Moment Generating Function to Probability Density Function Transform Methods

1st Darya Gaponenko
Peter the Great St.Petersburg Polytechnic University
St.Petersburg, Russia
daria.gaponenko@yandex.ru

2nd Aleksandr Sidnev
Peter the Great St.Petersburg Polytechnic University
St.Petersburg, Russia
alex2922@gmail.com

***Abstract*—The moment generating function completely characterizes the random variable distribution. At the same time, the natural desire of the researcher is to obtain the density and the distribution function as more informative characteristics. The article analyzes popular transition methods from the moment generating function to the probability density, in particular, the saddlepoint method, the method based on the Heaviside theorem and Padé approximation, a significant number of the inverse Laplace transform numerical methods. Computational complexity and practical feasibility of the methods are investigated for a number of practical problems of the inverse Laplace transform. Recommendations on the optimal method choice are given and the directions for the further development of the topic are suggested as well.**

***Index Terms*—probability density function, moment generating function, numerical Laplace inversion methods, Padé approximation**

I. Introduction

Moment generating function (MGF) is an important statistical characteristic that uniquely describes the random variable distribution. Since the probability density function (PDF) is related to the MGF through the inverse Laplace transform [1]:

$$f(t) = L^{-1}\Big[M(-s)\Big],$$

the computational complexity can be significantly reduced by using the MGF instead of PDF. For example, MGF of the two random variables sum is calculated as the product of the MGF of these variables [1].

Flowgraph models operate MGF to obtain the stochastic process total time distribution. They can be applied in different areas. For example, in reliability theory for the uptime estimation, in medical statistics for the disease progression and the treatment effectiveness analysis [2], [3], for the power system survivability analysis and it's weak points identification [4].

Padé approximation, Heaviside's theorem, Saddlepoint approximation and numerical inverse Laplace transform methods are used to obtain the PDF as the MGF inversion (while the MGF is acquired as the result of the flowgraph specific analysis). The transform method must provide an acceptable approximation error with a relatively low computational complexity and must generate non-negative PDF values.

The paper is organised as follows. Section II introduces the MGF transform via Padé approximation, section III presents the Saddlepoint approximation, section IV provides the review of eight numerical inverse Laplace transform methods. In section V we select methods for the software implementation and comparison on the basis of the preliminary analysis. Finally section VI is devoted to the results discussing.

II. Padé approximation

The Heaviside theorem is suggested to obtain the PDF applying the inverse Laplace transform of the MGF [5], [6]:

$$L^{-1}\Big[\frac{U(s)}{R(s)}\Big] = \sum_{k=1}^{q} \frac{U(\alpha_k)}{R'(\alpha_k)} * e^{\alpha_k * t},$$

where α_k are the roots of the polynomial R(s).

MGF Padé approximation can be straightforward provided as series:

$$\sum_{i=0}^{\infty} c_i * x^i = \frac{\sum_{j=0}^{p} a_j * x^j}{\sum_{k=0}^{q} b_k * x^k}$$

Padé approximation is applicable to MGF because:

$$M(s) = \sum_{n=0}^{\infty} \frac{\mu_n}{n!} * s^n,$$

where μ_n is n-th moment.

Ren [5] compares three methods: Padé approximation, maximum entropy method and saddlepoint approximation. It is noted that the Padé approximation is more informative and always provides an analytical approximation for the unknown original PDF.

But this approach has a significant disadvantage: Padé approximation requires the high order derivatives computation (because the n-th moment is the n-th derivative of the MGF). This fact leads to great computational complexity often exceeding the hardware possibilities for the high order derivatives calculation in the symbolic form.

III. Saddlepoint approximation

Ren [5] also considers the technique of saddlepoint approximation for the MGF-PDF transform:

$$f(t) = \Big(\frac{n}{2 * \pi * K''(s)}\Big)^{\frac{1}{2}} * \exp(n * (K(s) - st)),$$

where $K(s) = log(M(s))$, $K'(s) = t$, n is the number of random variables. For one random variable PDF approximation is equal to [7]:

$$f(t) = \Big(2 * \pi * K''(s)\Big)^{-\frac{1}{2}} * \exp(K(s) - st)$$

The error of this approximation is estimated as $O(n^{-\frac{1}{2}})$ [5]. Collins [8] notes this method to have less efficiency than the inverse Laplace transform.

IV. Inverse Laplace transform methods

There are over a hundred different numerical inverse Laplace transform (ILT) methods. Being applied to the various types of functions they differ in computational complexity and approximation accuracy.

Some ILT methods have Gibbs oscillations – an oscillatory error which occurs when a discontinuous function is approximated with Fourier series. Gibbs oscillations do not disappear when more terms are added to the approximation (the ”size” of the excess does not decrease vertically, but only horizontally [9]).

A. Abate–Whitt framework

Different ILT numerical methods can be combined within the same mathematical framework. Abate and Whitt propose the unified formula (1) and notice, that such algorithms as Gaver–Stehfest, Talbot and Fourier series method with Euler summation can fit into this framework [10].

Abate–Whitt framework [11]:

$$f(t) \approx \sum_{k=1}^{N} \frac{\eta_k}{t} F(\frac{\beta_k}{t}), \tag{1}$$

where $t > 0$, nodes β_k and weights η_k (internal and external scaling constants) are real or complex numbers that depend on N, but do not depend on the transform F or the time argument t [12].

Abate–Whitt framework has low computational complexity, but it is characterized by a deterioration of the approximation on periodic functions at large values of t [9]. Since PDF is not a periodic function, this disadvantage can be ignored.

Various ILT methods in this framework differ only by approaches to η_k and β_k calculation.

Several methods belonging to the Abate-Whitt framework are considered below.

1) CME: Method is based on the concentrated matrix-exponential distribution [11].

Matrix-exponential distributions of order N contain positive random variables with PDF $f(t) = -\alpha A e^{At} 1$, at $t \geq 0$, where α is a real row vector of length N, A is an NxN real matrix, and 1 is a column vector of ones of size N [11].

Let A be diagonalizable with spectral decomposition $A = \sum_{k=1}^{N} u_k \lambda_k \nu_k$, where λ_k are the eigenvalues, u_k are the right eigenvectors and ν_k are the left eigenvectors of A with $\nu_k u_k = 1$, then the PDF of the matrix-exponential distribution can be written as:

$$f(t) = \sum_{k=1}^{N} c_k e^{\lambda_k t},$$

where $c_k = -\alpha A u_k v_k 1$.

A matrix-exponential distribution is called concentrated if its squared coefficient of variation (SCV) is minimal [13]. The density of the matrix-exponential distribution with the minimal SCV is known analytically only for the order $N < 3$. Therefore, the matrix-exponential distributions required for the approximation were calculated using numerical optimization by the expression:

$$f(t) = ce^{-\lambda t} \prod_{j=0}^{\frac{N-1}{2}} cos^2(\omega t - \phi_j) = \sum_{k=1}^{N} \eta_k e^{-\beta_k t},$$

where $c, \lambda, \omega, \phi_j$ are positive real values.

Thus, the values η_k and β_k providing the minimum SCV for the matrix-exponential distribution are the coefficients of the Abate-Whitt framework and are obtained from numerical optimization [14].

It is guaranteed that the CME method is free from Gibbs oscillations and preserves monotonicity [11]. Many of the known methods do not have these properties: for example, the $f_n(t)$ function for the Euler and Gaver–Stehfest methods can take negative values.

Advantages of the method: it does not cause overshoot, maintains monotonicity, is accurate when using floating point arithmetic, the quality of the approximation improves with increasing order [11].

2) Gaver–Stehfest method: Method is based on the three-parameter exponential PDF properties (Gaver method [15]). Stehfest [16] improves method by taking the weighted average of a Gaver approximations sequence for fixed t.

Gaver–Stehfest algorithm does not use complex numbers; weights and nodes are real numbers and are calculated as follows [10]:

$$\beta_k = k * \ln(2)$$

$$\eta_k = (-1)^{(\frac{N}{2}+k)} \ln(2) *$$

$$* \sum_{j=\lfloor \frac{k+1}{2} \rfloor}^{min(k,\frac{N}{2})} \frac{j^{\frac{N}{2}} 2j!}{(\frac{N}{2}-j)! j! (j-1)! (k-j)! (2j-k)!},$$

where N is the number of terms used in the equation (the method is defined only for even N). Wang in [10] notices that N is recommended to be taken in the range from 10 to 14.

The main disadvantage of the method is Gibbs oscillations [11].

Abate and Whitt [12] show that Euler and Talbot methods are more efficient than show that Euler and Talbot methods are more efficient than those of Gaver–Stehfest. It is also noted that the algorithm implementation requires high computational accuracy due to manipulations with large numbers [12], [17]. However, the Gaver–Stehfest algorithm has the advantage of using only real numbers.

3) Euler method: is an implementation of the Fourier series method using Euler summation to accelerate convergence [18]. Method is applicable only to odd orders and assumes nodes with positive imaginary parts.

Nodes and weights are calculated as follows [11]:

$$\beta_k = \frac{(N-1)\ln(10)}{6} + \pi i(k-1)$$

$$\eta_k = 10^{\frac{N-1}{6}}(-1)^k \xi_k,$$

where $\xi_1 = \frac{1}{2}$;
$\xi_k = 1,\ 2 < k < \frac{n+1}{2}$;
$\xi_n = \frac{1}{2^{\frac{n-1}{2}}}$;
$\xi_{n-k} = \xi_{n-k+1} + 2^{-\frac{n-1}{2}}\binom{\frac{n-1}{2}}{k},\ 1 < k < \frac{n-1}{2}$

Method suffers from Gibbs oscillations [11].

4) Talbot method: Method is based on deforming the contour integral in the Bromwich inversion [19]. It assumes nodes with positive imaginary parts and also applies values of β_k with a negative real part, which is usually outside the region of convergence, but there can be an analytic continuation of $F(s)$ [9].

Nodes and weights are calculated as follows [11]:

$$\beta_1 = \frac{2N}{5}$$

$$\beta_k = \frac{2(k-1)\pi}{5}\Big(\cot\big(\frac{(k-1)\pi}{N}\big) + i\Big)$$

$$\eta_1 = \frac{1}{5}e^{\beta_1}$$

$$\eta_k = \frac{2}{5}\Big[1 + i\frac{(k-1)\pi}{N}\Big(1 + \big[\cot\big(\frac{(k-1)\pi}{N}\big)\big]^2\Big) -$$

$$-i\cot\big(\frac{(k-1)\pi}{N}\big)\Big]e^{\beta_k}$$

Method is not free from Gibbs oscillations [11].

5) Zakian method: Method is based on Fourier series method with Padé approximation [11].

General formula is [10]:

$$f(t) = \frac{1}{t}\sum_{k=1}^{N} Re\{K_k F(\frac{\alpha_k}{t})\}, \tag{2}$$

where K_k and α_k are real or complex constants. N represents the number of terms used in the summation. According to [10], [20] a sufficient value of N in most cases is $N = 10$. Coefficients for $1 \leq N \leq 10$ are calculated in [20].

Zakian proposes two methods for choosing coefficients K_k and α_k. In the first he compares the Laplace transform $\delta(t,u)$ (a rational function) with the Laplace transform of $\delta_n(u-t)$ (an exponential function) and chooses the coefficients so that the rational functions are equal to the classical Padé approximations of the exponential function [21]. Another method includes least squares optimization [22].

6) Hyperbolic kernel approximation method: Method is based on the Laplace transform inverse kernel approximation by expressions containing hyperbolic functions $\sinh$ and $\cosh$ or their combinations [23].

The general formula [24] is based on the combining two approaches to the Laplace transform inverse kernel approximation $e^{st} \approx \frac{e^a}{2sinh(a-st)}$ and $e^{st} \approx \frac{e^a}{cosh(a-st)}$ to increase accuracy:

$$f(t) = \frac{e^a}{2t}\Big(\frac{1}{2}F(\frac{a}{t}) + \sum_{n=1}^{N}(-1)^n\big(Re\big\{F\big(\frac{a}{t} + in\frac{\pi}{t}\big)\big\} +$$

$$+Im\big\{F\big(\frac{a}{t} + i(n-\frac{1}{2})\frac{\pi}{t}\big)\big\}\big)\Big) \tag{3}$$

The absolute error depends on the a as follows [24]:

$$\varepsilon_{aM} \approx Me^{-4a},$$

where M is the maximum absolute value of the original $f(t)$.

The parameter a is had to be found empirically. It is recommended to have a in the following range $2 \leq a \leq 6$ [23], [25]. Brančík and Smith [26] also notice that one of the ways to improve the accuracy of this approximation is to increase the parameter a.

B. Post–Widder method

ILT is calculated by the formula [27]:

$$f(t) := \frac{(-1)^{N-1}}{(N-1)!}\Big(\frac{N}{t}\Big)^N F^{(N-1)}(\frac{N}{t}), \tag{4}$$

where $F^{(n-1)}(x)$ is the n-th derivative of $F(x)$.

The main issue is that method requires high order derivatives computation and is characterized by slow convergence [27].

Post-Widder method does not cause the overshoot and maintains monotonicity [11]. However, according to Horvath's research [11], the CME method gives better approximation. Davies [28] also notices that the method rarely gives a high accuracy of the approximation.

C. Laguerre method

Method is based on Laguerre polynomials. ILT is calculated by the formula [29]:

$$f(t) \approx \sum_{n=0}^{\infty} q_n l_n(t), \tag{5}$$

where l_n is Laguerre function, q_n are Laguerre coefficients, dependent on F.

Laguerre function:

$$l_n(x) = e^{-\frac{x}{2}}L_n(x), \tag{6}$$

where L_n is Laguerre polynomial.

Laguerre polynomial:

$$L_n(x) = \sum_{m=0}^{n}\binom{n}{m}\frac{(-x)^m}{m!} \tag{7}$$

Laguerre coefficients generating function:

$$Q(z) = \sum_{n=0}^{\infty} q_n z^n = \frac{1}{1-z} F\Big(\frac{1+z}{2(1-z)}\Big) \tag{8}$$

Formulas 5 – 8 can be rewritten as follows [11]:

$$f(t) = \sum_{j=0}^{N-1} F^{(j)}(\frac{1}{2}) \sum_{k=j}^{N-1} \frac{k!}{(j!)^2(k-j)!} l_k(t), \tag{9}$$

where $F^{(j)}(s)$ is the j-th derivative of $F(s)$.

Davies [28] notices that Laguerre polynomials give an accurate approximation over a wide range of functions.

The disadvantage of this method is that computation requires high order differentiation. Laguerre method also suffers from Gibbs oscillations [11].

D. Other methods

Cohen method is based on the power series $F(s)$ as a function $\frac{1}{s}$. However, this method is not applicable when $F(s)$ has a pole at 0, and the approximation also gets rapidly worse for larger values of t [11].

Honig-Hirdes and de Hoog methods are the variation of the Fourier series method. Wellekens uses the Fourier series method based on the Padé approximation [11]. Schapery method is described in [30], but according to [28], it rarely gives the high accuracy of the approximation.

V. Choosing methods for the implementation

We will determine the most promising solutions for the considered methods.

Padé approximation, Post-Widder and Laguerre methods provide high computational complexity, because of calculating high order derivatives.

Gaver–Stehfest, Euler, Talbot methods suffer from the Gibbs oscillations, and since PDF is restricted to have negative values, these methods are also beyond our consideration.

So we choose four of the most promising methods for the comparison: the saddlepoint approximation, CME method, Zakian method and hyperbolic kernel approximation method.

VI. Implementation and comparison of the selected methods

The selected methods are implemented as Matlab functions (Matlab R2020a version was used), because Matlab [31] has one of the most powerful symbolic toolboxes. For the comparison with other methods the CME method is taken in the format suggested in [32] with some parameters pre-calculated by the authors of the method. The saddlepoint method, Zakian method, hyperbolic kernel approximation method are Matlab implemented according to [33]. Zakian method is realised with pre-calculated parameters K_i and α_i borrowed from [20]. The a parameter value is selected empirically for the hyperbolic kernel approximation method. Since Matlab has the built-in inverse Laplace transform function, it is reasonable to compare it with the selected methods.

The method's average running time is stated as the performance criteria. The hardware in use is Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz, RAM 8,00 GB with Windows 10.

The accuracy of the methods tested is assessed by two parameters: maximum absolute deviation ε_{abs}, to check for the sharp exceeds in the approximation, and 1-norm of the numerical error ε_n calculated by the formula:

$$\varepsilon_n = \int_0^T |f(t) - f_{appr}(t)| dt \approx \frac{1}{M} \sum_{m=1}^{M} |f(m) - f_{appr}(m)|$$

Original function $f(t)$ is compared with approximated PDF $f_{appr}(t)$ at $M = 200$ equidistant points.

The problem of the PDF mining from the flowgraph is stated for the beforehand known PDFs just for the testing purpose. The methods are checked on the approximation of exponential, normal, triangular and uniform distributions with well known Laplace transforms. The flowgraph with 5 nodes shown on fig. 1 is also investigated on the subject of the transition time distribution from the node 1 to the node 5. The flowgraph parameters are presented in the Table I. It should be noted that for this graph the 10-th MGF derivative calculation takes 2 seconds, the 12-th derivative calculation takes 4.5 seconds. It confirms that methods based on the high order derivatives calculation are not suitable for the implementation.

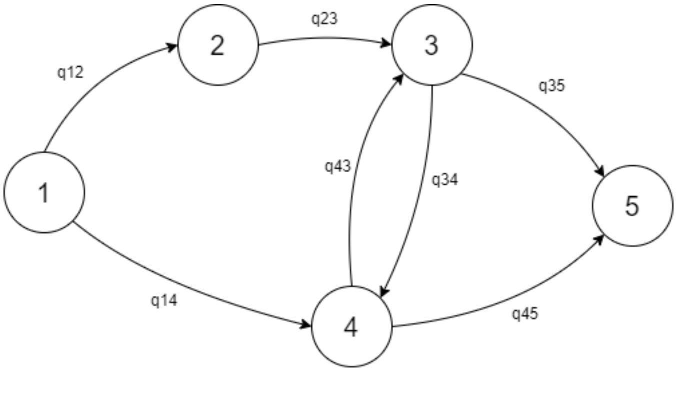

Fig. 1. Flowgraph

TABLE I
Flowgraph parameters

Edge	Probability	Distribution	Parameters
q12	0.2	Exponential	$\lambda = 0.2$
q14	0.8	Uniform	$a = 2.4, b = 9.6$
q23	1	Triangular	$a = 1.6, b = 6.4, c = 4$
q34	0.9	Exponential	$\lambda = 0.5$
q35	0.1	Uniform	$a = 6, b = 24$
q43	0.5	Triangular	$a = 4, b = 16, c = 10$
q45	0.5	Exponential	$\lambda = 0.3$

A. Exponential distribution

Built-in Matlab function demonstrates the best performance and the highest approximation accuracy (tables II - III) for the exponential distribution ($\lambda = 0.5$) among all the methods in question. A visual comparison (Fig. 2) confirms these results.

TABLE II
EXPONENTIAL DISTRIBUTION APPROXIMATION TIME

Matlab function, s	Zakian, s	Saddlepoint, s
0.0110 ± 0.0005	0.055 ± 0.005	0.111 ± 0.002

Order	CME, s	Hyperbolic, s
5	0.319 ± 0.009	0.098 ± 0.004
10	0.356 ± 0.009	0.182 ± 0.012
20	0.406 ± 0.007	0.353 ± 0.009
50	0.5915 ± 0.0104	0.860 ± 0.012
100	0.888 ± 0.019	1.714 ± 0.021

TABLE III
EXPONENTIAL DISTRIBUTION APPROXIMATION ERRORS

Matlab function		Zakian		Saddlepoint	
ε_n	ε_{abs}	ε_n	ε_{abs}	ε_n	ε_{abs}
0	0	2.51E-03	6.81E-03	3.25E-04	2.56E-02

Order	CME		Hyperbolic	
	ε_n	ε_{abs}	ε_n	ε_{abs}
5	3.40E-04	3.70E-03	2.68E-03	1.79E-02
10	5.62E-05	8.02E-04	1.62E-03	6.11E-03
20	1.03E-05	1.74E-04	1.02E-03	3.15E-03
50	9.60E-07	2.30E-05	8.29E-04	4.36E-03
100	3.60E-07	9.00E-06	5.54E-04	5.00E-03

B. Normal distribution

The Matlab built-in function is unable to provide the inverse Laplace transform of the normal (Gauss) distribution with the expected value $\mu = 30$ and standard deviation $\sigma = 3$. It returns an unevaluated call to ilaplace function. Saddlepoint approximation provides the set of values that exactly approximates the Gauss distribution only for the bounded segment [0;116]. Zakian and CME methods show sharp spikes in the values of the approximated function. Only the hyperbolic method gives a sufficiently accurate approximation (Table V). A visual comparison shows that CME method is characterized by the sharp exceed of values in small t, but Zakian method cannot provide an accurate approximation for $t > 30$ (Fig. 3). Zakian method ensures maximum performance, but because of low accuracy, we recommend to use hyperbolic approximation for this distribution.

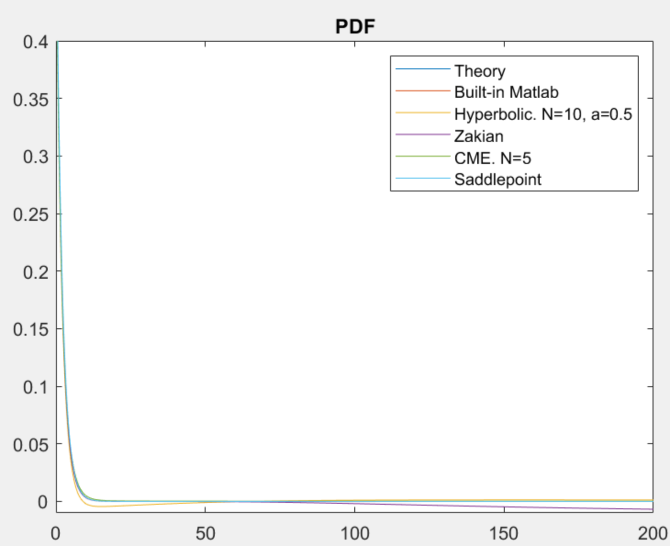

Fig. 2. PDF approximation. Exponential distribution

TABLE IV
NORMAL DISTRIBUTION APPROXIMATION TIME

Matlab function, s	Zakian, s	Saddlepoint, s
-	0.072 ± 0.004	0.112 ± 0.004

Order	CME, s	Hyperbolic, s
5	0.319 ± 0.005	0.096 ± 0.003
10	0.376 ± 0.014	0.219 ± 0.009
20	0.44 ± 0.02	0.420 ± 0.018
50	2.92 ± 0.03	2.955 ± 0.009
100	4.55 ± 0.06	5.89 ± 0.02

TABLE V
NORMAL DISTRIBUTION APPROXIMATION ERRORS

Matlab function		Zakian		Saddlepoint, $t \in [0; 116]$	
ε_n	ε_{abs}	ε_n	ε_{abs}	ε_n	ε_{abs}
-	-	p. inf.	p. inf.	0	0

Order	CME		Hyperbolic	
	ε_n	ε_{abs}	ε_n	ε_{abs}
5	2.94E-03	6.03E-02	5.03E-03	1.95E-02
10	1.75E+00	3.49E+02	2.82E-03	6.01E-03
20	p. inf.	p. inf.	8.91E-04	2.92E-03
50	p. inf.	p. inf.	4.66E-05	2.44E-03
100	p. inf.	p. inf.	1.85E-05	2.44E-03

*p. inf. (practically infinite) stands for values larger than 1000

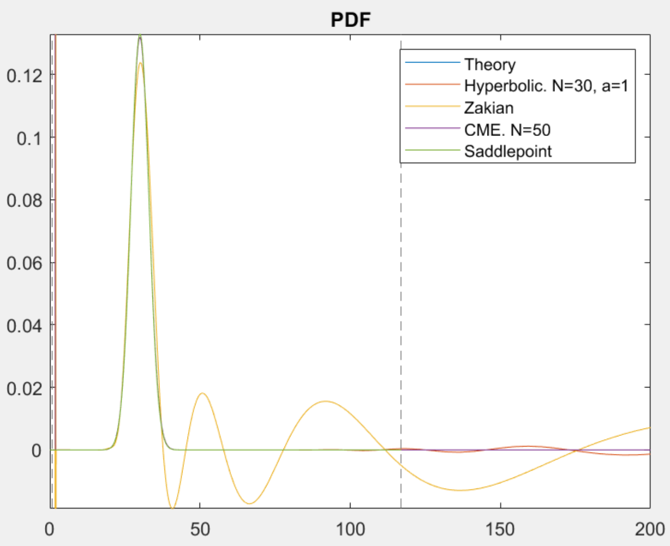

Fig. 3. PDF approximation. Normal distribution

C. Triangular distribution

The saddlepoint approximation for the triangular distribution with parameters $a = 10$; $b = 50$; $c = 30$ is unable to find the answer: method fails with error ”Unable to find explicit solution”. The highest approximation accuracy and performance is demonstrated by the built-in Matlab function (Tables VI, VII). Zakian method does not provide the required approximation accuracy (Fig. 4).

TABLE VI
TRIANGULAR DISTRIBUTION APPROXIMATION TIME

Matlab function, s	**Zakian, s**	**Saddlepoint, s**
0.0108 ± 0.0005	0.16 ± 0.03	-
Order	**CME, s**	**Hyperbolic, s**
5	0.46 ± 0.08	0.144 ± 0.005
10	0.68 ± 0.02	2.56 ± 0.03
20	1.068 ± 0.018	5.03 ± 0.04
50	2.34 ± 0.08	13 ± 1
100	3.39 ± 0.04	25.56 ± 0.17

TABLE VII
TRIANGULAR DISTRIBUTION APPROXIMATION ERRORS

Matlab function		**Zakian**		**Saddlepoint**	
ε_n	ε_{abs}	ε_n	ε_{abs}	ε_n	ε_{abs}
2.50E-04	5.00E-02	1.17E-03	5.28E-02	-	-
		CME		**Hyperbolic**	
Order		ε_n	ε_{abs}	ε_n	ε_{abs}
5		9.24E-04	5.92E-02	3.86E-03	5.23E-02
10		4.08E-04	5.43E-02	1.25E-03	5.15E-02
20		2.88E-04	5.20E-02	4.13E-04	5.07E-02
50		2.58E-04	5.07E-02	2.84E-04	5.03E-02
100		2.55E-04	5.05E-02	2.59E-04	5.02E-02

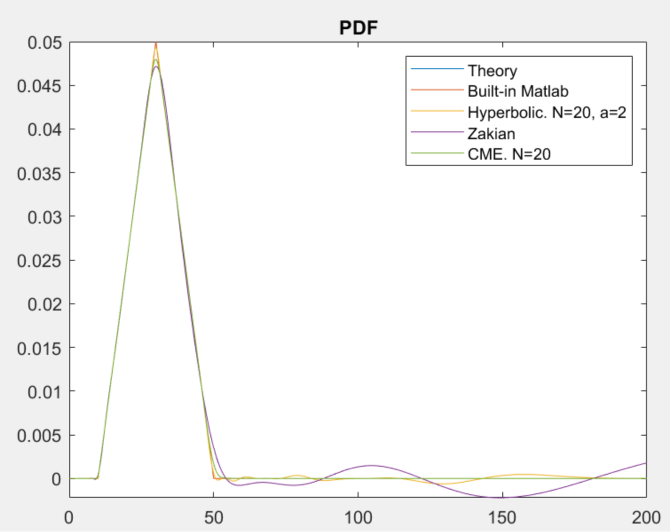

Fig. 4. PDF approximation. Triangular distribution

D. Uniform distribution

The saddlepoint approximation for the uniform distribution with parameterss $a = 20$; $b = 40$ fails with the error ”Unable to find explicit solution”. The error for the built-in Matlab function and CME method is explained by the PDF approximation corners rounding (Fig. 5). The CME method provides the best accuracy for the order of $N = 50$ (Table IX). The built-in Matlab function provides the highest performance with the approximation error of the order 10^{-2}, that makes it the best option among the considered methods. The hyperbolic approximation method suffers from Gibbs oscillations. The Zakian method does not provide the required approximation accuracy.

TABLE VIII
UNIFORM DISTRIBUTION APPROXIMATION TIME

Matlab function, s	**Zakian, s**	**Saddlepoint, s**
0.0114 ± 0.0016	0.18 ± 0.08	-
Order	**CME, s**	**Hyperbolic, s**
5	0.331 ± 0.004	0.21 ± 0.08
10	0.6 ± 0.10	0.42 ± 0.16
20	0.83 ± 0.03	3.55 ± 0.03
50	1.70 ± 0.07	10.2 ± 0.2
100	2.98 ± 0.11	21.6 ± 0.8

TABLE IX
UNIFORM DISTRIBUTION APPROXIMATION ERRORS

Matlab function		**Zakian**		**Saddlepoint**	
ε_n	ε_{abs}	ε_n	ε_{abs}	ε_n	ε_{abs}
2.50E-04	2.50E-02	3.14E-03	2.72E-02	-	-
		CME		**Hyperbolic**	
Order		ε_n	ε_{abs}	ε_n	ε_{abs}
5		1.93E-03	2.56E-02	5.59E-03	2.68E-02
10		8.93E-04	2.55E-02	2.50E-03	2.60E-02
20		4.45E-04	2.52E-02	1.55E-03	2.58E-02
50		2.61E-04	2.51E-02	8.91E-04	2.52E-02
100		2.52E-04	2.51E-02	6.91E-04	2.52E-02

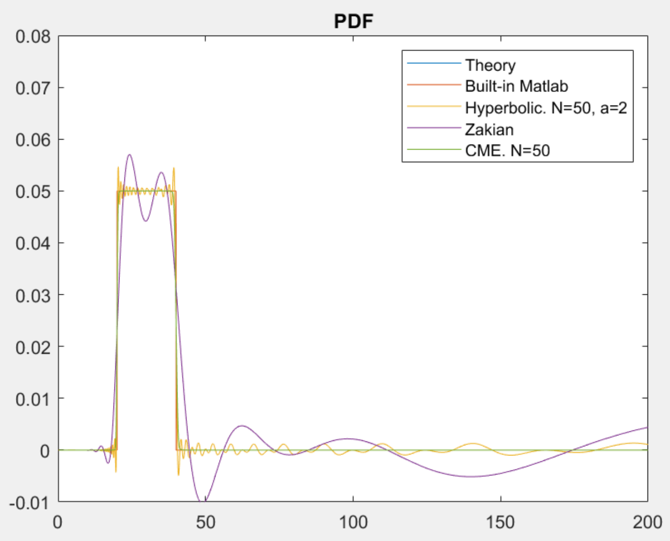

Fig. 5. PDF approximation. Uniform distribution

E. Flowgraph

For the flowgraph with different transition time distributions (Fig. 1, Table I) the saddlepoint approximation method fails with the error ”Unable to find explicit solution” and the same result is demonstrated by the built-in Matlab function that returns an unevaluated call to ilaplace function. The Zakian method does not provide the required approximation accuracy (Fig. 6). The best approximation is provided by the CME method (Table XI). For this case the CME method is faster than the hyperbolic approximation and the Zakian method as well (Table X).

TABLE X
FLOWGRAPH APPROXIMATION TIME

Matlab function, s	Zakian, s	Saddlepoint, s
-	$5.09 \pm 0,14$	-

Order	CME, s	Hyperbolic, s
5	2.48 ± 0.05	10.20 ± 0.12
10	4.83 ± 0.11	20.25 ± 0.19
20	8.77 ± 0.12	32.40 ± 0.18
50	21.8 ± 0.6	81.50 ± 0.82
100	34.5 ± 1.6	182 ± 8

TABLE XI
FLOWGRAPH APPROXIMATION ERRORS

Matlab function		Zakian		Saddlepoint	
ε_n	ε_{abs}	ε_n	ε_{abs}	ε_n	ε_{abs}
-	-	2.26E-03	9.85E-03	-	-

Order	CME		Hyperbolic	
	ε_n	ε_{abs}	ε_n	ε_{abs}
5	5.52E-04	7.08E-03	3.87E-03	9.15E-03
10	5.34E-04	7.95E-03	2.26E-03	1.06E-02
20	5.67E-04	9.83E-03	1.04E-03	1.08E-02
50	5.80E-04	1.05E-02	6.21E-04	1.04E-02
100	5.81E-04	1.04E-02	5.90E-04	1.05E-02

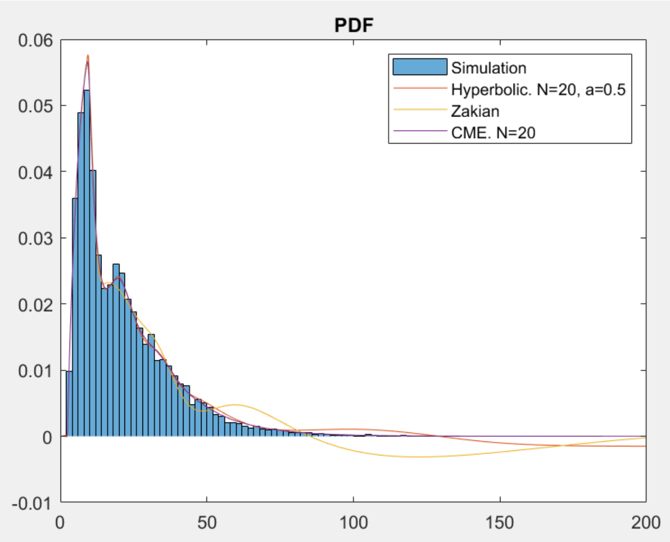

Fig. 6. PDF approximation of the transition time between the first and the last nodes of the graph

VII. CONCLUSION

In this paper we consider some well-known MGF inversion methods. Next, we have identified a group of methods that, according to the sources, are characterized by the least computational complexity and the absence of Gibbs oscillations. This group, which includes saddlepoint approximation, CME method, Zakian method and hyperbolic kernel approximation method, was tested on inverse Laplace transform problems for a number of distributions (exponential, normal, triangular, uniform). We also tested these methods on the meaningful problem represented by the five-state flowgraph. All methods are compared with the built-in Matlab function according to the criteria of accuracy and performance. The following conclusions were made on the test results:

- the Zakian method with the recommended in [10], [20] order $N = 10$ failed in most cases. So the calculation of parameters for higher orders is required;
- the hyperbolic kernel approximation method is not free from Gibbs oscillations and also requires the empirical selection of the a parameter, which makes it difficult to use method in the real cases when true distribution is unknown;
- the saddlepoint method gives an exact approximation for exponential and normal distributions, but it unable to provide the acceptable approximation of the PDF for uniform and triangular distributions;
- the built-in Matlab function provides the most accurate and fast result for exponential, triangular and uniform distributions, but does not cope with the MGF transform for the normal distribution and the flowgraph (fig. 1);
- thus, we can recommend the built-in Matlab function or the saddlepoint approximation only for a number of distributions;
- the CME method is the most stable among the considered methods. It is not subject to Gibbs oscillations, uses pre-calculated parameter values, so it has a relatively low computational complexity. In fact the definition of PDF here is reduced to summation and depends only on the complexity of the MGF expression. It is advisable to test the method on flowgraphs with large number of probabilistic transitions.

REFERENCES

[1] M. G. Bulmer, *Principles of statistics*. Courier Corporation, 1979.

[2] B. García-Mora, C. Santamaría, G. Rubio, and J. L. Pontones, “Bayesian prediction for flowgraph models with covariates. an application to bladder carcinoma,” *Journal of Computational and Applied Mathematics*, vol. 291, pp. 85–93, 2016.

[3] B. García-Mora, C. Santamaría, and G. Rubio, “Markovian modeling for dependent interrecurrence times in bladder cancer,” *Mathematical Methods in the Applied Sciences*, vol. 43, no. 14, pp. 8302–8310, 2020.

[4] A. Fetanat, “A decision support tool for modeling time-to-event data in power systems.” *Journal of Theoretical & Applied Information Technology*, vol. 60, no. 2, 2014.

[5] Y. Ren, “The methodology of flowgraph models,” Ph.D. dissertation, The London School of Economics and Political Science (LSE), 2011.

[6] A. Sidnev, “Business process analytical modeling — business analysis body of knowledge required component,” in *The European Proceedings of Social Behavioural Sciences. Edited by: Prof. Valeria Chernyavskaya and Prof. Holger Kuße*. Future Academy, 2018, pp. 1337–1348.

[7] A. V. Huzurbazar, “Flowgraph models: a bayesian case study in construction engineering,” *Journal of statistical planning and inference*, vol. 129, no. 1-2, pp. 181–193, 2005.

[8] D. H. Collins, R. L. Warr, and A. V. Huzurbazar, “An introduction to statistical flowgraph models for engineering systems,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 227, no. 5, pp. 461–470, 2013.

[9] I. Horváth, Z. Talyigás, and M. Telek, “An optimal inverse laplace transform method without positive and negative overshoot–an integral based interpretation,” *Electronic Notes in Theoretical Computer Science*, vol. 337, pp. 87–104, 2018.

[10] Q. Wang and H. Zhan, “On different numerical inverse laplace methods for solute transport problems,” *Advances in Water Resources*, vol. 75, pp. 80–92, 2015.

[11] G. Horváth, I. Horváth, S. A.-D. Almousa, and M. Telek, “Numerical inverse laplace transformation using concentrated matrix exponential distributions,” *Performance Evaluation*, vol. 137, p. 102067, 2020.

[12] J. Abate and W. Whitt, “A unified framework for numerically inverting laplace transforms,” *INFORMS Journal on Computing*, vol. 18, no. 4, pp. 408–421, 2006.

[13] G. Horváth, I. Horváth, S. A.-D. Almousa, and M. Telek, “High order low variance matrix-exponential distributions,” in *The Tenth International Conference on Matrix-Analytic Methods in Stochastic Models (MAM10)*, vol. 2, 2019.

[14] I. Horváth, O. Sáfár, M. Telek, and B. Zámbó, “Concentrated matrix exponential distributions,” in *European Workshop on Performance Engineering*. Springer, 2016, pp. 18–31.

[15] D. P. Gaver Jr, “Observing stochastic processes, and approximate transform inversion,” *Operations Research*, vol. 14, no. 3, pp. 444–459, 1966.

[16] H. Stehfest, “Algorithm 368: Numerical inversion of laplace transforms [d5],” *Communications of the ACM*, vol. 13, no. 1, pp. 47–49, 1970.

[17] R. Jacquot, J. Steadman, and C. Rhodine, “The gaver-stehfest algorithm for approximate inversion of laplace transforms,” *IEEE Circuits & Systems Magazine*, vol. 5, no. 1, pp. 4–8, 1983.

[18] J. Abate, G. L. Choudhury, and W. Whitt, “An introduction to numerical transform inversion and its application to probability models,” in *Computational probability*. Springer, 2000, pp. 257–323.

[19] A. Talbot, “The accurate numerical inversion of laplace transforms,” *IMA Journal of Applied Mathematics*, vol. 23, no. 1, pp. 97–120, 1979.

[20] V. Zakian and M. Edwards, “Tabulation of constants for full grade i_{MN} approximants,” *Mathematics of Computation*, vol. 32, no. 142, pp. 519–531, 1978.

[21] V. Zakian, “Optimisation of numerical inversion of laplace transforms,” *Electronics Letters*, vol. 6, no. 21, pp. 677–679, 1970.

[22] V. Zakian and R. Littlewood, “Numerical inversion of laplace transforms by weighted least-squares approximation,” *The Computer Journal*, vol. 16, no. 1, pp. 66–68, 1973.

[23] J. Valsa and L. Brančik, “Approximate formulae for numerical inversion of laplace transforms,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 11, no. 3, pp. 153–166, 1998.

[24] N. Smith and L. Brancik, “Comparative study on one-dimensional numerical inverse laplace transform methods for electrical engineering,” *Elektrorevue*, vol. 18, no. 1, pp. 1–6, 2016.

[25] N. A.-Z. R-Smith and L. Brančík, “Convergence acceleration techniques for proposed numerical inverse laplace transform method,” in *2016 24th Telecommunications Forum (TELFOR)*. IEEE, 2016, pp. 1–4.

[26] L. Brančík and N. Smith, “Two approaches to derive approximate formulae of nilt method with generalization,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2015, pp. 155–160.

[27] A. M. Cohen, “Quadrature methods,” in *Numerical methods for Laplace transform inversion*. Springer, 2007, pp. 71–101.

[28] B. Davies and B. Martin, “Numerical inversion of the laplace transform: a survey and comparison of methods,” *Journal of computational physics*, vol. 33, no. 1, pp. 1–32, 1979.

[29] J. Abate, G. L. Choudhury, and W. Whitt, “On the laguerre method for numerically inverting laplace transforms,” 1995.

[30] K. L. Kuhlman, “Review of inverse laplace transform algorithms for laplace-space numerical approaches,” *Numerical Algorithms*, vol. 63, no. 2, pp. 339–355, 2013.

[31] “[electronic source] matlab - mathworks - matlab simulink,” (Access date: 25.04.2021). [Online]. Available: https://uk.mathworks.com/products/matlab.html

[32] “[electronic source] github - ghorvath78/iltcme: Inverse laplace transform based on concentrated matrix-exponential functions,” (Access date: 15.07.2020). [Online]. Available: https://github.com/ghorvath78/iltcme

[33] “[electronic source] github - dariagap/mgf_to_pdf_transform_methods,” (Access date: 14.02.2021). [Online]. Available: https://github.com/dariaGap/mgf_to_pdf_transform_methods