and exponentially distributed setup time (that is $r^*(s) = \frac{\rho}{s + \rho}$). The PDF of the Weibull distribution is $f(t) = \alpha\lambda(\lambda t)^{\alpha-1}e^{-(\lambda t)^\alpha}$, where the α shape parameter determines the decay rate as t tends to infinity. We consider

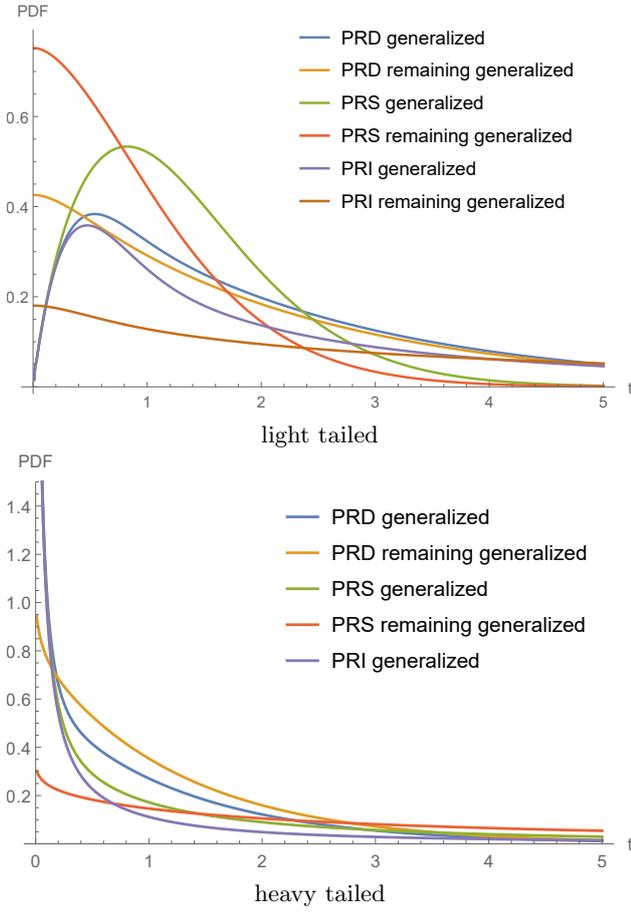


Figure 6: Density function of the generalized service time distribution and the remaining generalized service time distribution with PRD, PRS and PRI policies, when the service time is light ($\lambda = 1, \alpha = 2$) and heavy ($\lambda = 1, \alpha = 1/2$) tailed Weibull distributed, the setup time is exponentially distributed with parameter $\varrho = 4$ and the failure rate of the server is $\nu = 2$.

two cases: In the heavy tailed case with $\alpha = 1/2$

$$f^*(s) = \frac{\sqrt{\pi\lambda}}{2\sqrt{s}} e^{\frac{\lambda}{4s}} \operatorname{Erfc}\left(\frac{\sqrt{\lambda/s}}{2}\right),$$

and in the light tailed case with $\alpha = 2$

$$f^*(s) = 1 - \frac{s\sqrt{\pi}}{2\lambda} e^{\frac{s^2}{4\lambda^2}} \operatorname{Erfc}\left(\frac{s}{2\lambda}\right),$$

where Erfc is the complementary error function defined as $\operatorname{Erfc}(z) = \frac{2}{\sqrt{\pi}} \int_{t=z}^{\infty} e^{-t^2} dt$. Figure 6 plots the density functions of the different cases obtained from the NILT of (7)-(9). The plot of the heavy tailed case contains only 5 curves because the mean generalized service time is infinite with the PRI preemption policy in this case. Figure 7 presents an implementation example associated with the queueing model.

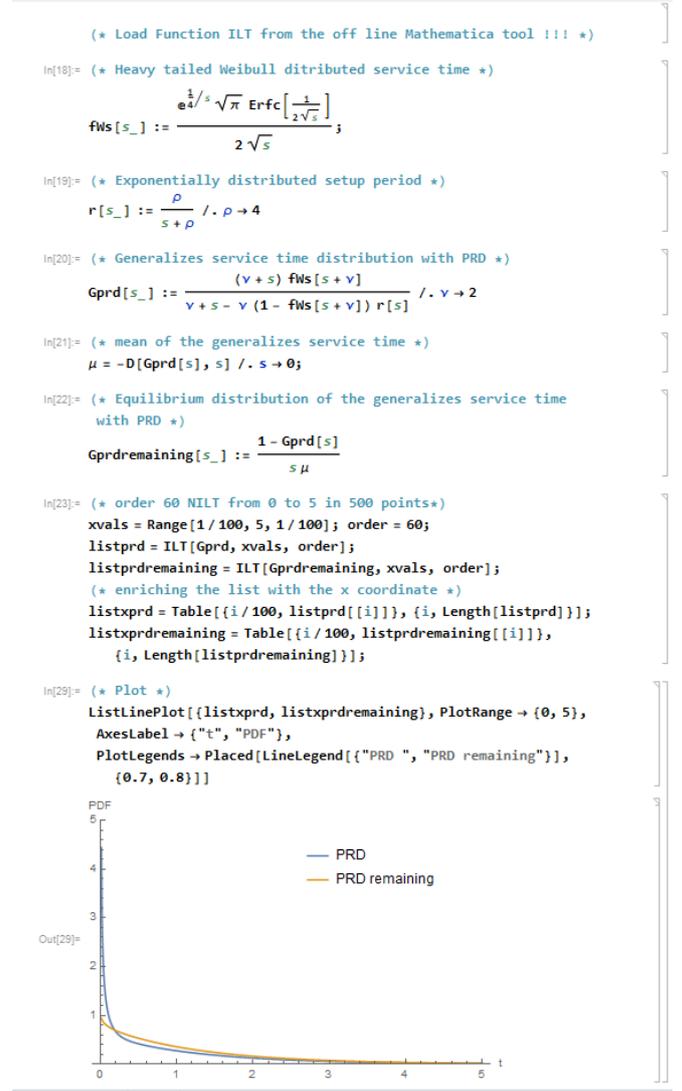


Figure 7: Mathematica code of the implementation.

7. CONCLUSION

In this paper we presented the CME method and demonstrated its advantages compared to other NILT methods. We showed that, unlike its most popular competitors, it provides results free of Gibbs oscillation, improves with increasing order, and is very stable numerically. These assets make it an ideal choice as a general method for NILT. We illustrated the above benefits using our online tool, which has a simple, intuitive interface and is optimal for testing and demonstrative purposes, while its offline version, implemented in multiple popular scientific languages, offers more options for fine-tuning and is recommended for general research purposes.

8. REFERENCES

- [1] J. Abate, G. L. Choudhury, and W. Whitt. An introduction to numerical transform inversion and its application to probability models. In *Computational Probability*, pages 257–323. Springer US, Boston, MA, 2000.
- [2] J. Abate and W. Whitt. A Unified Framework for Numerically Inverting Laplace Transforms. *INFORMS Journal on Computing*, 18(4):408–421, Fall 2006.
- [3] S. Ahn, A. Badescu, and V. Ramaswami. Time dependent analysis of finite buffer fluid flows and risk models with a dividend barrier. *Queueing Syst.*, 55:207–222, 05 2007.
- [4] S. A.-D. Almousa, G. Horváth, and M. Telek. Transient analysis of piecewise homogeneous markov fluid models. *Annals of Operations Research*, 2020.
- [5] S. A.-D. Almousa, G. Horváth, and M. Telek. Transient analysis of piecewise homogeneous QBD process. *Stochastic Models*, 37(1):59–84, 2020.
- [6] S. A.-D. Almousa, G. Horváth, and M. Telek. EM based parameter estimation for Markov modulated fluid arrival processes. In *26th International Conference on Analytical & Stochastic Modelling Techniques & Applications (ASMTA)*, Tsukuba, Japan – online, Dec 9-14 2021.
- [7] S. A.-D. Almousa and M. Telek. Enhanced optimization of high order concentrated matrix-exponential distributions. *Annales Mathematicae et Informaticae*, 2021.
- [8] B. Biedermann and T. Meurer. Motion planning for a class of boundary controlled 1D port-Hamiltonian systems. *IFAC-PapersOnLine*, 53:7710–7715, 2020.
- [9] L. Cui, J. Chen, and B. Wu. New interval availability indexes for markov repairable systems. *Reliability Engineering & System Safety*, 168:12–17, 2017.
- [10] E. Deiana, G. Latouche, and M.-A. Remiche. Sojourn time distribution in fluid queues. In *International Conference on Queueing Theory and Network Applications*, pages 295–313. Springer, 2019.
- [11] Y. Fang. Modeling and performance analysis for wireless mobile networks: a new analytical approach. *IEEE/ACM Transactions on Networking*, 13(5):989–1002, 2005.
- [12] M. FINKELSTEIN. On some models of acceptable risk. *International Journal of Performance Engineering*, 5(1):5, 2009.
- [13] D. P. Gaver. Observing stochastic processes and approximate transform inversion. *Oper. Res.*, 14:444–459, 1966.
- [14] M. Harchol-Balter. Open problems in queueing theory inspired by datacenter computing. *Queueing Systems*, 97(1):3–37, 2021.
- [15] M. O. Hasna and M.-S. Alouini. Outage probability of multihop transmission over nakagami fading channels. *IEEE Communications Letters*, 7:216–218, 2003.
- [16] G. Horváth, I. Horváth, and M. Telek. High order concentrated matrix-exponential distributions. *Stochastic Models*, 36(2):176–192, 2020.
- [17] I. Horváth, G. Horváth, S. A.-D. Almousa, and M. Telek. inverselaplace.org. <https://inverselaplace.org/>. [Online; accessed 24-Sept-2021].
- [18] I. Horváth, G. Horváth, S. A.-D. Almousa, and M. Telek. Numerical inverse Laplace transformation using concentrated matrix exponential distributions. *Performance Evaluation*, 137, 2020.
- [19] K. L. Kuhlman. Review of inverse Laplace transform algorithms for Laplace-space numerical approaches. *Numerical Algorithms*, 63(2):339–355, 2013.
- [20] Y.-C. Lin and H.-D. Yeh. An analytical model with a generalized nonlinear water transfer term for the flow in dual-porosity media induced by constant-rate pumping in a leaky fractured aquifer. *Water Resources Research*, 57(8):e2020WR029186, 2021.
- [21] D. A. Mahmood and G. Horváth. Analysis of the message propagation speed in VANET with disconnected RSUs. *Mathematics*, 8(5):782, 2020.
- [22] Y. Masuda and S. Yamakawa. A compound dependability measure arising from semi-markov reliability model. *Journal of the Operations Research Society of Japan*, 43(4):448–454, 2000.
- [23] A. Mészáros, E. Morozov, T. Morozova, and M. Telek. Numerical analysis of a retrial system with unreliable servers based on Laplace domain description. In *Distributed Computer and Communication Networks: Control, Computation, Communications (DCCN)*, Moscow, Russia, Sept 20-24 2021.
- [24] R. M. Neupauer, G. Lackey, and J. Pitlick. Exaggerated stream depletion in streams with spatiotemporally varying streambed conductance. *Journal of Hydrologic Engineering*, 26(2):04020066, 2021.
- [25] U. of Illinois. Ncsa open source license. <https://opensource.org/licenses/NCSA>. [Online; accessed 2002-03-28].
- [26] K. Reddy, K. Kumar, J. Satish, and S. Vaithyasubramanian. A review on applications of Laplace transformations in various fields. *Journal of Advanced Research in Dynamical and Control Systems*, 9:14–24, 01 2017.
- [27] J. Shen and L. Cui. Reliability performance for dynamic systems with cycles of k regimes. *IIE Transactions*, 48(4):389–402, 2016.
- [28] H. Stehfest. Algorithm 368: Numerical Inversion of Laplace Transforms [D5]. *Commun. ACM*, 13(1):47–49, Jan. 1970.
- [29] A. Talbot. The Accurate Numerical Inversion of Laplace Transforms. *IMA Journal of Applied Mathematics*, 23(1):97–120, 01 1979.
- [30] P. Valko and J. Abate. Comparison of sequence accelerators for the Gaver method of numerical Laplace transform inversion. *Comput. Math. Appl.*, 48:629–636, March 2004.
- [31] Q. Wang and H. Zhan. On different numerical inverse Laplace methods for solute transport problems. *Advances in Water Resources*, 75:80 – 92, 2015.