Available online at www.sciencedirect.com
SCIENCE DIRECT
An International Journal
computers & mathematics with applications
ELSEVIER Computers and Mathematics with Applications 48 (2004) 629–636
www.elsevier.com/locate/camwa

Comparison of Sequence Accelerators for the Gaver Method of Numerical Laplace Transform Inversion

P. P. VALKÓ
Department of Petroleum Engineering
Texas A&M University
College Station, TX 77843, U.S.A.
p-valko@tamu.edu

J. ABATE
900 Hammond Road
Ridgewood, NJ 07450-2908, U.S.A.

(Received August 2002; revised and accepted October 2002)

Abstract—The sequence of Gaver functionals is useful in the numerical inversion of Laplace transforms. The convergence behavior of the sequence is logarithmic, therefore, an acceleration scheme is required. The accepted procedure utilizes Salzer summation, because in many cases the Gaver functionals have the asymptotic behavior $f_n(t) - f_{n-1}(t) \sim An^{-2}$ as $n \to \infty$ for fixed t. It seems that no other acceleration schemes have been investigated in this area. Surely, the popular nonlinear methods should be more effective. However, to our surprise, only one nonlinear method was superior to Salzer summation, namely the Wynn rho algorithm. © 2004 Elsevier Ltd. All rights reserved.

Keywords—Convergence acceleration, Sequence transformation, Laplace transform, Numerical transform inversion.

1. INTRODUCTION

The sequence of Gaver functionals is useful in the numerical inversion of Laplace transforms. The convergence behavior of the sequence is logarithmic, therefore, an acceleration scheme is required. The so-called Gaver-Stehfest method utilizes Salzer summation to accelerate convergence. The Salzer summation is one of the so-called linear acceleration methods and can be considered as the optimal method within that family, as explained in a recent review by Frolov and Kitaev [1]. The purpose of this paper is to examine the performance of some nonlinear sequence transformations applied to the Gaver functionals, namely,

- Wynn's rho algorithm
- Levin's u-transformation
- Lubkin's iterated w-transformation
- Brezinski's theta algorithm

0898-1221/04/$ - see front matter © 2004 Elsevier Ltd. All rights reserved. Typeset by $\mathcal{AMS}$-TEX
doi:10.1016/j.camwa.2002.10.017

In Section 2, we introduce the Gaver functionals and discuss their convergence behavior. In Section 3, we present the convergence acceleration methods. The results of the numerical examples are given in Section 4.

2. THE GAVER FUNCTIONALS

The problem of numerical inversion of the Laplace transform is to obtain approximations for $f(t)$ when numerical values of the transform function

$$\hat{f}(s) = \int_0^\infty e^{-st} f(t)\, dt \tag{1}$$

can be computed. There are many methods available to solve this problem. A comprehensive list of references is available on the web, see [2].

One of the most powerful and proven methods involves using the so-called Gaver functionals (see [3]), which are given by

$$f_k(t) = (-1)^k \tau k \binom{2k}{k} \Delta^k \hat{f}(k\tau) = k\tau \binom{2k}{k} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \hat{f}((k+j)\tau), \tag{2}$$

where $\tau = \ln(2)/t$ and Δ is the forward difference operator, i.e.,

$$\Delta \hat{f}(n\tau) = \hat{f}((n+1)\tau) - \hat{f}(n\tau).$$

Stehfest [4] suggested a reliable inversion algorithm based on (2). For a recent performance analysis of the algorithm the reader is referred to [5].

Under certain conditions the sequence of Gaver functionals converges logarithmically, that is

$$\lim_{k\to\infty} \frac{f(t) - f_{k+1}(t)}{f(t) - f_k(t)} = 1. \tag{3}$$

In fact, Gaver [3] showed that if $f(t)$ is representable by a Taylor series for all $t > 0$, $f_k(t)$ has the convergence behavior

$$f_k(t) \sim f(t) + \frac{c_1(t)}{k} + \frac{c_2(t)}{k^2} + \cdots \qquad \text{as } k \to \infty, \tag{4}$$

for fixed t.

To illustrate the asymptotic form (4), we consider some examples selected from Table 1. We compute the quantity $k(f(t) - f_k(t))$ for various values of k at fixed t. Table 2 displays these calculations and shows that condition (3) is satisfied for the examples.

Note that $F10$ and $F11$ in Table 1 are not bona fide transform pairs. That is, the forward transform (1) of $f(t) = 1/t$ does not exist. However, there is a sense in which the inverses of $F10$ and $F11$ are valid see [6, p. 62]. It is reasonable to expect that a good numerical inversion method should be able to handle these so-called pseudotransforms.

The Gaver functionals (2) can also be obtained by a recursive algorithm, as follows:

$$\begin{aligned} G_0^{(n)} &= n\tau \hat{f}(n\tau), && n \ge 1, \\ G_k^{(n)} &= \left(1 + \frac{n}{k}\right) G_{k-1}^{(n)} - \left(\frac{n}{k}\right) G_{k-1}^{(n+1)}, && k \ge 1, n \ge k, \\ f_k(t) &= G_k^{(k)}, \end{aligned} \tag{5}$$

see [3, p. 450].

Table 1. The test set of transforms and their inverses.

ID	$\hat{f}(s)$	$f(t)$
$F01$	$\frac{1}{1+s^2}$	$\sin t$
$F02$	$\frac{1}{\sqrt{s}+\sqrt{s+1}}$	$\frac{1-e^{-t}}{2\sqrt{\pi t^3}}$
$F03$	$\frac{1}{\sqrt{s(s+2)}}$	$e^{-t}I_0(t)$
$F04$	$\frac{1}{\sqrt{s}\,(1+\sqrt{s})}$	$e^t erfc\left(\sqrt{t}\right)$
$F05$	$\exp\left(-2\sqrt{s}\right)$	$\frac{e^{-1/t}}{\sqrt{\pi t^3}}$
$F06$	$\frac{\exp\left(-\sqrt{s}\right)\cos\sqrt{s}}{\sqrt{s}}$	$\frac{\cos(1/2t)}{\sqrt{\pi t}}$
$F07$	$\frac{\exp(-1/s)}{\sqrt{s^3}}$	$\frac{\sin 2\sqrt{t}}{\sqrt{\pi}}$
$F08$	$\frac{-\ln(s)}{s}$	$\ln(t)+\gamma$
$F09$	$\frac{e^s K_1(s)}{s}$	$\sqrt{t(t+2)}$
$F10$*	$\frac{1}{2}(\ln(s))^2$	$\frac{\ln(t)+\gamma}{t}$
$F11$*	$s^3\ln(s)$	$\frac{6}{t^4}$

*Pseudotransforms

Table 2. Estimate of $c_1(t)$ in (4) for $t=3$.

k	$F01$	$F03$	$F06$	$F08$	$F10$
4	0.653	0.0287	0.0610	−0.541	0.188
8	1.55	0.0267	0.0589	−0.551	0.185
16	2.49	0.0256	0.0580	−0.557	0.184
32	3.14	0.0251	0.0577	−0.559	0.183
64	3.50	0.0249	0.0575	−0.560	0.183
128	3.68	0.0248	0.0575	−0.561	0.183
256	3.78	0.0247	0.0574	−0.561	0.183

3. SEQUENCE TRANSFORMATIONS

To examine the acceleration of convergence for the Gaver functionals, we consider five sequence transformations. A study of the literature on methods, which accelerate logarithmically convergent sequences (see [7–15]) indicates that the five methods chosen are considered the best available for sequences that exhibit the asymptotic behavior given by (4).

The book by Wimp [7] provides a nice introduction to all the methods considered herein. The early comparison studies of Smith and Ford [8–10] were favorably disposed to Levin's u-transformation and Brezinski's theta algorithm. On the other hand, Weniger [11–13] found Lubkin's iterated w-transformation to be effective. Van Tuyl [14] and Osada [15] determined that Wynn's rho algorithm also works effectively.

Note, there seems to be no universal "best" method for logarithmically convergent sequences, see [16,17].

3.1. Salzer Summation and the Neville Table

Salzer summation is a linear method. The approximant for $f(t)$ is given by

$$f(t, M) = \sum_{k=1}^{M} W_k f_k(t), \tag{6}$$

where the weights are

$$W_k = (-1)^{k+M} \frac{k^M}{M!} \binom{M}{k}, \tag{7}$$

see [7, pp. 35–38]. Approximant (6) is known as the Gaver-Stehfest method [4] for numerical inversion of Laplace transform. The acceleration method itself may be viewed as a special case of the Richardson extrapolation process, see [4]. As such, it has been widely known prior to 1955 when Salzer presented his weights (7). It was previously given by a recursive scheme, known as the Neville table and also referred to as the Neville-Aitken extrapolation process. An equivalent recursive scheme is given by

$$\begin{aligned} T_0^{(n)} &= f_n(t), && n \geq 0, \\ T_k^{(n)} &= \left(1 + \frac{n}{k}\right) T_{k-1}^{(n+1)} - \left(\frac{n}{k}\right) T_{k-1}^{(n)}, && k \geq 1, \end{aligned} \tag{8}$$

see [7, p. 75]. Then, the approximant $f(t, M) = T_M^{(0)}$.

3.2. Wynn's Rho Algorithm

The Wynn rho algorithm is given by

$$\begin{aligned} \rho_{-1}^{(n)} = 0, \quad \rho_0^{(n)} &= f_n(t), && n \geq 0, \\ \rho_k^{(n)} &= \rho_{k-2}^{(n+1)} + \frac{k}{\rho_{k-1}^{(n+1)} - \rho_{k-1}^{(n)}}, && k \geq 1, \end{aligned} \tag{9}$$

see [7, p. 168]. The approximant $f(t, 2m) = \rho_{2m}^{(0)}$.

3.3. Levin's u-Transformation

Levin's u-transformation is given by

$$f(t, M) = \frac{\sum\limits_{k=1}^{M} W_k f_k(t)}{\sum\limits_{k=1}^{M} W_k}, \tag{10}$$

where the weights are

$$W_k = (-1)^k k^{M-2} \binom{M}{k} \frac{1}{f_k(t) - f_{k-1}(t)}, \tag{11}$$

see [8, p. 227] and also [7, p. 193]. Also, there is a recursive scheme for the Levin u-transformation, see [10,11,18]. Note that the Levin u-transformation has the same structure as the Salzer summation. Indeed, if in (11) we let $f_k(t) - f_{k-1}(t) = 1/k^2$ then (10) is equivalent to (6) and (7).

3.4. Lubkin's Iterated W-Transformation

The Lubkin iterated w-transformation is given by

$$\begin{aligned} W_0^{(n)} &= f_n(t), && n \geq 0, \\ W_{k+1}^{(n)} &= W_k^{(n+1)} - \frac{\left(\Delta W_k^n\right)\left(\Delta W_k^{n+1}\right)\left(\Delta^2 W_k^{n+1}\right)}{\left(\Delta W_k^{n+2}\right)\left(\Delta^2 W_k^n\right) - \left(\Delta W_k^n\right)\left(\Delta^2 W_k^{n+1}\right)}, && k \geq 0, \end{aligned} \tag{12}$$

where Δ is the forward difference operator acting on the superscript n; see [7, p. 152; 14, p. 231]. This method is also called the Brezinski iterated theta algorithm, see [13, p. 342]. Then, the approximant $f(t, 3m) = W_m^{(0)}$.

3.5. Brezinski's Theta Algorithm

The Brezinski theta algorithm is given by

$$\begin{aligned} \theta_{-1}^{(n)} &= 0, \quad \theta_0^{(n)} = f_n(t), && n \geq 0, \\ \theta_{2k+1}^{(n)} &= \theta_{2k-1}^{(n+1)} + \frac{1}{\Delta\theta_{2k}^{(n)}}, && k \geq 0, \\ \theta_{2k+2}^{(n)} &= \frac{\theta_{2k}^{(n+2)}\Delta\theta_{2k+1}^{(n+1)} - \theta_{2k}^{(n+1)}\Delta\theta_{2k+1}^{(n)}}{\Delta^2\theta_{2k+1}^{(n)}}, && k \geq 0 \end{aligned} \tag{13}$$

see [7, p. 171]. The approximant $f(t, 2m) = \theta_{2m}^{(0)}$. Note that $\theta_2^{(n)} = W_1^{(n+1)}$, which is why the Lubkin scheme (12) is also called the iterated theta algorithm, see [7, p. 171].

4. RESULTS

We use multiprecision computing provided by *Mathematica*. Table 3 shows the accuracy (number of significant digits) in the result of the approximant $f(t, M)$ at $t = 0.3$ with $M = 24$. To create the table, we used 50 decimal digits of precision.

Table 3. Number of significant digits obtained in acceleration of convergence for each method at $t = 0.3$, $M = 24$, and precision = 50.

	Salzer	Wynn Rho	Levin U	Lubkin W	Brezinski Theta
$F01$	22	23	20	13	14
$F02$	23	25	21	14	14
$F03$	24	25	20	11	11
$F04$	23	23	21	15	15
$F05$	11	12	9	4	4
$F06$	5	6	1	1	1
$F07$	23	24	11	9	7
$F08$	23	24	20	18	18
$F09$	23	25	20	11	13
$F10$	22	24	20	15	15
$F11$	7	24	6	8	8

Repeating the exercise for $t = 3, 30$, and 300, Tables 4–6 were created in a similar manner.

The results show remarkable consistency in the sense that the relative performance of any two methods remain basically the same, regardless which transform we look at. Though testing a limited set of transforms and time points always leaves some possibility for erroneous generalization, the results lead us to conclude, that for accelerating the convergence of Gaver functionals two methods perform significantly better than the others: the Salzer summation and Wynn's rho algorithm. In fact, we have more extensive computational experience to support this statement. We have tested these algorithms on the set of 105 transforms defined as Test Set A in [5]. The results of the extensive testing effort are similar to those given herewith in Tables 3–6. The transforms in Table 1 can be considered as a good representative sample of the larger set of 105 test pairs.

These two algorithms also have the property that with increasing M the number of correct decimals is increasing approximately linearly. To illustrate this point, we show results for $t = 300$ with doubled M (and precision) in Table 7. We note, however, that the Levin u algorithm, though always somewhat less effective than the Salzer summation, also exhibits linear increase of accuracy with the increase of M.

Note that in all the considered algorithms the overwhelming part of the computational effort is related to the multiprecision evaluation of the Gaver functionals. The slight difference in the

Table 4. Number of significant digits obtained in acceleration of convergence for each method at $t = 3$, $M = 24$, and precision = 50.

	Salzer	Wynn Rho	Levin U	Lubkin W	Brezinski Theta
$F01$	11	15	0	0	0
$F02$	21	24	4	5	5
$F03$	17	24	12	6	7
$F04$	22	24	19	13	15
$F05$	12	14	11	6	5
$F06$	11	15	10	10	10
$F07$	20	22	7	7	2
$F08$	23	25	21	19	18
$F09$	21	24	18	14	13
$F10$	23	25	21	17	16
$F11$	7	24	6	8	8

Table 5. Number of significant digits obtained in acceleration of convergence for each method at $t = 30$, $M = 24$, and precision = 50.

	Salzer	Wynn Rho	Levin U	Lubkin W	Brezinski Theta
$F01$	0	0	0	0	0
$F02$	12	14	10	8	8
$F03$	13	13	11	8	8
$F04$	20	22	19	14	10
$F05$	14	18	13	10	8
$F06$	15	19	14	11	14
$F07$	14	16	9	5	2
$F08$	23	25	21	19	19
$F09$	21	22	19	16	15
$F10$	23	25	21	14	14
$F11$	7	24	6	8	8

Table 6. Number of significant digits obtained in acceleration of convergence for each method at $t = 300$, $M = 24$, and precision = 50.

	Salzer	Wynn Rho	Levin U	Lubkin W	Brezinski Theta
$F01$	0	0	0	0	0
$F02$	13	16	12	10	10
$F03$	16	19	15	13	14
$F04$	21	22	19	16	17
$F05$	17	20	15	12	8
$F06$	20	22	18	16	16
$F07$	1	0	0	0	0
$F08$	24	25	21	19	19
$F09$	22	23	20	15	15
$F10$	23	25	22	16	15
$F11$	7	24	7	8	8

actual number of arithmetic operations during the sequence acceleration process is negligible. Therefore, the overall effectiveness of an inversion method based on the Gaver functionals would be primarily determined by the necessary number of terms (M) and by the required precision that is related to M.

In the following, we concentrate only on the two outstanding performers. The Salzer summation needs approximately $2M$ precision while the Wynn rho algorithm needs $2.1M$. Note

Table 7. Number of significant digits obtained in acceleration of convergence for each method at $t = 300$, $M = 48$, and precision = 50.

	Salzer	Wynn Rho	Levin U	Lubkin W	Brezinski Theta
$F01$	0	0	0	0	0
$F02$	24	29	23	13	13
$F03$	28	32	27	20	21
$F04$	40	44	38	20	20
$F05$	35	42	34	17	14
$F06$	38	43	37	23	19
$F07$	16	22	0	0	0
$F08$	46	49	44	20	22
$F09$	42	44	41	17	16
$F10$	45	48	44	20	23
$F11$	32	44	30	12	17

Table 8. Inversion of $F01$ at various t values using $M = 80 + t$.

t	M	Salzer	Wynn Rho
30	110	22	40
60	140	13	28
90	170	9	23
150	230	6	19
300	380	3	19
600	680	1	18
900	980	1	22
1500	1580	2	29
3000	3080	0	50

that these rules were obtained by detailed numerical experimentation. Since the small deviation in the required precision is not significant, the final outcome of the performance comparison is determined by the number of terms necessary to reach a certain accuracy goal. The difference in performance of the Salzer summation and the Wynn rho algorithm is illustrated here by a somewhat artificial but very informative numerical experiment involving the most demanding transform, $F01$: for increasing t values we increase the M value linearly and compare the convergence behavior. In particular, we use $M = 80 + t$ terms, while the precision of the calculations is increased accordingly. The results are shown in Table 8.

The sine function is of course notoriously difficult to obtain from its transform. Nevertheless, the Wynn rho algorithm shows remarkable consistency even in this case, indicating that to achieve a fixed accuracy of the inverse, the necessary M increases maximum linearly with t. Based on our experience, the Wynn rho algorithm outperforms the Salzer summation for accelerating the convergence of the Gaver functionals. So far we could not find any substantial counter example to this statement.

5. CONCLUSIONS

The results lead us to believe that for the convergence acceleration of the Gaver functionals the Wynn rho algorithm is the most effective among the acceleration schemes considered in this work. The reliable performance of the Wynn rho algorithm is predicted by a theorem by Osada [19] that ensures the convergence if condition (4) is satisfied. In our numerical investigation the Lubkin iterated transformation and the Brezinski theta algorithm performed worse than we had anticipated based on the views regarding their "across the board" properties. For example, see Section 5.2 of [19]. A reasonable explanation to our finding might have been that the number of arithmetic operations during the acceleration process is somewhat less in the Salzer and Wynn rho algorithms than in the other three. However, the accuracy indicated for a given M in Tables 2–6 is already "final" in the sense, that any further increase of the precision leaves the number of significant digits intact. In other words, the difference in performance has a deeper cause than the appearance and propagation of numerical error due to round-off and computer arithmetic.

Some of the results indicate that there may exist a nice error estimate for the Wynn rho algorithm, namely,

$$\left| \frac{f(t) - \rho_{2m}^{(0)}}{f(t)} \right| \approx 10^{-2m}. \tag{14}$$

Unfortunately, we are unable to determine a precise set of conditions when the error estimate is valid. This is an open question which we are pursuing.

REFERENCES

1. G.A. Frolov and M.Y. Kitaev, Improvement of accuracy in numerical methods for inverting Laplace transforms based on the Post-Widder formula, *Computers Math. Applic.* **36** (5), 23–34, (1998).
2. P.P. Valkó and B.L. Vojta, The list, `http://pumpjack.tamu.edu/~valko`, (2001).
3. D.P. Gaver, Jr., Observing stochastic processes and approximate transform inversion, *Oper. Res.* **14**, 444–459, (1966).
4. H. Stehfest, Algorithm 368: Numerical inversion of Laplace Transforms, *Comm. ACM* **13**, 47–49 and 624, (1970).
5. P.P. Valkó and S. Vajda, Inversion of noise-free Laplace transforms: Towards a standardized set of test problems, *Inverse Problems in Engineering* **10**, 467–483, (2002).
6. G. Doetsch, *Introduction to the Theory and Application of the Laplace Transformation*, Springer, New York, (1974).
7. J. Wimp, *Sequence Transformations and Their Applications*, Academic Press, New York, (1981).
8. D.A. Smith and W.F. Ford, Acceleration of linear and logarithmic convergence, *Comm. SIAM J. Numer. Anal.* **16**, 223–240, (1979).
9. D.A. Smith and W.F. Ford, Numerical comparison of nonlinear convergence accelerators, *Math. Comp.* **38**, 481–499, (1982).
10. T. Fessler, W.F. Ford and D.A. Smith, HURRY: An accelerator algorithm for scalar sequence and series, *ACM Trans. Math. Soft.* **9**, 346–354, (1983).
11. E.J. Weniger, Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series, *Comput. Phys. Rep.* **10**, 189–371, (1989).
12. E.J. Weniger, On the derivation of iterated sequence transformations for the acceleration of convergence and the summation of divergent series, *Comput. Phys. Comm.* **64**, 19–45, (1991).
13. E.J. Weniger, Prediction properties of Aitken's iterated Δ^2 process, of Wynn's epsilon algorithm, and of Bresinski's iterated theta algorithm, *J. Comput. Appl. Math.* **122**, 329–356, (2000).
14. A.H. Van Tuyl, Acceleration of convergence of a family of logarithmically convergent sequences, *Math. Comp.* **63**, 229–246, (1994).
15. N. Osada, A convergence acceleration method for some logarithmically convergent sequences, *SIAM J. Numer. Anal.* **19**, 178–189, (1990).
16. J.P. Delaheye and B. Germain-Bonne, The set of logarithmically convergent sequences cannot be accelerated, *SIAM J. Numer. Anal.* **19**, 840–844, (1982).
17. J. E. Drummond, Convergence speeding, convergence and summability, *J. Comput. Appl. Math.* **11**, 145–159, (1984).
18. H.H.H. Homeier, Scalar Levin-type sequence transformations, *J. Comput. Appl. Math.* **122**, 81–147, (2000).
19. N. Osada, An acceleration theorem for the ρ-algorithm, *Numer. Math.* **73**, 521–531, (1995).