Available online at [www.sciencedirect.com](http://www.sciencedirect.com/)

[Electronic Notes in Theoretical Computer Science 337 (2018) 87–104](https://doi.org/10.1016/j.entcs.2018.03.035)

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

An Optimal Inverse Laplace Transform Method Without Positive and Negative Overshoot

– An Integral Based Interpretation [1](#_bookmark0)

Ill´es Horva´th[2](#_bookmark0)

*MTA-BME Information Systems Research Group, Budapest, Hungary*

Zsofia Talyigas[3](#_bookmark0)

*Technical University of Budapest, Budapest, Hungary*

Miklos Telek[4](#_bookmark0)

*Department of Networked Systems and Services, Technical University of Budapest, Budapest, Hungary*

Abstract

We propose a numerical inverse Laplace transformation method without overshoot which is derived from matrix exponential (ME) distributions with minimal coefficient of variation. We discuss the properties of the method through an integral based interpretation of numerical inverse Laplace transformation methods belonging to the Abate–Whitt framework. Compared to the previously applied non-overshooting alternative,

the “Erlang” method, the error of the proposed method improves from *O*(1*/n*) to *O*(1*/n*2) while it maintains the same computational complexity.

*Keywords:* numerical Inverse Laplace transformation, matrix exponential distribution, overshoot, minimal coefficient of variation.

# Introduction

The implementation of numerical inverse Laplace transformation in stochastic mod- els where the original function is a probability or probability distribution requires

1 This work is partially supported by the OTKA K123914 project.

2 Email:[horvath.illes.antal@gmail.com](mailto:horvath.illes.antal@gmail.com)

3 Email:[talyigas.zsofi@gmail.com](mailto:talyigas.zsofi@gmail.com)

4 Email:[telek@hit.bme.hu](mailto:telek@hit.bme.hu)

<https://doi.org/10.1016/j.entcs.2018.03.035>

1571-0661/© 2018 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

that the inverse transformation procedure preserves the property that the numer- ical inverse transform is between zero and one. Most numerical inverse transform methods generate positive or negative overshoot which might result in numerical inverse transform values below zero or above one, which is an unacceptable quali- tative error in stochastic modelling. That is why we focus on the numerical inverse transformation procedures without overshoot in this work.

There are plenty of numerical inverse Laplace transformation methods published in the literature, whose survey is out of the scope of the current work. For a rela- tively recent initial pointer we refer to [[13](#_bookmark50)] and the references therein, however the application field of our focus is significantly different. While a large part of the literature, including mentioned one, lists plenty of methods, a clear and systematic classification of these methods is not easy to find. One exception that significantly helped the intuitive understanding of the authors is by Abate and Whitt in [[8](#_bookmark42)], which we refer to as Abate–Whitt framework. Indeed, this framework implicitly defines function families in which various optimizations can be performed in order to obtain optimal inverse Laplace transformation methods. We propose a procedure which is based on the most general function family of the Abate–Whitt framework (referred to as Class III, later on) where we adopt a restriction coming from the non-overshooting requirement. It turns out that the obtained functional optimiza- tion problem is known in probability theory as the problem of Matrix exponential distribution with minimal coefficient of variation. As a result, in the current work we apply available extremal results for Matrix exponential distribution for optimal numerical inverse Laplace transformation without overshooting in the Abate–Whitt framework.

The rest of the paper is organized as follows. Section [2](#_bookmark1) provides general intro- duction to inverse Laplace transformation and presents the Abate–Whitt framework together with some of its methods. Section [3](#_bookmark6) is devoted to the integral based inter- pretation of the framework. The proposed matrix exponential distribution based procedure is described in Section [4](#_bookmark17) and its performance is compared with other methods from the Abate–Whitt framework in Section [5](#_bookmark25). The paper is concluded in Section [6.](#_bookmark37)

# Inverse Laplace transformation and the Abate–Whitt framework

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

*∗* ∫ *∞*

*h* (*s*)=

*e*

*t*=0

*−st*

and the inverse transform problem is to find an approximate value of *h* at point *T*

*h*(*t*)d*t.* (1)

(i.e., *h*(*T* )) based on *h∗*(*s*).

**Remark 2.1** We assume that ∫ *∞ e−sth*(*t*)d*t* is finite for Re(*s*) *>* 0 thus *h∗*(*s*) is

*t*=0

well-defined by ([1](#_bookmark2)) for Re(*s*) *>* 0. For Re(*s*) *≤* 0, *h∗*(*s*) is the analytic continuation

of *h∗*(*s*) for Re(*s*) *>* 0.

**Remark 2.2** We assume that *h*(*t*) is real in this work. As a result, *h∗*(*s*¯)= *h*¯*∗*(*s*) and *h∗*(*s*¯)+ *h∗*(*s*)= 2Re(*h∗*(*s*)).

Our main interest is inverse Laplace transformation without overshooting. Due to this reason we restrict our attention to the Abate–Whitt framework which we summarize below.

* 1. *The Abate–Whitt framework*

The idea is to approximate *h* by a finite linear combination of the transform values, via

*h*(*T* ) *≈ hn*(*T* ) := Σ

*n*

*ηk h∗*

*βk*

, *T >* 0*,* (2)

*T T*

*k*=1

where the nodes *βk* and weights *ηk* are complex numbers, which depend on *n*, but not on the transform *h∗* or the time argument *t*. This framework was introduced and investigated by Abate and Whitt in [[8](#_bookmark42)]. When *h*(*t*) in ([1](#_bookmark2)) is real valued it can be approximated by the real part of the weighted transform values:

Re(*h*(*T* )) *≈* Re(*hn*(*T* )) = Σ

Re *η*

*n*

*k h∗*

*βk .*

*T T*

*k*=1

In the special case when there is a complex conjugate pair among the nodes and weights (that is, *ηi* = *η*¯*j* and *βi* = *β*¯*j* ) then

*ηi h∗* *βi* + *ηj h∗* *βj* = 2Re *ηi h∗* *βi* *.*

*T*

*T*

*T*

*T*

*T*

*T*

We consider three classic algorithms of the Abate–Whitt framework: the Gaver– Stehfest method, the Euler method and the Talbot method. All three were already investigated by Abate and Whitt [[8](#_bookmark42)]. We now give a different approach and also compare them with the matrix exponential method we propose.

The Gaver–Stehfest algorithm is based on the sequence of Gaver approximants, derived by Gaver [[5](#_bookmark43)]. Since the convergence of the Gaver approximants was only logarithmic, it needed acceleration. A linear acceleration method was proposed by Stehfest [[10](#_bookmark47)], using the Salzer acceleration scheme. As it is observed in [[8](#_bookmark42)], the accelerated version fits in framework ([2](#_bookmark3)), so we can investigate it in this class.

Euler’s method and its derivation can be found e.g. in [[1](#_bookmark38)]. In [[8](#_bookmark42)] it is rewritten in a form that fits the structure of ([2](#_bookmark3)).

Talbot’s algorithm was first introduced by A. Talbot in 1979 in [[11](#_bookmark48)]. The version of the form of ([2](#_bookmark3)) is derived in [[8](#_bookmark42)] based on an earlier work of Abate and Valko [[12](#_bookmark49)]. The three methods in question all approximate *h*(*t*) by *hn*(*t*), where *hn*(*t*) has structure ([2](#_bookmark3)) with weights *ηk* and nodes *βk*, *k* = 1*,* 2*,... n*. The weights and nodes

for the considered algorithms are given below.

*Gaver–Stehfest method (for even n)*

*βk* = *k* ln(2)*,* 1 *≤ k ≤ n*

min(*k,n/*2)

Σ

*ηk* = ln(2)(*−*1)*n/*2+*k*

*jn/*2+1 *n/*2 2*j* *j*

*,* 1 *≤ k ≤ n*

*j*=*[*(*k*+1)*/*2*♩*

(*n/*2)! *j*

*j k − j*

where *[x♩* is the greatest integer less than or equal to *x*.

*Euler method (for odd n)*

*βk* =

(*n −* 1) ln(10)

+ *πi*(*k −* 1)*,* 1 *≤ k ≤ n*

6

*ηk* = 10(*n−*1)*/*6(*−*1)*kξk,* 1 *≤ k ≤ n*

where

1

*ξ*1 = 2

*ξk* = 1*,* 2 *≤ k ≤* (*n* + 1)*/*2

1

*ξn* = 2(*n−*1)*/*2

*k*

*ξn−k*

= *ξn−k*+1

2*−*(*n−*1)*/*2 (*n −* 1)*/*2 for 1 *≤ k <* (*n −* 1)*/*2*.*

*Talbot method*

2*n*

*β*1 = 5

*β* = 2(*k −* 1)*π* cot (*k −* 1)*π* + *i* *,* 2 *≤ k ≤ n*

*k* 5 *n*

* + 1. *β*1

*η*1 = 5 *e*

1+

cot

*— i* cot

*eβk ,* 2 *≤ k ≤ n.*



*ηk* =

1+ *i*

5

(*k−*1)*π*

*n*

(*k−*1)*π* 2

*n*

(*k−*1)*π*

*n*

Figure [1](#_bookmark5) depicts the location of the *βk* nodes of the three methods. The Gaver– Stehfest method operates with real *βk* values which is beneficial when complex arithmetic is not supported, while the Euler and Talbot methods apply complex *βk* values as well. Both assume nodes with positive imaginary parts. The Talbot method applies *βk* values with negative real part as well, which is commonly outside of the convergence region of ([1](#_bookmark2)), but the analytic continuation of *h∗*(*s*) may exist.

Σ

**Remark 2.3** The set of real valued functions *k ηke−βkt* with potentially complex valued coefficients has the following real representations.

**Class I** If both *ηk* and *βk* are real then Σ*k ηke−βkt* is a real representation.

Im

Re

Euler Gaver

Talbot

30

25

20

15

10

5

-30 -20 -10 10

Figure 1. Location of *βk* nodes on the complex plane for Gaver (*n* = 10), Euler (*n* = 11), Talbot (*n* = 10) methods

**Class II** If *ηk* is real and *βk* is complex then

Re Σ *ηke−βkt* = Σ *ηke−bkt* cos(*ωkt*) is its real representation, where *βk* = *bk* + *iωk*.

*k*

*k*

**Class III** If both *ηk* and *βk* are complex then

Re Σ *ηke−βkt* = Σ *ake−bkt* cos(*ωkt* + *φk*)

*k*

*k*

is its real representation, with Re(*ηke−βkt*) = *ake−bkt* cos(*ωkt* + *φk*) where *βk* = *bk* + *iωk* and *ak, φk* are real and obtained from the real and imaginary parts of *ηk* [[6](#_bookmark44)].

The Gaver–Stehfest method falls into Class I, the Euler method falls into Class II, the Talbot method falls into Class III. The matrix exponential method (described in detail in Section [4.4](#_bookmark23)) falls into Class III.

# Integral interpretation of the Abate–Whitt frame- work

For Re(*βk*) *>* 0*, ∀k*, we can reformulate the inverse Laplace transformation methods of the Abate–Whitt framework as

1 Σ *∗*

*n*

*βk*

1 Σ ∫ *∞*

*— βk t*

∫ *∞ n*

*hn*(*T* )= *T*

*k*=1

*ηkh*

= *ηk*

*T T*

*n*

0

*k*=1

*e T h*(*t*)d*t* =

0

*h*(*t*)*fT* (*t*)d*t,* (3)

where the numerical approximation of the Laplace inverse at point *T* is obtained as the integral of the original function, *h*(*t*), with

*fn*(*t*)= 1 Σ *η e− βk t.* (4)

*n*

*T*

*T*

*k*

*T*

*k*=1

If *fn*(*t*) was the Dirac impulse function at point *T* then the Laplace inversion would be perfect, but depending on the order of the approximation (*n*), the applied inverse transformation method (weights *ηk*, nodes *βk*) and the time point (*T* ), function *fn*(*t*)

only approximates the Dirac impulse function with a given accuracy.

*T*

*T*

The shape of *fn*(*t*) is similar to the shape of

*T*

*n*

*f* (*t*)= Σ *ηke* (5)

*n −βkt*

1

*k*=1

because, according to ([4](#_bookmark8)),

*fT* (*t*)= *T f*1

*.* (6)

*n* 1 *n* *t*

*T*

8

15

Euler

Gaver

10

5

0.5

1.0

1.5

2.0

-5

Euler

6 Gaver

4

2

0.5 1.0 1.5 2.0

-2

Figure 2. *fn*(*t*) for the Gaver–Stehfest (*n* = 10) and the Euler (*n* = 11) methods for *T* =1

*T*

Figure 3. *fn*(*t*) for the Gaver–Stehfest (*n* = 22) and the Euler (*n* = 23) methods for *T* =1

The Re(*βk*) *>* 0*, ∀k* property holds for the Euler and the Gaver–Stehfest meth- ods, but not for the Talbot method (c.f. Fig. [1](#_bookmark5)). Figure [2](#_bookmark11) and [3](#_bookmark11) depicts the approximate Dirac impulse functions for *n* = 11 and *n* = 23 for the Euler and *n* = 10 and *n* = 22 for the Gaver–Stehfest methods at *T* = 1. In accordance with expectations, the functions get visually “narrower” (more concentrated) for higher orders in both cases. For a more quantitative comparison we collected some properties of the functions *fn*(*t*) in Table [1.](#_bookmark13)

*T*

*T*

We introduce a measure of concentration from probability theory, the coefficient of variation (cv). For a random variable *X* with probability density function *f* , cv(*X*) is defined as

var(*X*) cv(*X*)= (E(*X*))2

= *m*2 *−* 1*,* where *m*

1

*m*

2

*i*

= *tif* (*t*)d*t.*

*t*

∫

Our goal is to measure how concentrated *fn* is. Note that *fn* is not necessarily

*T T*

a proper normalized pdf, since we allow its integral to be different from 1 and *fn*

*T*

may take negative values. However, it is possible to introduce cv for *fn*, and it will be informative later on. The coefficient of variation for *fn* is defined as

*T*

*T*

∫

cv = *m*0*m*2 *−* 1*,* with *m*

*m*

2

*i*

1

= *ti|fn*(*t*)*|*d*t. t*

(The use of *m*0 is necessary if *fn* is not normalized.)

*T*

*T*

We introduce some further measures that help quantify the concentration of *fn*: *peak* denotes the maximal value of *fn*, while *min* and *max* denote the first point and the last point where *fn*(*t*) = 0*.*01*peak* (where 0*.*01 is arbitrarily chosen). The main impulse of the Euler method is much higher and narrower than the one for the Gaver–Stehfest method for both orders as indicated by the *peak* values and the distance of the zeros (*fn*(*t*) = 0) closest to *t* = 1. This trend is also visible on the *min* and *max* values (*min* increases/*max* decreases with *n*, and Euler is narrower than Gaver–Stehfest), except the *min* value of the Euler method. The particular value, *min* = 0*.*376047, is due to a local maximum only a little larger than 0*.*01*max* (if that peak was less than 0*.*01*max* then *min* would be around 0*.*474721, see Fig. [4](#_bookmark12)), but the increase of *min* for the Euler method is significantly slower than the one for the Gaver–Stehfest method.

*T*

*T*

*T*

*T*

For a properly scaled numerical inverse Laplace transformation method, *fn*(*t*)

*T*

should preserve the integral property of the Dirac impulse, that is,

*fn*(*t*)d*t* = 1*,*

*T*

∫

*t*

and the weight and node parameters of the Gaver–Stehfest and the Euler methods are indeed such that the unit integral is ensured for all orders. Having the unit integral property, a larger *m*0 indicates that the function *fn*(*t*) is “waving” between positive and negative values more. *m*0 is higher and more rapidly increasing with the order for the Euler method. This fact together with the slowly increasing *min* value indicate that the *fn*(*t*) function of the Euler method is sharply alternating mainly before the main impulse and this strong waving behaviour gets more significant with increasing order (see Figures [2](#_bookmark11) and [3](#_bookmark11) again).

*T*

*T*

A cumulative effect of the properties discussed above is reflected in the cv values. For *n* = 11, the Euler method provides much lower cv due to the relatively narrow non-vanishing curve (from *min* to *max*), but with increasing order the *min* value of the Euler method does not increase as fast as for the Gaver–Stehfest method and the waving of the Euler curve from *min* to the main impulse gets more and more dominant. As a consequence of these two effects, the cv value of the Euler method decreases much slower than the one of the Gaver–Stehfest method.

As we have seen, for a given order the *fn*(*t*) function by the Euler method is much narrower than the one by the Gaver–Stehfest method. This difference comes also from the fact that the *βk* values of the Gaver–Stehfest method are real, while they are complex in case of the Euler method, while the *ηk* values are real in both

*T*

cases (c.f., Remark [2.3](#_bookmark4)). As a result *fn*(*t*)= Σ*n ak e− bk t* (with real *a ,b* ) for the

*T*

*k*=1 *T*

*T*

*k*

*k*

1.0

Euler

0.01 peak

0.5

0.25

0.30

0.35

0.40

0.45

0.50

-0.5

Figure 4. Local peak of *fn*(*t*) at around 0*.*01*peak* (horizontal line) with Euler *n* = 11, *T* =1

*T*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | *m*0 | cv | *peak* | *min* | *max* |
| 10  22 | Gaver | | | | |
| 1.32819  1.46379 | 0.261505  0.0950682 | 1.65108  3.53038 | 0.0969432  0.359999 | 3.94368  2.10102 |
| 11  23 | Euler | | | | |
| 1.79425  2.47539 | 0.0395198  0.0299859 | 8.26811  17.6272 | 0.383763  0.376047 | 1.34461  1.16585 |

Table 1

Properties of *fn*(*t*) with the Gaver–Stehfest (*n* = 10*,* 22) and the Euler (*n* = 11*,* 23) methods at *T* =1

*T*

Gaver–Stehfest method and *fn*(*t*)= Σ*n ak e− bk t* sin( *ωk t*) (with real *a ,b ,ω* ) for

*T*

*k*=1 *T*

*T*

*T*

*k*

*k*

*k*

the Euler method, that is, the Euler method approximates the Dirac impulse using a larger set of exponential functions.

Up to this point we investigated the main properties of the *fn*(*t*) functions. Below we study the effect of these properties on the inverse transform computed by the Gaver–Stehfest and the Euler methods. The Gaver–Stehfest and the Euler Laplace inversion of the unit step function at 1 (denoted by *h*(*t*) and with Laplace transform) *h∗*(*s*)= *e−s* , are depicted in Figure [5](#_bookmark14) for *n* = 10*/*11 and in Figure [6](#_bookmark14) for

*T*

*s*

*n* = 22*/*23.

1.2

1.0

Euler

Gaver

0.8

0.6

0.4

0.2

0.5

1.0

1.5

2.0

2.5

3.0

1.2

1.0

Euler

Gaver

0.8

0.6

0.4

0.2

0.5

1.0

1.5

2.0

2.5

3.0

Figure 5. Inverse Laplace transform of the unit step function with Gaver–Stehfest (*n* = 10) and Euler (*n* = 11) methods

Figure 6. Inverse Laplace transform of the unit step function with Gaver–Stehfest (*n* = 22) and Euler (*n* = 23) methods

The increase of the inverse Laplace curve at around *t* = 1 is the steepest for the Euler method with *n* = 23 and slowest for the Gaver–Stehfest method with *n* = 10. This steepness is associated with the width of the main peak in the *fn*(*t*) function

*T*

(see also the related discussion on the waves based on ([7](#_bookmark16))).

In general, a wider *fn*(*t*) function makes the *h*(*t*)*fn*(*t*) product non-vanishing

*T* ∫ *T n*

for *t* values far from *T* and as a consequence the

*t h*(*t*)*fT* (*t*)d*t* integral is affected

6 11

8

*f* 11 (t)

1

6

*f* 11 (t)

10

4

2

2

4

6

8

10

12

-2

*f* (t)

1.14

4

2

0.5 1.0 1.5 2.0

-2

Figure 7. *f* 11

(*t*) with the Euler method. Its inte-

Figure 8. Scaling at *T* = 10: *f* 11(*t*) and *f* 11(*t*) with

1*.*14

gral from 1 to *∞* is maximal.

the Euler method

1 10

by *h*(*t*) for values of *t* far from *T* as well. The error caused by the width of *fn*(*t*) is less significant if *h*(*t*) is almost constant around *T* , but it gets to be very significant when *h*(*t*) sharply changes around *T* in the range of the width of *fn*(*t*) (like in the

*T*

*T*

case of the unit step function).

As a counterpart of the sharp increase the Euler inverse Laplace curve has a strongly waving behaviour mainly after the jump, while the Gaver–Stehfest curve settles more quickly. To better relate *fn*(*t*) and *hn*(*T* ) in this special case (where *h*(*t*)=1 for *t >* 1 and zero otherwise) we write

*T*

*∞*

∫

*hn*(*T* )=

0

*h*(*t*)*fn*(*t*)d*t* =

*∞*

*fn*(*t*)d*t* (7)

∫

*T*

1

and depict *f* 11

*T*

1*.*14

(*t*) of the Euler method in Figure [7](#_bookmark15), because the first peak of the

Euler *n* = 11 inverse Laplace curve on Figure [5](#_bookmark14) after the step is at *t* = 1*.*14.

The integral of *f* 11

1*.*14

(*t*) from 1 to infinity is maximal, because it starts right at the

end of the largest negative wave. Following the same reasoning one can see that the waves of the Euler inverse Laplace curve after the step are related with the waves of the *fn*(*t*) function before the main impulse and similarly the waves of the Euler inverse Laplace curve before the step are related with the waves of the *fn*(*t*) function after the main impulse. Since the *fn*(*t*) function of the Euler method is heavily alternating before the main impulse the Euler inverse transform curve of any jump have small waves before the jump and larger waves after the jump (as it is seen e.g. in Fig. [5](#_bookmark14) and [6](#_bookmark14)).

*T*

*T*

*T*

The unit step function at 1 is a valid probability function in the sense that 0 *≤ h*(*t*) *≤* 1*, ∀t >* 0. As it is discussed in details the alternating nature of *fn*(*t*) generates waves in the transform domain curve, which goes below zero before the step and above one after the step. That is, both, the Euler and the Gaver–Stehfest method overshoots at a jump of *h*(*t*) and provide numerical inverse outside the [0*,* 1] interval. To obtain a numerical inverse of the unit step function inside the [0*,* 1] interval, *fn*(*t*) should be nonnegative, which is violated by both the Euler and the Gaver–Stehfest methods and which is the subject of the next sections.

*T*

*T*

# Matrix exponential distributions

* 1. *Probability background*

Probability density functions (pdf) similar to the form ([5](#_bookmark9)) have been examined for a long time [[2](#_bookmark39)]. The class of matrix exponential distributions of order *N* , denoted ME(*N* ), contains all distributions *X* with pdf of the form

*fX* (*t*)= *−α***A***e****A****t***1***, t ≥* 0*,* (8)

where *α* is a row vector of length *N* , **A** is a matrix of size *N × N* and **1** is a column vector of ones of size *N* .

Rewriting ([8](#_bookmark18)) based on the Jordan decomposition of **A**, we obtain the following general form of ME functions:

*k li−*1

Σ Σ

*fX* (*t*)= *ci,jtjeλit,* (9)

*i*=1 *j*=0

where *λ*1*,..., λk* are eigenvalues of **A**, and *λi* has multiplicity *li*. The form ([9](#_bookmark19)) is not quite consistent with ([5](#_bookmark9)) since ([9](#_bookmark19)) may contain polynomial factors as well; nevertheless, if *fn*(*t*) (as defined in ([5](#_bookmark9))) is nonnegative, then *fn*(*t*) *∈* ME(*N* ).

1 1

For a given *fX* (*t*), *α* and **A** are not unique; if ([8](#_bookmark18)) holds, we say that *X* is matrix

exponentially distributed with representation (*α,* **A**), or *X ∼* ME(*α,* **A**) for short. *fX* (*t*) is normalized (that is, *∞ fX* (*t*)d*t* = 1) iff *α***1** = 1. This is usually assumed. The nonnegativity of *fX* (*t*) does not follow from either ([8](#_bookmark18)) or ([9](#_bookmark19)), and needs to be checked separately.

0

∫

A special subclass of ME(*N* ) is the phase-type distributions PH(*n*): if, in addi- tion to ([8](#_bookmark18)), we also assume that

* *αi ≥* 0,
* *Ai,j ≥* 0 for *i /*= *j*, *Aj,j <* 0,
* *A***1** *≤* 0

then we say *X* is phase type (PH) distributed. Phase type distributions have a nice stochastic interpretation: *X* is the absorption time of an absorbing continuous time Markov chain with a finite state space of size *N* with infinitesimal generator **A** and initial distribution *α*. As a consequence, nonnegativity of *fX* (*t*) follows from the additional assumptions.

Evidently, PH(*N* ) *⊆* ME(*N* ); however, the difference is not well understood in general.

* 1. *Concentrated PH and ME distributions*

For the class PH(*N* ), the minimal cv is known. Aldous and Shepp [[4](#_bookmark41)] have proven the following theorem:

**Theorem 4.1**

min

*X∈PH*(*N* )

1

cv(*X*)=

*N*

*,* (10)

*and the minimum is obtained for the Erlang distribution with parameters* (*N, λ*)

*where λ >* 0 *is arbitrary.*

For the class ME(*N* ), the minimal cv has been calculated in [[6](#_bookmark44)], with the follow- ing conjecture based on numerical optimization for odd values of *N* up to *N* = 47:

min

*X∈*ME(*N* )

2

cv(*X*) *∼ N* 2 *,* (11)

and the optimal matrix exponential distribution was of the following form (assuming odd *N* ):

*f*ME(*t*)= *c* e*−λt*

(*N−*1)*/*2

*i*=0

cos2(*ωt − φi*) (12)

with the real values of *c, λ, ω* and *φ*1*,..., φ*(*N−*1)*/*2 obtained from numerical opti- mization.

* 1. *Erlang distribution-based inverse Laplace transformation*

In this paper, we would like to emphasize the benefit of using ME-based inverse Laplace transformation compared to the commonly applied non-overshooting in- verse Laplace transformation method, based on Erlang distribution [[3,](#_bookmark40)[9](#_bookmark45)]. The Er- lang distribution has pdf

*λN f*Erlang(*t*)= (*N −* 1)! *t*

*N−*1

*e−λt,*

which is not consistent with ([5](#_bookmark9)) due to the polynomial factor *tN−*1.

For the inverse Laplace transformation, the polynomial factor *tN−*1 corresponds

to

d*N−*1 *∗*

d*sN−*1 *h* (*s*)*,*

which technically does not fit in Class III. While calculating or approximating the (*N −* 1)th derivative may be infeasible in general, efficient methods are available for some applications. For example, Asmussen, Avram and Usabel [[3](#_bookmark40)] use it to approximate the probability of ruin in a fluid model before a fixed time *T* .

* 1. *ME distribution-based inverse Laplace transformation*

Our main focus is the application of the concentrated ME distribution described in

([12](#_bookmark22)) for inverse Laplace transformation. ([12](#_bookmark22)) can be rewritten in a form consistent

with either ([8](#_bookmark18)) or ([5](#_bookmark9)):

(*N−*1)*/*2 *N n*

*f*ME(*t*)= *c* e*−λt*

cos2(*ωt − φi*)= Σ *ηie−βit* = *η*1*e−β*1*t* +2 Σ Re *ηie−βit*

*i*=0

*i*=1

*i*=2

(13)

where *n* = (*N* + 1)*/*2, *η*1*, β*1 are real, and the values *β*2*,..., βn* have positive imagi- nary parts. For details, see the Appendix of [[6](#_bookmark44)]. Using the function ([13](#_bookmark24)) in ([4](#_bookmark8)) and

([3](#_bookmark7)) is what we call the ME based inverse Laplace transformation method.

A full list of the values of *η*1 *... ηn, β*1*,..., βn* for each *n* up to *n* = 24 is available online at [[7](#_bookmark46)]. As an example, we include the values for *n* = 4 (*N* = 7). The imaginary parts of *β*1*,..., βn* form an arithmetic sequence due to the special form of ([12](#_bookmark22)) (see the Appendix in [[6](#_bookmark44)]).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | 1 | 2 | 3 | 4 |
| *ηi* | 38*.*5032 | *−*18*.*9855 *−* 23*.*2984*i* | *−*2*.*70326 + 13*.*374*i* | 2*.*47829 *−* 1*.*37694*i* |
| *βi* | *−*3*.*93763 | *−*3*.*93763 + 3*.*48448*i* | *−*3*.*93763 + 6*.*96896*i* | *−*3*.*93763 + 10*.*4534*i* |

# Numerical comparison with the ME based method

In this section we compare the proposed matrix exponential method (ME method) with the three known algorithms from the Abate–Whitt framework (Euler, Talbot, Gaver–Stehfest) via numerical examples. It is investigated how well the methods approximate *h* when *h∗* is given. We compare the different algorithms for the same order *n*, where *n* refers to the number nodes at which *h∗* is evaluated. Note that the Euler method is only applicable for odd order and the Gaver-Stehfest method is only applicable for even order. Thus for each *n* tested, either the order of the Euler method or the order of the Gaver–Stehfest method is shifted by 1 compared to the others.

Table [2](#_bookmark26) contains the Laplace transform pairs of the list of test functions, *h* and

*h∗*.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *h*(*t*) | *e−t* | sin *t* | 1(*t >* 1) | 1(*t >* 1)*e*1*−t* | *[t♩* | square wave function |
| *h∗*(*s*) | 1  1+*s* | 1  *s*2+1 | 1 *e−s/*2  *s* | *e−s*  1+*s* | 1 1  *s es−*1 | 1 1  *s es*+1 |

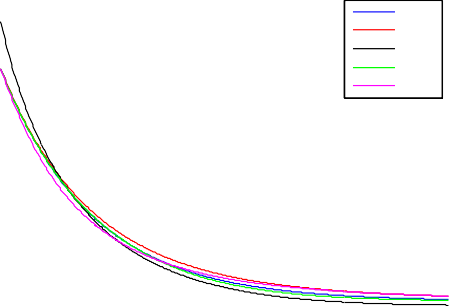
Table 2

Set of test functions

Figures [9](#_bookmark27) and [10](#_bookmark27) depict the approximations for *h*(*t*) = *e−t* with the various methods for order *n* =3 (*n* = 4 for Gaver–Stehfest) and *n* =5 (*n* =6 for Gaver– Stehfest) respectively. The approximations are relatively accurate already for order *n* =5 for all methods.

Figures [11](#_bookmark28) and [12](#_bookmark28) depict the approximations for *h*(*t*)= sin(*t*) with the various methods for order *n* = 13 (*n* = 14 for Gaver–Stehfest) and *n* = 24 (*n* = 25 for

1.2



1

0.8

0.6

0.4

0.2

0

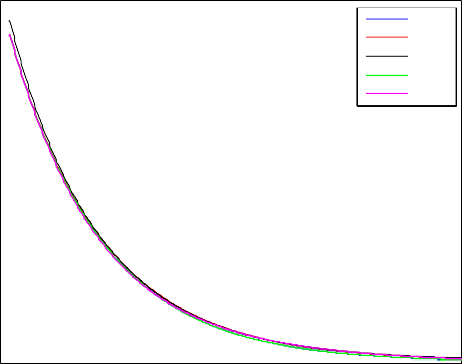
−0.2

Exact ME

Euler Talbot Gaver

1

0.9



Exact ME

Euler Talbot Gaver

0.8

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 9. *h*(*t*)= *e−t* for order *n* = 3 (Gaver: *n* = 4)

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 10. *h*(*t*)= *e−t* for *n* = 5 (Gaver: *n* = 6)

Euler) respectively. Note that the approximation gets worse for larger values of *T* . This is inherent to all Abate–Whitt class methods since in ([3](#_bookmark7)), the approximant for the Dirac impulse function *fn*(*t*) is scaled from *fn*(*t*) according to ([6](#_bookmark10)) and thus

*T* 1

*fn*(*t*) is less concentrated for larger values of *T* . Figure [8](#_bookmark15) demonstrates this effect

*T*

for *f* 11(*t*) and *f* 11(*t*) with the Euler method. Intuitively, when the main impulse of

1 10

*fn*(*t*) is wider than a period of the sinus wave, then the integral in ([3](#_bookmark7)) averages out

*T*

the waves.

That said, the rate at which the approximation becomes worse as *T* increases is different for different methods. The Talbot and Gaver–Stehfest methods provide relatively poor approximations with the functions near constant 0 after only two periods for *n* = 13 and three periods for *n* = 24.

The Euler method provides relatively accurate approximation for 4 periods for *n* = 14 and 7 periods for *n* = 25; the overshooting is visible as the Euler-method also provides values larger than 1 and values smaller than *−*1 (outside the range of the original function sin(*t*)). The period is also distorted for larger values of *t*, shifting the local extrema (due to the strong asymmetry of *fn*(*t*) with the Euler method). Also, the “amplitude” of the function provided by the Euler-method decays rather abruptly.

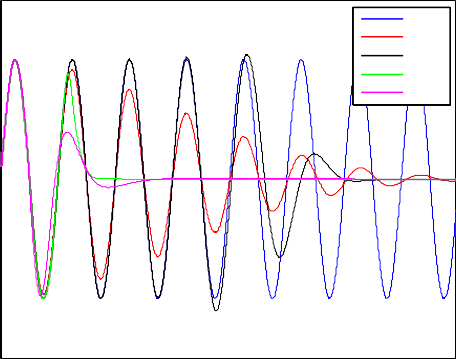
*T*

The matrix exponential method retains the periodic nature of sin(*t*), it does not overshoot, retains the period length as well, but the amplitude decays gradually. Overall, the Euler method provides the best approximation for sin(*t*), but with some overshooting.

Figures [13](#_bookmark29) and [14](#_bookmark29) depict the approximations for the step function *h*(*t*)= 1(*t >* 1) for order *n* = 13 (*n* = 14 for Gaver–Stehfest) and *n* = 24 (*n* = 25 for Euler) respectively. The Talbot method was omitted since it would sample values of *h∗*(*s*) outside the domain of convergence of the integral. The other methods provide a reasonable approximation; note that again, the Gaver–Stehfest and the Euler methods overshoot. Note also that the “size” of the overshooting does not decrease with the order vertically, only horizontally. The matrix exponential method does not overshoot, and provides the best approximation overall.

Figures [15](#_bookmark30) and [16](#_bookmark30) depict the approximations for the shifted exponential function

1.5



1

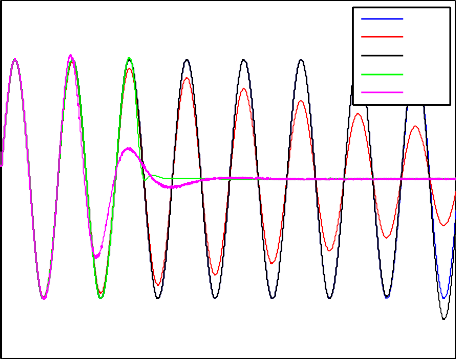
0.5

Exact ME

Euler Talbot Gaver

1.5

1



0.5

Exact ME

Euler Talbot Gaver

0 0

−0.5 −0.5

−1 −1

−1.5

0 5 10 15 20 25 30 35 40 45 50

−1.5

0 5 10 15 20 25 30 35 40 45 50

Figure 11. *h*(*t*) = sin(*t*) for *n* = 13 (Gaver: *n* = 14)

1.2



Exact

Figure 12. *h*(*t*) = sin(*t*) for *n* = 24 (Gaver: *n* = 25)

1.2



Exact

1

0.8

ME

Euler Gaver

1

0.8

ME

Euler Gaver

0.6 0.6

0.4 0.4

0.2 0.2

0 0

−0.2

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 13. *h*(*t*) = 1(*t >* 1) for *n* = 13 (Gaver:

*n* = 14)

−0.2

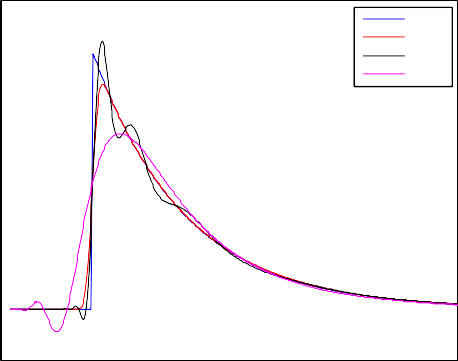
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 14. *h*(*t*) = 1(*t >* 1) for *n* = 24 (Euler:

*n* = 25)

1(*t >* 1)*e*1*−t* for order *n* = 13 (*n* = 14 for Gaver–Stehfest) and *n* = 24 (*n* = 25 for Euler) respectively. The Talbot method was omitted again. Similarly to Figures [13](#_bookmark29) and [14](#_bookmark29), the Gaver–Stehfest and the Euler methods overshoot again, and the matrix exponential method provides the best approximation overall.

1.2



1

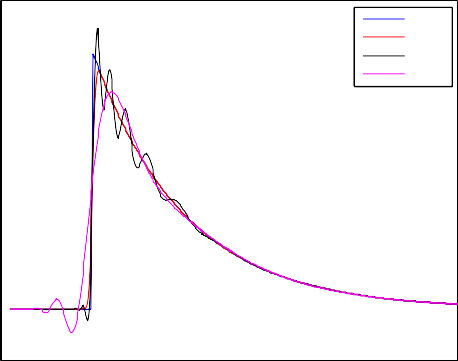
0.8

Exact ME

Euler Gaver

1.2

1



0.8

Exact ME

Euler Gaver

0.6 0.6

0.4 0.4

0.2 0.2

0 0

−0.2

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 15. *h*(*t*) = 1(*t >* 1)*e*1*−t* for *n* = 13 (Gaver:

*n* = 14)

−0.2

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

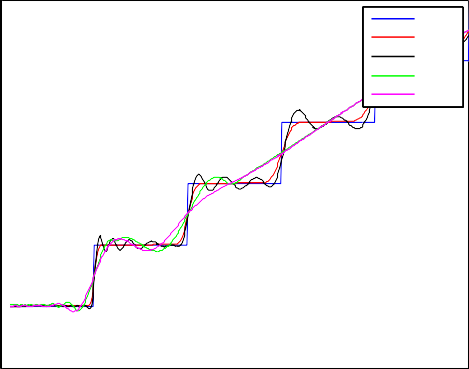
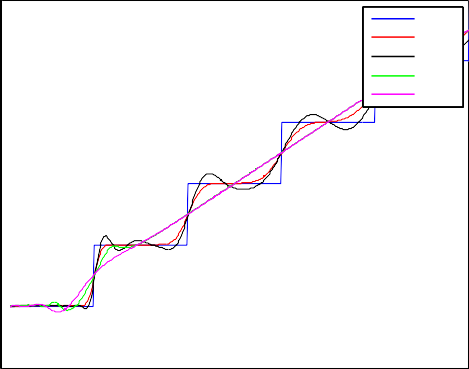
Figure 16. *h*(*t*) = 1(*t >* 1)*e*1*−t* for *n* = 24 (Euler:

*n* = 25)

Figures [17](#_bookmark31) and [18](#_bookmark31) depict the approximations for the stepping function *h*(*t*)= *[t♩*

for order *n* = 13 (*n* = 14 for Gaver–Stehfest) and *n* = 24 (*n* = 25 for Euler) respec- tively. Note that the error increases with *t*, and asymptotically, all approximating functions become linear (see Figure [19](#_bookmark32)). The Gaver–Stehfest method provides a poor approximation by converging very fast to the linear asymptote; the other methods provide a reasonable approximation, but both the Euler and Talbot meth- ods overshoot, and the Euler method has a considerable error even at the middle of the steps (halfway between jumps). The matrix exponential method provides a good approximation without overshooting.

5 5



Exact ME

4 Euler 4

Talbot Gaver

3 3

Exact ME

Euler Talbot Gaver

2 2

1 1

0 0

−1

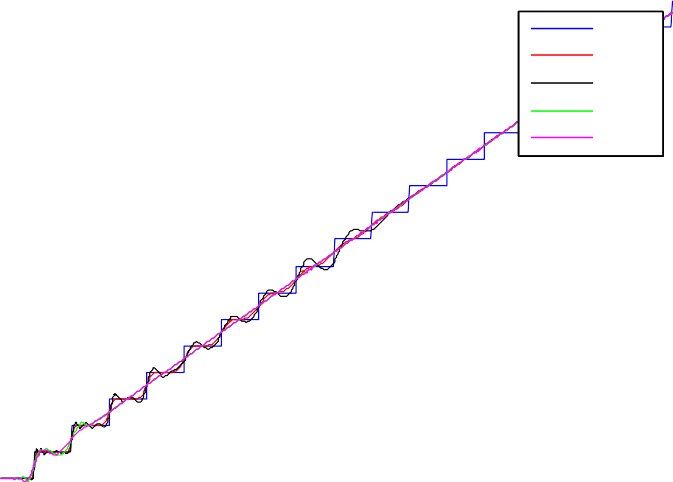
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

−1

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 17. *h*(*t*)= *[t♩* for *n* = 13 (Gaver: *n* = 14) Figure 18. *h*(*t*)= *[t♩* for *n* = 24 (Euler: *n* = 25)

18



Exact

16 ME

Euler

14 Talbot

12 Gaver

10

8

6

4

2

0

−2

0 2 4 6 8 10 12 14 16 18

Figure 19. *h*(*t*)= *[t♩* for *n* = 24 (Euler: *n* = 25)

Figures [20](#_bookmark33) and [21](#_bookmark33) depict the approximations for the square wave function for order *n* = 13 (*n* = 14 for Gaver–Stehfest) and *n* = 24 (*n* = 25 for Euler and Talbot) respectively, with Figure [22](#_bookmark34) zooming in on smaller values of *t*. Similarly to *h*(*t*) = sin(*t*), the Talbot and Gaver methods provide a poor approximation, with both functions close to constant after very few periods. The Euler method provides

relatively accurate approximation for 4 periods for *n* = 14 and 7 periods for *n* = 25. The overshooting of the Euler, Talbot and Gaver–Stehfest methods is evident again. Similarly to *h*(*t*)= sin(*t*), the matrix exponential method retains the periodic-

ity (along with the period length), does not overshoot, but the amplitude decays gradually.

1.4

1.2

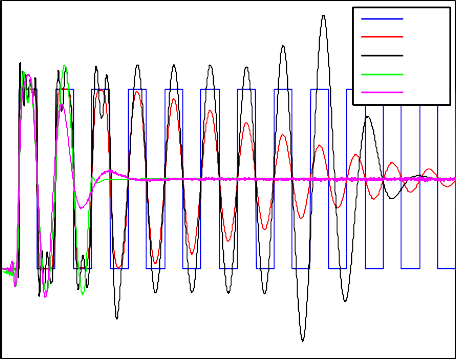
1

Exact ME

Euler Talbot Gaver

1.5

1



Exact ME

Euler Talbot Gaver

0.8

0.6 0.5

0.4

0.2 0

0

−0.2

0 2 4 6 8 10 12 14

−0.5

0 5 10 15 20 25

Figure 20. square wave function for *n* = 13 (Gaver:

*n* = 14)

1.2

1

0.8

Figure 21. square wave function for *n* = 24 (Euler, Talbot: *n* = 25)

Exact ME

Euler Talbot Gaver

0.6

0.4

0.2

0

−0.2

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Figure 22. square wave function for *n* = 24 (Euler, Talbot: *n* = 25)

In Table [3](#_bookmark35), the absolute maximal error for *h*(*t*)= *e−t* is compared for the various methods for various orders. For this measure, the Talbot method outperforms the other, with the matrix exponential method and the Gaver–Stehfest method performing similarly and the Euler method worse than the others. (Note that the absolute maximal error makes sense only when *h*(*t*) is continuous and has a linear asymptote, which is valid only for *e−t* from among the test functions examined.)

In Table [4](#_bookmark36), the  *· *2 errors in the interval (0*,* 14) for each test function were examined. The table reflects the conclusion of the figures that the Euler method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *h* | *n* | ME | Euler | Talbot | Gaver |
| *e−t* | 3  5 | 0.0230  0.007423 | 0.1841  0.03915 | 0.0086  0.0003 | 0.0334  0.006858 |

Table 3

Errors for *h*(*t*)= *e−t* in  *· ∞*

occasionally outperforms the matrix exponential method for periodic functions, but this error measure of the matrix exponential method is also close to the best one in each case.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *h* | *n* | ME | Euler | Talbot | Gaver |
| *e−t* | 3  5 | 0.03909  0.01233 | 0.08237  0.02139 | 0.01206  0.0003876 | 0.03864  0.01064 |
| sin *t* | 13  24 | 3.248  1.611 | 3.345  0.2538 | 4.548  4.168 | 4.597  4.405 |
| 1(*t >* 1) | 13  24 | 0.102  0.06643 | 0.1012  0.07439 |  | 0.1957  0.1473 |
| 1(*t >* 1)*e*1*−t* | 13  24 | 0.1022  0.06647 | 0.1013  0.07442 |  | 0.197  0.1453 |
| *[t♩* | 13  24 | 0.3984  0.2914 | 0.356  0.295 | 0.5683  0.5131 | 0.582  0.5372 |
| square wave function | 13  24 | 1.24  0.8226 | 1.273  0.7228 | 1.702  1.569 | 1.719  1.604 |

Table 4

Errors in  *· *2 in the interval (0*,* 14)

# Conclusion

Laplace transform functions are often used in stochastic and probability models, where a numerical inverse transformation method with overshooting might result in negative values or values larger than 1. In this work we propose a numerical inverse Laplace transformation method without overshooting based on recent results on ma- trix exponential distributions, which on one hand falls into the Abate–Whitt frame- work and inherits some of its main properties (e.g. light computational complexity) and on the other hand reduces the error of the previously applied non-overshooting alternative, the “Erlang” method (cf. ([10](#_bookmark20)) and ([11](#_bookmark21))).

# References

1. Joseph Abate, Gagan L. Choudhury, and Ward Whitt. *An Introduction to Numerical Transform Inversion and Its Application to Probability Models*, pages 257–323. Springer US, Boston, MA, 2000.
2. S. Asmussen and M. Bladt. Renewal theory and queueing algorithms for matrix-exponential distributions. In *Matrix-analytic methods in stochastic models*, volume 183 of *Lecture Notes in Pure and Appl. Math.*, pages 313–341, 1997.
3. Soren Asmussen, Florin Avram, and Miguel Usabel. Erlangian approximations for finite-horizon ruin probabilities. *ASTIN Bulletin*, 32(2):267281, 2002.
4. Aldous David and Shepp Larry. The least variable phase type distribution is erlang. *Stochastic Models*, 3(3):467–473, 1987.
5. D. P Gaver. Observing stochastic processes and approximate transform inversion. *Oper. Res.*, 14:444459, 1966.
6. Ill´es Horv´ath, Orsolya S´af´ar, Mikl´os Telek, and Bence Z´amb´o. *Concentrated Matrix Exponential* *Distributions*, pages 18–31. Springer International Publishing, Cham, 2016.
7. Ill´es Horv´ath, Orsolya S´af´ar, Mikl´os Telek, and Bence Z´amb´o. List of coefficients for Matrix Exponential distributions with minimal coefficiant of variation. [http://webspn.hit.bme.hu/~telek/techrep/](http://webspn.hit.bme.hu/~telek/techrep/mecoefficients.zip) [mecoefficients.zip](http://webspn.hit.bme.hu/~telek/techrep/mecoefficients.zip), 2017. [Online; accessed 3-July-2017].
8. W. Whitt J. Abate. A unified framework for numerically inverting laplace transforms. *INFORMS* *Journal on Computing*, 18(4):408–421, Fall 2006.
9. V. Ramaswami, Douglas G. Woolford, and David A. Stanford. The erlangization method for markovian fluid flows. *Annals of Operations Research*, 160(1):215–225, Apr 2008.
10. H. Stehfest. Algorithm 368: Numerical inversion of laplace transforms [d5]. *Commun. ACM*, 13(1):47– 49, January 1970.
11. A. Talbot. The accurate numerical inversion of laplace transforms. *IMA Journal of Applied Mathematics*, 23(1):97, 1979.
12. J. Abate Valko, P. P. Comparison of sequence accelerators for the gaver method of numerical laplace transform inversion. *Comput. Math. Appl.*, 48:629636, March 2004.
13. Quanrong Wang and Hongbin Zhan. On different numerical inverse laplace methods for solute transport problems. *Advances in Water Resources*, 75:80 – 92, 2015.