Efficient computation of Sommerfeld integral tails – methods and algorithms

Krzysztof A. Michalski & Juan R. Mosig

Published online: 30 Mar 2016.

Submit your article to this journal

View related articles

View Crossmark data

Taylor & Francis
Taylor & Francis Group

INVITED REVIEW ARTICLE

Efficient computation of Sommerfeld integral tails – methods and algorithms

Krzysztof A. Michalski[a]*   and Juan R. Mosig[b]

[a]*Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA;* [b]*Laboratory of Electromagnetics and Acoustics, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

A review is presented of the most effective methods for the computation of Sommerfeld integral tails. Such integrals, which are often oscillatory, singular and divergent, commonly arise in layered media Green functions. The mathematical foundations of various pertinent methods are discussed in detail and their performance is illustrated by relevant numerical examples. The associated algorithms are given in pseudocode for their easy computer implementation.

Keywords: Sommerfeld integral; Green function; extrapolation; double-exponential rule

1. Introduction

In a variety of applications of new and well-established technologies a need arises to compute electromagnetic fields of time-harmonic dipoles embedded in plane-stratified media. In most cases these fields may be expressed in terms of the Sommerfeld integrals (also known as Fourier–Bessel or Hankel transforms [1, p.20],[2]) of the form

$$\int_0^\infty \tilde{G}(k_\rho; z|z') J_\nu(k_\rho \rho) k_\rho \, \mathrm{d}k_\rho, \quad \nu = 0, 1, 2, \tag{1}$$

where the kernel $\tilde{G}$ comprises the appropriate spectral-domain Green function,[3,4] z and z' are the vertical (perpendicular to the stratification) coordinates of the field- and source-point, respectively, ρ is the lateral (in the plane of the stratification) distance between these points, and J_ν denotes the Bessel function.[5, Section 10.2] $\tilde{G}$ is an even (odd) function of k_ρ for even (odd) ν. Note that the second-order transform can always be expressed in terms of the transforms of orders 0 an 1, using the recurrence relations for the Bessel functions.[5, Section 10.6] In certain loop antenna and aperture problems the driving current is not a point source, but is distributed over a circular region of radius τ, which leads to a more general Sommerfeld-type integral of the form [4,6–9]

$$\int_0^\infty \tilde{G}(k_\rho; z|z') J_\nu(k_\rho \rho) J_\sigma(k_\rho \tau) k_\rho \, \mathrm{d}k_\rho, \tag{2}$$

where the order σ may be integer or half-integer. Note that the (cylindrical) Bessel function of order $n + 1/2$ is related to spherical Bessel function of order n.[5, Section 10.47]

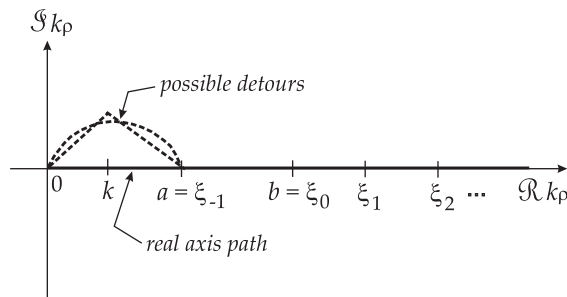*Corresponding author. Email: k-michalski@tamu.edu

Figure 1. Real axis integration path in the complex k_ρ-plane, with possible detours.

The integration path in (1)–(2) follows the positive-real axis of the complex k_ρ-plane, as illustrated in Figure 1. It is understood that this path should be indented to pass above any branch-point or pole singularities that may lie on or close to the positive-real axis, which may be accomplished by a path detour into the first quadrant, as also indicated in Figure 1.[9–15] Although the shape of this detour is not critical, its height is restricted by the exponential growth of the Bessel function off the real axis and the concomitant loss of accuracy when large numbers are subtracted to produce a small net result. Alternatively, one may first extract the poles [16] and keep the integration path on the real axis as

$$\left(\int_0^k + \int_k^a + \int_a^\infty \right) \tilde{G}(k_\rho; z|z') J_\nu(k_\rho \rho) k_\rho \, dk_\rho \equiv I_k + I_a + S, \qquad (3)$$

and similarly for (2), where k is the branch point associated with the upper halfspace and $a > k$ is a suitably selected break point – for example, we may set $a = 2k$. If the second branch point is also located on or near the real axis, the corresponding break point should be inserted between k and a. The finite-range integrals I_k and I_a in (1) may readily be evaluated by the Gauss–Jacobi quadrature,[17],[18, p.186] or by the *tanh-sinh* double-exponential (DE) rule, which suppresses endpoint singularities,[19–21] or by a suitable local change of the variable of integration on both sides of the singular point,[22–25] The computation of the *tail integral S* poses a much greater challenge, owing to the fact that $\tilde{G}$ typically behaves as [12]

$$\tilde{G}(k_\rho; z|z') \sim \frac{e^{-k_\rho \zeta}}{k_\rho^\delta} \left[C + \mathcal{O}(k_\rho^{-1}) \right], \quad k_\rho \to \infty, \qquad (4)$$

where C is a constant and $\zeta > 0$ is a parameter related to z and z' – for example, in a homogeneous medium $\zeta = |z - z'|$. In the above and elsewhere in this paper we utilize the "big-oh" and "little-oh" Landau symbols.[26, p.9] Hence, as long as $\rho > 0$, the integrand in (3) is oscillatory and – depending on the values of ζ and δ, may be slowly convergent or may even diverge. The oscillations are only absent in the $\rho = 0$ "zero offset" case. [27–29] This is why evaluating Sommerfeld integrals has been characterized as "a standard nightmare for many electromagnetic engineers", [30] and it should not be surprising that a multitude of papers have been published – and still continue to be published – devoted to this topic.[12,15,31–45]

The purpose of the present review is to present in detail the most effective techniques and algorithms for the computation of Sommerfeld integrals along the real axis, with emphasis on the Sommerfeld tails. The mathematical foundations of the various methods are discussed in detail, with references to the relevant literature, and their performance is

illustrated by representative numerical examples. In most cases algorithms are given in pseudocode for easy computer implementation. We exclude from consideration methods based on path deformation into the complex plane. [46,47] (applicable when $\rho > 0$), as well as the complex-image and similar approximate methods.[48–51] The advantage of using the real-axis integration path is that the exact location of the poles intercepted on the proper and improper Riemann sheets need not be determined and that only Bessel functions of real arguments arise in the integral tail. The method is generally applicable, but is not recommended when the lateral distance ρ exceeds about a hundred wavelengths, in which case the Bessel function undergoes an excessive number of oscillations in the initial interval $(0, a)$ of the path in Figure 1. Hence, when far-zone fields are of interest, other techniques – such as numerical integration along the steepest descent path or along vertical branch cuts [52–55] – are more appropriate.

The contents of the remainder of this paper is as follows. In Section 2, we describe the DE rules and their adaptive implementation. In Section 3, we discuss sequence acceleration by extrapolation and describe and compare the performance of the most effective algorithms. In Section 4, we discuss the partition-extrapolation method for the oscillatory Sommerfeld tails and describe its efficient implementation. This method relies on the DE rules and extrapolation techniques introduced in the previous sections. In Section 5, we describe another type of DE rule applicable to oscillatory tails. In Section 6, we include sample numerical results for Sommerfeld integral tails evaluated by different methods. Finally, we end with conclusions in Section 7. Throughout the paper, we include algorithms in pseudocode, which may easily be translated into any higher level computer language, such as modern Fortran.[56]

2. Double-exponential (DE) rules

In this section, we describe efficient methods for the numerical integration over a finite or semi-infinite range of analytic functions that may be singular at one or both of the endpoints, which arise in Sommerfeld integrals. The DE rules of Takahasi and Mori [19] presented here are competitive in terms of efficiency with Gaussian quadratures,[57, Section 8.4],[58, Chapter 3] but are more versatile – as they are quite insensitive to endpoint behavior – and simple to program, since the weights and nodes (abscissae) are easily generated. Furthermore, the DE rules can be implemented in a progressive manner, with the step size halved each time the rule level is raised and reusing the function values computed on previous levels. The integral values generated at consecutive levels can be used to monitor the convergence of the quadrature. As with any other quadrature rule, singularities of the integrand near the integration path adversely affect the convergence and should preferably be extracted prior to the integration. However, any singularities on the integration path are easily treated by splitting the integration range, so that the singularities are placed at the endpoints. The development of the DE rules was motivated by the discovery by Goodwin [59] that the primitive trapezoidal rule,

$$\int_{-\infty}^{\infty} f(\xi) \, d\xi \approx h \sum_{k=-\infty}^{\infty} f(kh), \tag{5}$$

where $h > 0$ is the mesh size, gives remarkably accurate results for certain functions that are analytic in a strip about the real axis. This paradox can be explained by the fact that bell-shaped functions $f(\xi)$ that fall off rapidly as $\xi \to \pm\infty$ are well approximated by the *sinc expansion*, the integral of which gives exactly the rule (5).[18, p.178] In fact, it can

be shown by contour integration in the complex plane that the *discretization error* in (5) is $\mathcal{O}\left(e^{-2\pi w/h}\right)$ as $h \to 0$, where w is the distance from the real axis to the nearest singularity of the integrand.[60] The convergence being exponential, halving the step h doubles the number of correct figures in the result for sufficiently small values of h.

2.1. Finite intervals

To apply this method to finite-range integrals, we map the infinite interval $(-\infty, \infty)$ onto $[-1, 1]$ using a suitable monotonic function $\varphi(t)$, such that

$$\int_{-1}^{1} f(\xi)\,d\xi = \int_{-\infty}^{\infty} f[\varphi(t)]\dot{\varphi}(t)\,dt \approx h \sum_{k=-n}^{n} \dot{\varphi}(kh) f[\varphi(kh)], \tag{6}$$

where $\dot{\varphi} \equiv d\varphi/dt$ is the Jacobian of the transformation. As indicated, the infinite sum must in practice be terminated after a finite number of terms, thus introducing a *truncation error*, in addition to the discretization error. The high accuracy of the rule (6) is predicted by the Euler-Maclaurin sum formula,[61, p.24], [62, p.247],[63] if the transformation results in a bell-shaped function whose derivatives all vanish at $\pm\infty$. This formula is basically a trapezoidal quadrature rule with an error term in a form of an asymptotic series, whose terms comprise the derivatives of the integrand evaluated at the endpoints of the interval of integration. Hence, a transformation that forces these endpoint derivatives to vanish results in a great enhancement of the accuracy of the trapezoidal rule. The required mapping may be effected by choosing [20]

$$\varphi(t) = \tanh g(t), \quad \dot{\varphi}(t) = \dot{g}(t)\,\text{sech}^2 g(t), \tag{7}$$

where $g(t)$ is an odd function, which leads to the trapezoidal rule approximation

$$\int_{-1}^{1} f(\xi)\,d\xi \approx h \sum_{k=-n}^{n} w_k f(\xi_k), \tag{8}$$

with the nodes ξ_k and weights w_k defined as

$$\xi_k = 1 - \delta_k, \quad w_k = 2\dot{g}(kh)\delta_k(1 + q_k)^{-1}, \tag{9}$$

where we have introduced

$$\delta_k = 2q_k(1 + q_k)^{-1}, \quad q_k = e^{-2g(kh)}. \tag{10}$$

It is important to code the abscissae and weights as indicated in (9)–(10), since a more direct implementation of (7) would result in overflows. Using the fact that $q_{-k} = q_k^{-1}$, $\xi_{-k} = -\xi_k$, and $w_{-k} = w_k$, we may also express (8) in the more convenient form

$$\int_{-1}^{1} f(\xi)\,d\xi \approx h\left\{\dot{g}(0)f(0) + \sum_{k=1}^{n} w_k\big[f(-1+\delta_k) + f(+1-\delta_k)\big]\right\}. \tag{11}$$

Although we assume in the above the standard integration interval $[-1, 1]$, an arbitrary interval $[a, b]$ may be mapped onto $[-1, 1]$ by the linear transformation $\xi = \sigma x + \gamma$ with $\sigma = (b-a)/2$ and $\gamma = (a+b)/2$, which leads to [57, p.367]

$$\int_{a}^{b} f(\xi)\,d\xi = \sigma \int_{-1}^{1} f(\sigma x + \gamma)\,dx. \tag{12}$$

Hence, for an arbitrary interval $[a, b]$ the nodes and weights become $\sigma \xi_k + \gamma$ and σw_k, respectively, and (11) is transformed to

$$\int_a^b f(\xi)\,\mathrm{d}\xi \approx \sigma h \left\{ \dot{g}(0) f(\gamma) + \sum_{k=1}^n w_k \big[f(a+\sigma\delta_k) + f(b-\sigma\delta_k) \big] \right\}. \tag{13}$$

We have not yet specified the function $g(t)$ appearing in the argument of the tanh transformation (7). Perhaps the simplest odd function choice,

$$g(t) = t^\nu, \quad \dot{g}(t) = \nu t^{\nu-1}, \quad \nu = 1, 3, 5, \dots \tag{14}$$

generates the νth order *tanh rule*,[20] which for $\nu = 1$ reduces to the classical *tanh rule*, [64–67], whereas choosing

$$g(t) = \eta \sinh t, \quad \dot{g}(t) = \eta \cosh t, \tag{15}$$

where η is a positive parameter, generates the *tanh-sinh rule*.[19] The Jacobians $\dot{\varphi}(t)$ of the tanh and tanh-sinh transformations exhibit exponential and double-exponential decay, respectively, as $t \to \pm\infty$, resulting in the desired rapid fall-off of the integrands in most cases. These Jacobians also introduce singularities in the complex plane, so that the half-width of the strip of analyticity $w = \pi/2$ for the tanh rule, as well as the tanh-sinh rule, provided that $0 < \eta \le \pi/2$. The performance of the tanh-sinh rule tends to be insensitive to values of η in the neighborhood of unity, hence the convenient value of $\eta = 1$ has been suggested as a good practical choice.[20] Setting $\eta > \pi/2$ is counterproductive, as this causes w to decrease rapidly, with the concomitant deterioration of the convergence rate. With $w = \pi/2$, the discretization error of these rules is $\mathcal{O}(e^{-\pi^2/h})$, whereas the truncation error for a given truncation index n may be estimated as $\mathcal{O}(e^{-2g(nh)})$.[20] For a fixed n, the truncation error grows with diminishing h, while the discretization error decreases. One can therefore estimate the optimum value of h by making the two errors equal, which leads to $h \approx \pi/\sqrt{2n}$ and $h \approx \log(\pi^2 n)/n$ for the (classical) tanh and tanh-sinh rules, respectively, assuming $\eta = 1$ in the latter, and the corresponding total errors are found to be $\mathcal{O}(e^{-\pi\sqrt{2n}})$ and $\mathcal{O}(e^{-\pi^2 n/\log(\pi^2 n)})$, respectively.[20] The estimates for the higher order tanh rule are more elaborate and are also given in,[20] while a more refined step size estimate for the tanh-sinh rule may be found in [63]. Although these are only crude estimates based on the assumption that the integrand function itself makes no appreciable contribution to the decay rate, they are nevertheless indicative of the performance of the two methods. In particular, we note that with a growing truncation index n the double-exponential (DE) tanh-sinh rule converges faster that the single-exponential (SE) tanh rule. One might thus expect that a *triple*-exponential rule would offer further improvement. It turns out, however, that beyond the double-exponential, the poles of the Jacobian come nearer to the real axis, so that the discretization error gets worse, not better [18, p.176]; in short, the DE rule appears to be optimal.[19,68,69] We should mention that, from the classical point of view, the rule (13) is a strange quadrature method indeed, as it is not exact for any polynomials – not even a constant – for any h and finite n.[67] Nevertheless, it is remarkably accurate!

A salient feature of the tanh and tanh-sinh transformations is that the uniform sampling on the t-axis results in the clustering of nodes ξ_k near the endpoints of the integration interval, which – in view of the rapid decrease of the weights w_k – permits the suppression of any endpoint singularities of $f(\xi)$. This node clustering is also responsible for a pitfall in the computer implementation of (13), which may have been responsible for the initially slow rate of acceptance of the DE rules.[69] Namely, since δ_k can become very small, the round off errors in computing $a + \sigma\delta_k$ and $b - \sigma\delta_k$ may give rise to overflows as the singularities

at a and b are approached. To remedy this, the function $f(\xi)$ should be programmed as a two-parameter procedure $f(c,d)$, where $\xi = c+d$, $c = a$ or b, and $d = \pm\sigma\delta_k$, so that the singular part may be computed using d directly.[20,68], [18, p.175] For example, if the integrand comprises a term $(\xi-a)^{-\alpha}(b-\xi)^{-\beta}$ with $\alpha>0$ and $\beta>0$, this procedure should compute it as $d^{-\alpha}(b-\xi)^{-\beta}$ when $d>0$, and as $(\xi-a)^{-\alpha}(-d)^{-\beta}$ when $d<0$.

In view of the indeterminate nature of the error estimates, reliance cannot be placed on a calculation of (13) for a *single* value of the truncation index n, and if the optimum values of the mesh size h were employed, the earlier function evaluations would usually be lost when n is increased. Hence, here we adopt the following strategy, due to Evans et al.[20] First, with the selected starting (zero-level) mesh size h, the initial truncation index n is determined from the condition

$$|T_n| < \kappa\left|T_0 + \sum_{k=1}^{n} T_k\right|, \quad n = 1, 2, \ldots \tag{16}$$

where T_k denotes the kth term in the rule (13) and κ is a specified truncation error. Note that this empirical determination of the truncation index takes into account any endpoint singularities of the integrand function. In the process of computing n, the zero-level quadrature estimate is also obtained. Next, higher level estimates are computed with the number of points systematically doubled and the mesh size halved each time the level is raised, always re-utilizing in (13) the previously computed function values. The successive quadrature estimates are monitored until convergence to a specified tolerance ϵ is achieved. Since the number of correct digits approximately doubles with each iteration once the method starts converging, the *actual* error of the computed integral will typically be the square of the requested ϵ value.[70], [18, p.178] Accordingly, one can set ϵ to the square root of the desired precision. A posteriori error bounds for DE rules are also available, but require the computation of two derivatives of the integrand function at the sampling points. [71,72]

A pseudocode implementing an automatic and progressive tanh-sinh rule is given in Algorithm 1, which comprises the main procedure TANHSINHQUAD and two subsidiary procedures TRUNCINDEX and PARTSUM, as well as an auxiliary function TERM. This algorithm closely follows the steps described above and is for the most part self-explanatory. The procedure TRUNCINDEX computes the initial truncation index n and the initial quadrature estimate using the condition (16), and the procedure PARTSUM evaluates higher level partial sums, skipping the nodes of all previous levels. Finally, the function TERM computes the kth term of the rsule (13). Note that the transformation (15) is implemented as

$$2g(kh) = \eta\left(e^{kh} - 1/e^{kh}\right), \quad 2\dot{g}(kh) = \eta\left(e^{kh} + 1/e^{kh}\right), \tag{17}$$

with $\dot{g}(0) = \eta$. After some experimentation, the initial mesh size was set to $h = 1.5$ and the truncation threshold in (16) to $\kappa = 10^{-15}$. The user-provided integrand function f should be coded as a two-parameter procedure with overflow mitigation in the case of endpoint singularities, as discussed above. Implementations of the tanh-sinh rule are also given in [73, p.312], [74,75], and [18, p.176]. The tanh rule can similarly be implemented, but the algorithm is not listed in the interest of brevity.

We now illustrate the application of the tanh-sinh rule to the integral I_k in (3), where the integrand exhibits an inverse-square-root singularity at the upper interval end, which may be set to unity without loss of generality, since the integration variable can be normalized

---

Algorithm 1 Automatic tanh-sinh quadrature

---

    procedure TANHSINHQUAD(f, a, b, ϵ, val)   ▷ f is the integrand function
        parameter $\eta \leftarrow 1, \kappa \leftarrow 1E{-}15, nmax \leftarrow 24, maxlev \leftarrow 5$
        $\sigma \leftarrow (b{-}a)/2; \gamma \leftarrow (b{+}a)/2$
        $m \leftarrow 0$
        $h \leftarrow 1.5$ ▷ Initial mesh size
        $eh \leftarrow exp(h)$
        $s \leftarrow \eta * f(\gamma, 0)$
        TRUNCINDEX(eh, n, s)
        $old \leftarrow \sigma * h * s$ ▷ Level-0 estimate
        repeat   ▷ Add higher-level sums with refined mesh
            $m \leftarrow m + 1$
            $e2h \leftarrow eh$
            $h \leftarrow h/2$
            $eh \leftarrow exp(h)$
            PARTSUM($eh, e2h, n, s$)
            $val \leftarrow old/2 + \sigma * h * s$ ▷ Level$-m$ estimate
            if $abs(val{-}old) < \epsilon * abs(val)$ **exit**   ▷ Convergence check
            $old \leftarrow val ; n \leftarrow 2 * n$
        until $m = maxlev$
        return val ▷ Return integral estimate
    procedure TRUNCINDEX(eh, n, s)   ▷ Initial truncation index & partial sum
        $ekh \leftarrow eh ; n \leftarrow 0$
        repeat
            $n \leftarrow n + 1$
            $t \leftarrow$ TERM(ekh) $; s \leftarrow s + t$
            if $abs(t) \leq \kappa * abs(s)$ **exit**
            $ekh \leftarrow ekh * eh$
        until $n = nmax$
        return n, s
    procedure PARTSUM($eh, e2h, n, s$)   ▷ Higher-level partial sum
        $ekh \leftarrow eh$
        $s \leftarrow$ TERM(ekh)
        for $k \leftarrow 2, n$ **do**
            $ekh \leftarrow ekh * e2h$ ▷ Skipping previous levels nodes
            $s \leftarrow s +$ TERM(ekh)
        return s
    function TERM(ekh)   ▷ kth term of quadrature rule
        $q \leftarrow exp(-\eta * (ekh{-}1/ekh))$
        $\delta \leftarrow 2 * q/(1{+}q)$
        $w \leftarrow \eta * (ekh{+}1/ekh) * \delta/(1{+}q)$
        $t \leftarrow w * (f(a, \sigma * \delta) + f(b, -\sigma * \delta))$
        return t

Table 1. DE tanh-sinh rule applied to $I_1(\pi/2)$.

m	N	tanh-sinh rule	Gauss–Jacobi rule
0	7	0.649087553439706	0.636619772370296
1	13	0.636587540227121	0.636619772367582
2	25	0.636619773340205	0.636619772367581
3	49	0.636619772367581	0.636619772367581

Table 2. Classical tanh rule applied to $I_1(\pi/2)$.

m	N	tanh rule ($\nu = 5$)
0	49	0.657352039535013
1	97	0.636776444654924
2	193	0.636619775154870
3	385	0.636619772367581

to the wavenumber k. Hence, we consider the integral

$$I_1(\rho) = \int_0^1 J_0(\xi\rho) \frac{\xi \, d\xi}{\sqrt{1-\xi^2}} = \frac{\sin\rho}{\rho}, \tag{18}$$

where the integrand is singular at $\xi = 1$ and also oscillatory with the frequency proportional to the parameter ρ. Applying Algorithm 1 to compute $I_1(\pi/2)$, we obtain the results listed in Table 1, where m denotes the rule level and N the cumulative number of function evaluations. A rapid convergence is observed, with the number of correct digits, which are underlined, approximately doubled each time the rule level is raised. In the last column, we include the results of the Gauss–Jacobi quadrature.[17] Using the same number of nodes as in the corresponding tanh-sinh rule. The Gauss–Jacobi quadrature delivers unsurpassed accuracy, but unlike the tanh-sinh rule, is non-progressive, i.e. it cannot reuse the previously evaluated functions when the quadrature level is raised to check for convergence. For the same integral, in Tables 2 and 3, we include results obtained by the classical and fifth-order tanh rules, respectively. It is seen that although the fifth-order tanh rule requires much fewer function evaluations than the classical one, it is still inferior to the DE tanh-sinh rule.

Next, to examine the performance of the tanh-sinh rule in the case of highly oscillatory integrands, we evaluate I_1 with the parameter ρ increased over fiftyfold. The results for this case are listed in Table 4, where rapid convergence is observed, but only beginning with the fourth level, after a number of nodes has been reached, sufficient to capture the oscillations of $J_0(\xi\rho)$. In the last column, we again include for comparison the corresponding Gauss–Jacobi quadrature results.

To gain further insight into the performance of the tanh-sinh rule, in Figure 2(a) we plot the original integrand function $f(\xi)$ of $I_1(53\pi/2)$, as given by (18), and the transformed integrand $F(t) = f[\varphi(t)]\dot{\varphi}(t)$.

Note the singular inverse-square-root behavior of $f(\xi)$ near $\xi = 1$, modulated by the oscillations of the Bessel function. In the transformed integrand, this singularity has been moved to $t = \infty$ and suppressed by the double-exponential decay of the Jacobian $\dot{\varphi}(t)$.

Table 3. 5th-order tanh rule applied to $I_1(\frac{\pi}{2})$.

m	N	tanh rule ($\nu = 5$)
0	4	0.012763231686033
1	8	0.560903906568641
2	16	0.629592597929682
3	32	0.636863079947029
4	64	0.636619753669647
5	128	0.636619772367581

Table 4. DE tanh-sinh rule applied to $I_1(53\pi/2)$.

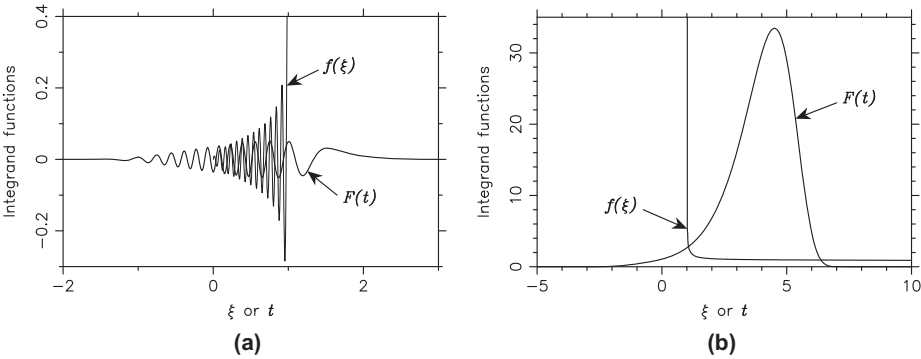| m	N	tanh-sinh rule	Gauss–Jacobi rule
0	7	$-0.547800669816593E-2$	$0.326373277124372E-1$
1	13	$0.478732669804770E-1$	$0.288200040625463E-1$
2	25	$0.430154765748145E-1$	$0.115952530174854E-1$
3	49	$0.378578537982330E-1$	$0.120116938182562E-1$
4	97	$0.120116938505470E-1$	$0.120116938182567E-1$
5	193	$0.120116938182563E-1$	$0.120116938182564E-1$

Figure 2. Plots of the original integrand $f(\xi)$ and the transformed integrand $F(t) = f[\varphi(t)]\dot{\varphi}(t)$ for (a) $I_1(53\pi/2)$ using the tanh-sinh transformation and (b) $I_2(0.011)$ using the mixed transformation.

Although still oscillatory, $F(t)$ is well behaved with a rapid decay at both ends, which explains the high accuracy of the trapezoidal rule.

2.2. Semi-infinite intervals

For a semi-infinite interval $[a, \infty)$, the appropriate DE transformation is [19]

$$\xi \leftarrow \varphi(t) = a + e^{2\eta \sinh t}, \quad \dot{\varphi}(t) = 2\eta \cosh t \, e^{2\eta \sinh t}, \tag{19}$$

which yields the quadrature rule

$$\int_a^\infty f(\xi)\,\mathrm{d}\xi \approx h \sum_{k=-n_1}^{n_2} w_k f(a+\delta_k)\,, \tag{20}$$

with the abscissas and weights given as

$$\delta_k = e^{\eta(e^{kh}-e^{-kh})}, \quad w_k = \eta\big(e^{kh}+e^{-kh}\big)\delta_k. \tag{21}$$

However, this transformation is not optimal if $f(\xi)$ exhibits an exponential decay at infinity, in which case a more appropriate choice is [19]

$$\xi \leftarrow \varphi(t) = a + e^{t-e^{-t}}, \quad \dot\varphi(t) = \big(1+e^{-t}\big)e^{t-e^{-t}}. \tag{22}$$

Note that this is a *mixed* DE–SE transformation – DE at $\xi = a$, but only SE for $\xi \to \infty$.[18, p.176] However, the exponential decay of $f(\xi)$ at infinity results in a DE behavior at the upper integration limit also. For the mixed transformation (22), the abscissae and weights in (20) become

$$\delta_k = e^{kh-e^{-kh}}, \quad w_k = \big(1+e^{-kh}\big)\delta_k. \tag{23}$$

Note that, due to the lack of symmetry in the transformations (19) and (22), the sum in (20) cannot be folded with respect to the $k=0$ element – as in (13), and the partial sums for negative and positive k must be evaluated separately.

A pseudocode implementing an automatic and progressive mixed rule is given in Algorithm 2, which comprises the main procedure MIXEDQUAD and two subsidiary procedures TRUNCINDEX and PARTSUM, as well as an auxiliary function TERM. This algorithm is very similar to Algorithm 1 – in fact, it shares the procedures TRUNCINDEX and PARTSUM with the latter. The function TERM is of course different, as it implements the kth term in (20), with the abscissae and weights given by (23). The other difference is that the partial sums for negative and positive indices k must be treated separately – hence the two calls to TRUNCINDEX and, at each level, two calls to PARTSUM. Note that changing $-k$ to k is tantamount to the replacement $e^{kh} \to 1/e^{kh}$. The initial truncation indices n_1 and n_2 for the negative and positive ranges of k are determined based on a condition similar to (16). As in the case of the tanh-sinh rule, the user-provided function $f(\xi)$ should be coded as a two-parameter procedure $f(a,\delta)$, where $\xi = a+\delta$, to facilitate overflow mitigation for integrands that have an algebraic singularity at the lower end a of the integration range.

As an example of the application of the mixed rule, consider the integral

$$I_2(z) = \int_1^\infty \frac{e^{-\sqrt{\xi^2-1}\,z}}{\sqrt{\xi^2-1}}\,\xi\,\mathrm{d}\xi = \frac{1}{z}, \quad z > 0, \tag{24}$$

where the integrand is singular at the lower limit and exhibits exponential decay at infinity. Applying Algorithms 2 to compute I_2 for $z=0.011$, we obtain the results listed in Table 5. Here again, once the method starts converging, approximate doubling of the number of correct digits is observed each time the rule level is raised. In this example, we specified the small value of z in order to make the integrand slowly-convergent at infinity and thus the problem more challenging. The summation limits in (20) (as determined by TRUNCINDEX) were in this case $n_1=5$ and $n_2=9$. It is found that increasing z ten-fold to $z=0.11$ reduces n_2 to 6 and the number of function evaluations from 57 to 45.

To get more insight into the performance of the mixed rule, in Figure 2(b) we plot the original integrand function $f(\xi)$ of $I_2(0.011)$, as given by (24), and the transformed

Algorithm 2 Automatic mixed quadrature rule

procedure MIXEDQUAD(f, a, ϵ, val)
    parameter $\kappa \leftarrow 1E-15$, $nmax \leftarrow 24$, $maxlev \leftarrow 5$
    $m \leftarrow 0$
    $h \leftarrow 1$ $\triangleright$ Initial mesh size
    $eh \leftarrow exp(h)$
    $\delta \leftarrow exp(-1)$; $w \leftarrow 2 * \delta$
    $s \leftarrow w * f(a, \delta)$ $\triangleright$ $k = 0$ term
    TRUNCINDEX($eh, n1, s$)
    TRUNCINDEX($1/eh, n2, s$)
    $old \leftarrow h * s$ $\triangleright$ Level-0 estimate
    repeat $\triangleright$ Add higher-level sums with refined mesh
        $m \leftarrow m + 1$
        $e2h \leftarrow eh$
        $h \leftarrow h/2$
        $eh \leftarrow exp(h)$
        PARTSUM($eh, e2h, n1, s1$)
        PARTSUM($1/eh, 1/e2h, n2, s2$)
        $val \leftarrow old/2 + h * (s1+s2)$ $\triangleright$ Level$-m$ estimate
        if $abs(val-old) < \epsilon * abs(val)$ **exit** $\triangleright$ Convergence check
        $old \leftarrow val$
        $n1 \leftarrow 2 * n1$; $n2 \leftarrow 2 * n2$
    until $m = maxlev$
    return val $\triangleright$ Return integral estimate
function TERM(ekh) $\triangleright$ kth term of quadrature rule
    $\delta \leftarrow exp(-ekh)/ekh$
    $w \leftarrow (1 + ekh) * \delta$
    $t \leftarrow w * f(a, \delta)$
    return t

Table 5. Mixed rule applied to $I_2(0.011)$.

m	N		Mixed rule
0	15		$0.908507679409979E+2$
1	29		$0.909090920844590E+2$
2	57		$0.909090909090909E+2$

integrand $F(t) = f[\varphi(t)]\dot{\varphi}(t)$. Note the inverse-square-root singularity of $f(\xi)$ at $\xi = 1$ and its very slow exponential decay for large ξ. In contrast, the transformed integrand $F(t)$ is a smooth hat-like shaped function, with rapid decay at both ends. The singularity of the original integrand at the lower integral limit has been sent to $t = -\infty$ and suppressed by the double-exponential decay of the Jacobian, and its exceedingly slow decay for $\xi \to \infty$ has also been spectacularly accelerated, which explains the remarkable accuracy of the trapezoidal rule in this case.

The DE rules for semi-infinite ranges discussed above are not recommended in the case of slowly decaying *and* oscillating functions.[76] For such integrands, a special DE rule has been developed based on a different principle, which we describe in Section 5.

3. Sequence acceleration by extrapolation

Extrapolation is a powerful tool frequently employed in the efficient computation of Sommerfeld integral tails.[12] The objective of extrapolation is to accelerate the rate of convergence of a series

$$S = \sum_{i=0}^{\infty} u_i, \tag{25}$$

or, equivalently, of a sequence of partial sums

$$S_n = \sum_{i=0}^{n} u_i, \quad n \to \infty. \tag{26}$$

The sequence element S_n may be partitioned as

$$S_n = S + r_n, \tag{27}$$

where r_n is the truncation error, commonly referred to as the *remainder*. To accelerate the convergence of (26), we seek a transformation $\{S_n\} \to \{S'_n\}$, such that the new sequence converges faster, i.e.

$$\frac{|r'_n|}{|r_n|} = \mathcal{O}(x_n^{-\mu}), \quad \mu > 0, \quad n \to \infty, \tag{28}$$

where $\{r'_n\}$ are the remainders of $\{S'_n\}$. Such a transformation may be viewed as an extrapolation method, since the underlying principle is to obtain better estimates from a sequence of computed values S_n.[77] The convergence of the sequence of partial sums (26) may be characterized based on the behavior of the limits

$$\lambda = \lim_{n\to\infty} \frac{r_{n+1}}{r_n} = \lim_{n\to\infty} \frac{u_{n+1}}{u_n}, \tag{29}$$

where the second equality applies with some rare exceptions.[78] If $|\lambda| < 1$, the convergence is *linear*, if $\lambda = 1$, it is *logarithmic*, if $\lambda = 0$, it is *hyperlinear*, and if $|\lambda| > 1$, the sequence diverges. We say that the sequence is asymptotically monotone if $\lambda > 0$, and alternating if $\lambda < 0$. Alternating sequences are generally the easiest to accelerate, the logarithmic convergence being the most challenging.

The most general extrapolation procedure is based on the assumption that the remainder estimates possess the asymptotic expansion

$$r_n \sim \sum_{j=0}^{\infty} c_j \psi_j(n), \quad n \to \infty, \tag{30}$$

where c_j are unknown coefficients and $\psi_j(n)$ are known, but otherwise arbitrary functions of n, with the property that $\psi_{j+1} = o(\psi_j(n))$. To accelerate the convergence of the sequence (26), a transformation is constructed that eliminates a number of leading coefficients c_j from r'_n, with the number of coefficients annihilated being the order of the transformation. For many sequences encountered in practice the remainders possess a Poincaré-type asymptotic expansion [78]

$$r_n \sim \omega_n \sum_{j=0}^{\infty} c_j x_n^{-j}, \quad n \to \infty, \tag{31}$$

where ω_n are the *remainder estimates*, which provide the structural information about the behavior of r_n for large n, and $\{x_n\}$ is an *auxiliary sequence* of interpolation points

(or nodes), such that $x_{n+1} > x_n$. Note that (31) is a specialized form of (30), with the basis functions $\psi_j(n) = \omega_n x_n^{-j}$. The most powerful convergence accelerators exploit the information contained in the remainder estimates, which are obtained either numerically from the elements of the sequence, or derived analytically, based on their known asymptotic behavior. The choice of the auxiliary sequence $\{x_n\}$ also affects the properties of the sequence transformation. If equispaced nodes are appropriate, we may use

$$x_n = \beta + n, \quad \beta > 0, \tag{32}$$

where n is a natural number and the *shift parameter* β is either dictated by the problem considered, or otherwise is set to 1.

3.1. The Levin transformation and the W-algorithm of Sidi

The Levin transformation is based on the remainder expansion (31) using the explicit remainder estimates ω_n. Hence, upon combining (31) and (27) and truncating the resulting asymptotic expansion after k terms, where k is the transformation order, we obtain the so-called *model sequence*

$$S_n = S + \omega_n \sum_{j=0}^{k-1} c_j x_n^{-j}, \quad n \geq 0, \ k \geq 1, \tag{33}$$

where the remainder estimates ω_n and the interpolation points x_n are assumed specified. The Levin transformation is constructed to be exact for this model sequence, which has $k+1$ unknowns, viz., the limit S and the coefficients $c_0, c_1, \ldots, c_{k-1}$. Thus, given a sequence of partial sums $\{S_n, S_{n+1}, \ldots, S_{n+k}\}$, we may use (33) to set up a system of $k+1$ linear equations and solve it for S using Cramer's rule, which gives S as a ratio of two $(k+1) \times (k+1)$ determinants. Upon expanding the numerator and denominator determinants by minors of the first column, one then finds [79, Section 10.5]

$$S_n^{(k)} = \sum_{j=0}^{k} \pi_n^{(k,j)} \frac{S_{n+j}}{\omega_{n+j}} \bigg/ \sum_{j=0}^{k} \pi_n^{(k,j)} \frac{1}{\omega_{n+j}}, \quad n \geq 0, \ k \geq 1, \ S_n^{(0)} = S_n, \tag{34}$$

where

$$\pi_n^{(k,j)} = \prod_{\substack{m=0 \\ m \neq j}}^{k} \frac{1}{x_{n+m}^{-1} - x_{n+j}^{-1}}. \tag{35}$$

In this section, we adopt the convention that the subscript of the transformation symbol refers to the sequence element index and the superscript denotes the transformation order. This notation is not always followed in the literature, where the subscripts and superscripts may have the opposite meaning, or the superscripts are dispensed with in favor of two subscripts. In the case of equispaced nodes, with x_n given by (32), the last expression may be further specialized as

$$\pi_n^{(k,j)} = (-1)^j \binom{k}{j} \left(\frac{\beta + n + j}{\beta + n + k} \right)^{k-1}, \tag{36}$$

where we have omitted the j-independent terms that cancel in the quotient (34), but inserted the scaling factor $(\beta + n + k)^{k-1}$ in the denominator to prevent overflows.[18, p.214] With properly selected reminder estimates ω_n, the values obtained from (34) with $k = 1, 2, \ldots$

will be much closer to the limit S than the last computed partial sum S_{n+k}. The transformation (34), which was derived by Levin [80], [5, p.94] in the special case of equispaced nodes with $\beta = 1$, will be referred to as the general Levin (GL) transformation.

The properties of the GL transformation critically depend on the choice of the remainder estimates ω_n. If the latter do not invoke the sequence elements S_n, the transformation is *linear*; otherwise, it is *nonlinear*. For example, setting $\omega_n = x_n^{-1}$ results in the Richardson extrapolation scheme,[78], [77, p.72] in which case, assuming equispaced nodes x_n, the denominator sum in (34) can be evaluated by the identity [81, p.4]

$$\sum_{j=0}^{k} (-1)^j \binom{k}{j} (\beta + n + j)^k = (-1)^k k!, \tag{37}$$

thus reducing (34) to the Salzer summation formula,[82],[79, p.35] which is a linear transformation effective for certain logarithmically convergent sequences $\{S_n\}$ that are well approximated by rational functions of n.[83] However, the following remainder estimates:

$$\omega_n = \begin{cases} u_n \text{ or } u_{n+1} & (t- \text{ or } d-\text{transformation}) \\ x_n u_n & (u-\text{transformation}) \\ \dfrac{u_n u_{n+1}}{u_n - u_{n+1}} & (v-\text{transformation}) \end{cases} \tag{38}$$

introduced by Levin [80] and (in the case of the d-transformation, which may be considered as a modified t-transformation) by Smith and Ford,[84] give rise to the most effective extrapolation methods. There is no agreed-upon name for the modified t-transformation, which has been variously referred to as d-, t'-, and $\tilde{t}$-transformation.[18, p.215], [12,78,85] The acceleration properties of the original and modified t-transformations are similar, but the latter requires the computation of one extra term of the series for the same transformation order. Note that we may write $u_n = \Delta S_{n-1} = S_n - S_{n-1}$, where Δ denotes the *forward difference* operator. Hence, with the remainder estimates (38), the GL transformation is non-linear, because ω_n depends explicitly on at least one element of $\{S_n\}$. Numerical experiments [84,86] supported in some cases by rigorous analysis [79, Section 10.5], [78] indicate that the t-transformation accelerates linear convergence and is well suited for alternating series, but fails to accelerate logarithmic convergence, whereas the u- and v-transformations are more versatile, as they accelerate both logarithmic and linear convergence. The u-transformation has been pronounced "the best available across-the-board method". [84]

The GL transformation (34) is non-recursive, i.e. it has to be computed from scratch each time the order k is increased. This drawback was remedied by Fessler et al. [87] who derived a recursive scheme for the computation of (34) in the case of equidistant nodes. In the general case, the GL transformation can be efficiently computed by the recursive W-algorithm of Sidi, which utilizes the properties of the *divided difference* operator δ^k, defined as

$$\delta^{k+1}(u_n) = \frac{\delta^k(u_{n+1}) - \delta^k(u_n)}{x_{n+k+1}^{-1} - x_n^{-1}}, \quad n \geq 0, \ k \geq 0, \tag{39}$$

with $\delta^0(u_n) \equiv u_n$.[88], [89, Section 3.8], [77, Section 2.7], [90, Chapter 7] To derive this algorithm, we rewrite (33) as

$$(S_n - S)/\omega_n = \sum_{j=0}^{k-1} c_j x_n^{-j} \tag{40}$$

and observe that the right-hand side member is a polynomial of degree $k-1$ in the variable x_n^{-1}. Hence, if we consider S_n and ω_n as functions of the continuous variable x evaluated at $x = x_n^{-1}$ and apply δ^k to both sides of (40), we obtain

$$\delta^k[(S_n - S)/\omega_n] = 0, \qquad (41)$$

because the divided difference operator of order k annihilates polynomials of degree $k-1$ or less. Finally, invoking the linearity of δ^k, we may express the solution S as

$$S_n^{(k)} = \frac{A_n^{(k)}}{B_n^{(k)}} = \frac{\delta^k(S_n/\omega_n)}{\delta^k(1/\omega_n)}, \qquad (42)$$

where, in view of (39), both the numerator and the denominator can be computed by the same recurrence formula

$$R_n^{(k+1)} = \frac{R_{n+1}^{(k)} - R_n^{(k)}}{x_{n+k+1}^{-1} - x_n^{-1}}, \quad n \geq 0, \ k \geq 0, \qquad (43)$$

with the staring values $A_n^{(0)} = S_n/\omega_n$ and $B_n^{(0)} = 1/\omega_n$, respectively. The elements connected by (43) form a triangle

$$\begin{array}{cc} R_n^{(k)} & R_n^{(k+1)} \\ R_{n+1}^{(k)} & \end{array}, \qquad (44)$$

which suggests their arrangement in the two-dimensional table, where the subscript n is the row index and the superscript k is the column index. Assuming, for simplicity, that the acceleration commences with $n = 0$, the table corresponding to the transformation order k has the form

$$\begin{array}{llllll} R_0^{(0)} & R_0^{(1)} & \cdots\cdots\cdots\cdots & & & R_0^{(k)} \\ R_1^{(0)} & R_1^{(1)} & \cdots\cdots\cdots & & R_1^{(k-1)} & \\ R_2^{(0)} & \cdots & & R_2^{(k-2)} & & \\ \vdots & \cdot\cdot\cdot & & & & \\ R_k^{(0)} & & & & & \end{array}, \qquad (45)$$

where the first column comprises the starting values and the computations proceed along counterdiagonals, which are added to the table as k is increased. Since the sum of the row and column indices of the kth counterdiagonal is constant and equal to k, it is convenient to rewrite the recurrence formula (43) as [78]

$$R_{k-j}^{(j)} = \frac{R_{k-j+1}^{(j-1)} - R_{k-j}^{(j-1)}}{x_k^{-1} - x_{k-j}^{-1}}, \quad k \geq 1, \ 1 \leq j \leq k. \qquad (46)$$

This recursion is performed twice, to update the numerator and the denominator of (42) using the appropriate starting values, and the accelerated sequence elements are returned as $S_0^{(k)} = A_0^{(k)}/B_0^{(k)}$. In computer implementation, the counterdiagonals can be processed in a one-dimensional array of expandable length.

A pseudocode for the GL transformation implemented via the W-algorithm is given in Algorithm 3. Here, the arrays $X(0:kmax)$, $A(0:kmax)$, and $B(0:kmax)$, where $kmax$ is the maximum transformation order, store the interpolation points $\{x_n\}$ and the numerator and denominator counterdiagonals, respectively.

Algorithm 3 The general Levin transformation via the W-algorithm of Sidi

---

    function LEVINSIDI(k, S_k, ω_k, X, A, B)   ▷ Invoke with $k = 0, \ldots, kmax$
        $B(k) \leftarrow 1/\omega_k$
        $A(k) \leftarrow S_k * B(k)$
        for $j \leftarrow 1, k$ **do**
            $d \leftarrow 1/X(k) - 1/X(k-j)$
            $A(k-j) \leftarrow (A(k-j+1) - A(k-j))/d$
            $B(k-j) \leftarrow (B(k-j+1) - B(k-j))/d$
        return $A(0)/B(0)$ ▷ Return $S_0^{(k)}$

3.2. The Shanks transformation and the ϵ-algorithm of Wynn

The Shanks transformation [91] results from the choice $\psi_j(n) = \Delta S_{n+j} = u_{n+j+1}$ in (30), which leads to the model sequence

$$S_n = S + \sum_{j=0}^{k-1} c_j \, \Delta S_{n+j}. \tag{47}$$

Essentially, this means that the limit S of the series (25) is approximated by the partial sum S_n plus a weighted sum of the next k terms $u_{n+1}, u_{n+2}, \ldots, u_{n+k}$.[78] As in the case of the Levin transformation, we may use (47) to set up a system of $k+1$ linear equations with the limit S and the coefficients $c_0, c_1, \ldots, c_{k-1}$ as unknowns and solve it for S using Cramer's rule. This solution, given as a ratio of determinants, defines the Shanks transformation $e_k(S_n)$, which is strongly nonlinear. It follows from the model sequence (47) that the sequence elements $\{S_n, \ldots, S_{n+2k}\}$ are needed to form this solution, hence $e_k(S_n)$ is a transformation of order $2k$. The determinantal form of the Shanks transformation is computationally expensive and is never used. Instead, $e_k(S_n)$ is computed by the so-called ϵ-algorithm of Wynn,[92] which is a recursive scheme given as

$$\epsilon_n^{(k+1)} = \epsilon_{n+1}^{(k-1)} + 1/\big[\epsilon_{n+1}^{(k)} - \epsilon_n^{(k)}\big], \quad k, n \geq 0, \tag{48}$$

with the starting values $\epsilon_n^{(0)} = S_n$ and $\epsilon_n^{(-1)} = 0$. The elements connected by the recursion (48) form a rhombus

$$\begin{matrix} & \epsilon_n^{(k)} & \epsilon_n^{(k+1)} \\ \epsilon_{n+1}^{(k-1)} & \epsilon_{n+1}^{(k)} & \end{matrix}, \tag{49}$$

which suggests an arrangement of $\epsilon_n^{(k)}$ in a two-dimensional ϵ-table similar to (45). Wynn [92] proved that the table elements with even superscripts give the Shanks transformation, whereas the elements with odd superscripts are only auxiliary quantities. Consequently, for even $n = 2k$, our transformation is

$$\{S_0, S_1, \ldots, S_{2k}\} \to \epsilon_0^{(2k)} = e_k(S_0), \tag{50}$$

whereas for odd $n = 2k + 1$, we use

$$\{S_1, S_2, \ldots, S_{2k+1}\} \to \epsilon_1^{(2k)} = e_k(S_1), \tag{51}$$

assuming that the extrapolation commences with the $n = 0$ term of the sequence. The recursion (48) proceeds along the counterdiagonals of the ϵ-table, with a new counterdiagonal added when the transformation order k is incremented by o ne and a new sequence element

(starting value) S_k becomes available. Since the elements of the kth counterdiagonal are given as $\epsilon_{k-j}^{(j)}$, $k \geq 1$, $1 \leq j \leq k$, it is convenient to rewrite the recurrence formula (48) as [78]

$$\epsilon_{k-j}^{(j)} = \epsilon_{k-j+1}^{(j-2)} + 1/\left[\epsilon_{k-j+1}^{(j-1)} - \epsilon_{k-j}^{(j-1)}\right], \quad k \geq 1, \ 1 \leq j \leq k. \tag{52}$$

Here too, the counterdiagonals can be processed in an expandable one-dimensional array. However, because (52) is a rhombus, rather than a triangle scheme, two auxiliary variables are also needed to store the table elements that are required in the processing of the next counterdiagonal, but would otherwise be overwritten during the computation of the current counterdiagonal.[78], [18, p.213]

A pseudocode for the Shanks transformation implemented via the ϵ-algorithm of Wynn is given in Algorithm 4. Here, the array $E(0 : kmax)$, where $kmax$ is the maximum transformation order, stores the counterdiagonals, and a and b are the auxiliary variables referred to above. The zero divide referred to in the comment line may occur if the algorithm is not stopped after convergence has been reached.

Algorithm 4 The Shanks transformation via the ϵ-algorithm of Wynn

---

    function SHANKSWYNN(k, S_k, E)   ▷ Invoke with $k = 0, \ldots, kmax$
        $E(k) \leftarrow S_k$
        $b \leftarrow 0$
        for $j \leftarrow 1, k$ **do**
            $a \leftarrow b$
            $b \leftarrow E(k-j)$
            $d \leftarrow E(k-j+1) - b$
            $E(k-j) \leftarrow a + 1/d$ ▷ Should check for zero divide
        return $E(k \bmod 2)$ ▷ Return $e_k(S_0)$ or $e_k(S_1)$

The Shanks transformation, or equivalently the ϵ-algorithm, is effective for alternating and linear convergence, but is unable to accelerate logarithmic convergence.[86] Since no remainder estimates are required, this algorithm is user-friendly and is implemented in some standard software libraries, such as QUADPACK.[93] Also, when applicable, the ϵ-algorithm is known to be robust in the presence of noise or irregularities in the sequence elements, and is thus frequently used as a reference benchmark method for other extrapolation techniques. Hence, a good extrapolation method should perform at least as well as the ϵ-algorithm.

3.3. *The weighted-averages method and the Mosig–Michalski algorithm*

Whereas the Levin and Shanks transformations are constructed to be exact for a certain model sequence, the weighted averages (WA) method belongs to the class of iterative sequence transformations, which are obtained by repeated application of some very simple transformation.[78,85,94,95] As the name indicates, the WA method is generated by a weighted mean of two consecutive partial sums [12,37,96–98]:

$$S_n' = \frac{W_n S_n + W_{n+1} S_{n+1}}{W_n + W_{n+1}}, \tag{53}$$

where W_n denotes the weight associated with S_n. In view of (27), we may also express the above as

$$S_n' = S + \frac{W_n r_n + W_{n+1} r_{n+1}}{W_n + W_{n+1}} \equiv S + r_n', \tag{54}$$

where r'_n is the remainder of the new sequence element S'_n. Upon introducing the weights ratio coefficient

$$\eta_n = -\frac{W_n}{W_{n+1}} \tag{55}$$

into (53) and (54), we may express them as

$$S'_n = \frac{S_{n+1} - \eta_n S_n}{1 - \eta_n}, \quad \frac{r'_n}{r_n} = \frac{r_{n+1}/r_n - \eta_n}{1 - \eta_n}. \tag{56}$$

We now assume that the remainders r_n possess the asymptotic expansion (31) and find

$$\frac{r_{n+1}}{r_n} = \frac{\omega_{n+1}}{\omega_n}\big[1 + \mathcal{O}\big(x_n^{-2}\big)\big], \quad n \to \infty, \tag{57}$$

which suggests selecting the weights ratio in (56) as

$$\eta_n = \frac{\omega_{n+1}}{\omega_n}. \tag{58}$$

With this choice of η_n, and assuming that the denominator in (56) has the asymptotic behavior

$$1 - \eta_n = \mathcal{O}\big(x_n^{-\sigma}\big), \quad n \to \infty, \tag{59}$$

where $\sigma < 2$, we find that the acceleration condition (28) is satisfied with $\mu = 2 - \sigma$. It can be shown that $\sigma = 0$ (and thus $\mu = 2$) for all sequences other than those with logarithmic convergence, for which $\sigma = 1$ ($\mu = 1$). In view of (56), we conclude that the remainders r'_n have the same asymptotic form (31) as r_n, except that the remainder estimates ω_n are now scaled by the factor $x_n^{-\mu}$. Consequently, the sequence $\{S'_n\}$ may again be accelerated by the transformation in (56), using the properly modified ω_n in (58). We thus arrive at the recursive scheme

$$S_n^{(k+1)} = \frac{S_{n+1}^{(k)} - \eta_n^{(k)} S_n^{(k)}}{1 - \eta_n^{(k)}}, \quad n, k \geq 0, \tag{60}$$

with the starting values $S_n^{(0)} = S_n$ and the coefficients

$$\eta_n^{(k)} = \eta_n \left(\frac{x_n}{x_{n+1}}\right)^{\mu k} = \frac{\eta_n}{\big(1 + \Delta x_n/x_n\big)^{\mu k}} \approx \frac{\eta_n}{\big(1 + \mu k\, \Delta x_n/x_n\big)}, \tag{61}$$

where the second expression is an asymptotic form applicable for large n. Note that with the weights ratio coefficients fixed as $\eta_n^{(k)} = \lambda$, (60) reduces to the so-called λ-transformation, [99–101] which for $\lambda = -1$ further reduces to the classical Euler transformation.[102], [103, p.124] The power of WA derives from the fact that the coefficients $\eta_n^{(k)}$ are not fixed, but adaptively adjusted.

The algorithm given above is equivalent to the so-called generalized WA procedure,[12] which is also referred to as the Mosig–Michalski algorithm.[85] Although the asymptotic form of the weights ratio coefficient given in (61) can hardly be justified, as the recursion is often applied beginning with $n = 0$ and satisfactory convergence is in many cases reached before n exceeds 10, Michalski [12] observed that in practice this particular approximation results in conspicuously better performance than the exact form.

The recursion (60) is a triangle scheme similar to (43), hence the elements $S_n^{(k)}$ may be arranged in a two-dimensional table as in (45). Here again, we assume that the acceleration is applied commencing with S_0. The first column of this table comprises the starting values $S_k^{(0)} = S_k$ and the recursion proceeds along the counterdiagonals, where the sum of the row and column indices is constant. The table is expanded by a new counterdiagonal for each

new sequence element S_k and the upper-right corner element $S_0^{(k)}$ is the best approximation of S, given the partial sums $\{S_0, \ldots, S_k\}$. Since the elements of the kth counterdiagonal are given as $S_{k-j}^{(j)}, k \geq 1, 1 \leq j \leq k$, we may rewrite the recurrence formula (60) in the more convenient form

$$S_{k-j}^{(j)} = \frac{S_{k-j+1}^{(j-1)} - \eta_{k-j}^{(j-1)} S_{k-j}^{(j-1)}}{1 - \eta_{k-j}^{(j-1)}}, \quad k \geq 1, \ 1 \leq j \leq k. \tag{62}$$

Similar to the W-transformation, this recursion may be processed in an expandable one-dimensional array.

Algorithm 5 The weighted averages method via the Mosig–Michalski algorithm

---

    function MOSIGMICHALSKI$(\mu, k, S_k, \Omega_k, X, R)$ ▷ Invoke with $k = 0, \ldots, kmax$
         $R(k) \leftarrow S_k$
         for $j \leftarrow 1, k$ **do**
             $d \leftarrow X(k-j+1) - X(k-j)$
             $\eta \leftarrow \Omega_k / (1 + \mu * (j-1) * d / X(k-j))$
             $R(k-j) \leftarrow (R(k-j+1) - \eta * R(k-j)) / (1 - \eta)$
         return $R(0)$ ▷ Return $S_0^{(k)}$

A pseudocode of the WA method, implemented by a simplified Mosig–Michalski algorithm with the asymptotic weights ratio coefficients, is given in Algorithm 5. Here, the arrays $X(0 : kmax)$ and $R(0 : kmax)$, where $kmax$ is the maximum transformation order, store the nodes $\{x_n\}$ and the counterdiagonals, respectively. Recall that the coefficient μ should be set to 1 (for logarithmic convergence) or 2 (otherwise). The parameter $\Omega_k \equiv \eta_{k-1}, k \geq 1$, is the ratio of the current and previous remainder estimates at the transformation order k – cf. (58). Note that Ω_0 is not invoked by the algorithm and may be set to zero.

Since the WA method depends on the explicit remainder estimates $\{\omega_n\}$, it may be classified as a Levin-type transformation,[85] and is thus expected to have similar properties as the GL transformation, if the same remainder estimates are employed.

3.4. Comparison of extrapolation methods

To illustrate the relative merits of the Levin–Sidi, Mosig–Michalski, and Shanks–Wynn algorithms, we compare their performance when applied to the following slowly convergent series

$$\Sigma_1 = \sum_{i=0}^{\infty} \frac{(-1)^i}{\sqrt{i+1}} \approx 0.604898643421630, \tag{63}$$

$$\Sigma_2 = \sum_{i=0}^{\infty} \frac{(4/5)^{i+1}}{i+1} = \ln(5), \tag{64}$$

$$\Sigma_3 = \sum_{i=0}^{\infty} \frac{1}{(i+1)^2} = \frac{\pi^2}{6}, \tag{65}$$

which exhibit alternating, linear, and logarithmic convergence, respectively.[104] As the performance measure we use plots of the number of significant digits (defined here as $-\log_{10} |\varepsilon_r|$, where ε_r is the relative error) in the transformed sequence vs. the number of terms evaluated. It will be recalled that the Levin–Sidi and Mosig–Michalski algorithms come in different variants – t, d, u, and v – depending on which of the numerical remainder
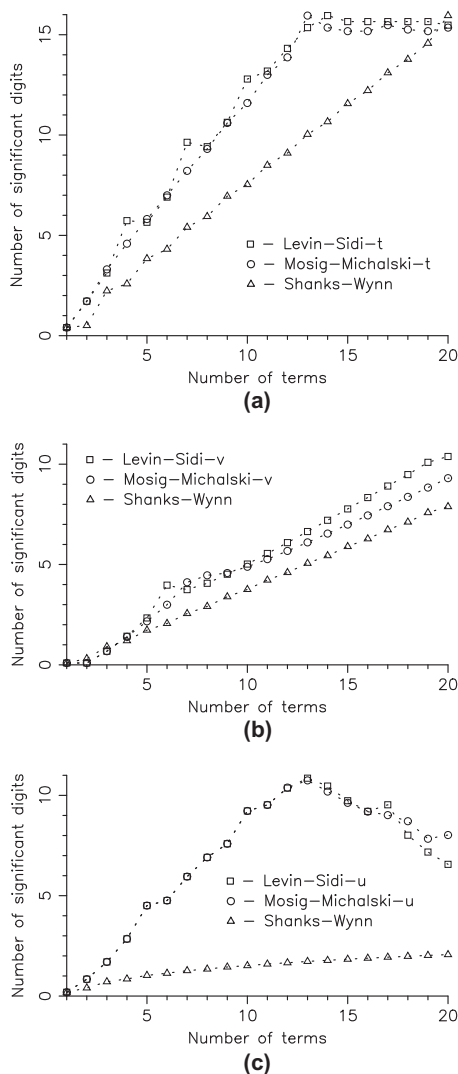
Figure 3. Performance of the extrapolation methods for (a) Σ_1 (alternating), (b) Σ_2 (monotone linear), and (c) Σ_3 (monotone logarithmic).

estimates (38) is employed. We apply these methods using the equidistant interpolation points (32), with the shift parameter $\beta = 1$ and without acceleration delay, i.e. the transformation commences with the partial sum S_0. There is usually little or no advantage in delaying the commencement of acceleration, and increasing β much beyond 1 slows down the convergence rate.

In Figure 3, we compare the Levin–Sidi-t, Mosig–Michalski-t, and Shanks-Wynn extrapolation algorithms for the series Σ_1, Σ_2, and Σ_3. The performance of the former two methods is seen to be almost identical for the alternating series Σ_1, with both reaching machine precision accuracy in under 15 iterations and outperforming the ϵ-algorithm by a significant margin. The Levin–Sidi-d and Mosig–Michalski-d algorithms perform similarly to the respective t-variants in this case. For the monotone linear series Σ_2, we use

the v-variants of the Levin–Sidi and Mosig–Michalski algorithms. Again, these methods perform similarly and both outperform the Shanks-Wynn algorithm, but the convergence is significantly slower than for the alternating case. For the monotone logarithmic series Σ_3, we use the u-variants of the Levin–Sidi and Mosig–Michalski algorithms. The performance of these two methods is again almost identical, but the curve is not monotonic in this case, as it bends down after climbing to approximately 11 significant digits in about this many iterations. This behavior can be attributed to the numerical instability of Levin-type algorithms for high transformation orders, caused by cancellation errors that are known to occur in the monotone case.[87] The Shanks-Wynn algorithm performs poorly, as was to be expected for logarithmic convergence.

The above results confirm that the Levin–Sidi and Mosig–Michalski algorithms are among the most versatile and powerful extrapolation methods currently known. Since both algorithms are equally easy to implement, which one should be selected is largely a matter of personal preference. However, when approaching a new class of problems, one will be well-advised to try at least two different extrapolation methods, preferably including the Shanks-Wynn algorithm among them.

4. Partition-extrapolation method for oscillatory integral tails

We now describe a powerful technique for automatic integration of Sommerfeld integral tails, which combines the use of progressive quadratures, such as the DE rule, and the extrapolation methods discussed above. Hence, consider the tail integral

$$S = \int_a^\infty f(\xi)\, \mathrm{d}\xi, \tag{66}$$

over the real axis segment (a, ∞), where the integrand function $f(\xi)$ is oscillatory and may be slowly decaying or even divergent, as in (1). We compute S as the limit of the sequence of the partial sums

$$S_n = \sum_{i=0}^{n} u_i, \quad u_i = \int_{\xi_{i-1}}^{\xi_i} f(\xi)\, \mathrm{d}\xi, \tag{67}$$

as $n \to \infty$, where $\xi_{-1} \equiv a$ and ξ_i, $i = 0, 1, \ldots$, are suitably selected *break points*, e.g. the consecutive zeros of $f(\xi)$ – see Figure 1. The finite-range integrals u_i are readily computed by a low-order Gauss-Legendre quadrature. For Sommerfeld integrals, the 16th-order Gauss-Legendre rule typically yields double precision machine accuracy for the partial integrals between consecutive zeros of the Bessel function,[105,106] but for greater reliability, one can resort to Gauss-Kronrod or Patterson rules,[58, p.131], [107] or to the progressive Clenshaw-Curtis quadrature,[108] which provide error estimates. The recursive DE tanh-sinh rule described in Section 2 can also be employed, even in the absence of endpoint singularities. Typically, the sequence $\{S_n\}$ is slowly convergent, hence acceleration by one of the extrapolation methods discussed in Section 3 is a good strategy. The combination of the partition of the integral in (67) with extrapolation to speed the convergence of $\{S_n\}$ gives rise to the *partition-extrapolation* (PE) method, which is the subject of many papers, including [12,15,102,109–120].

In order to effectively utilize extrapolation methods to accelerate (67), it is advisable to first investigate the behavior of the remainders

$$r_n = S_n - S = -\int_{\xi_n}^\infty f(\xi)\, \mathrm{d}\xi. \tag{68}$$

Here, we assume here that for $\xi > b$, where $b > a$ is a suitably selected break point, the integrand function in (66) may be expressed as

$$f(\xi) \approx g(\xi) p(\xi), \tag{69}$$

where $g(\xi)$ has the asymptotic expansion

$$g(\xi) \sim \frac{e^{-\xi\zeta}}{\xi^\alpha} \sum_{j=0}^\infty a_j \xi^{-j}, \quad \xi \to \infty, \tag{70}$$

with the parameter $\zeta \geq 0$, while $p(\xi)$ is periodic, with half-period q, such that

$$p(\xi + q) = -p(\xi). \tag{71}$$

Consequently, we may assume equidistant break points

$$\xi_i = b + iq, \quad i = 0, 1, \ldots, \tag{72}$$

where b is the first zero of $p(\xi)$ in the range where the representation (69) is applicable. We now substitute (69) into (68) and use (70) to obtain

$$r_n \sim -\sum_{j=0}^\infty a_j \int_{\xi_n}^\infty \frac{p(\xi) e^{-\xi\zeta}}{\xi^{\alpha+j}} \, d\xi, \tag{73}$$

which upon changing the variable of integration to $t = \xi - nq$ and using (71) becomes

$$r_n \sim -(-1)^n e^{-nq\zeta} \sum_{j=0}^\infty a_j \int_b^\infty \frac{p(t) e^{-t\zeta}}{(t + nq)^{\alpha+j}} \, dt. \tag{74}$$

Finally, upon expanding the inverse denominator of the integrand of (74) in a Taylor series about $t = b$ and integrating term-by-term, we arrive at the Poincaré-type expansion

$$r_n \sim \omega_n \sum_{j=0}^\infty c_j \xi_n^{-j}, \tag{75}$$

where c_j are n-independent coefficients and ω_n are the remainder estimates, given as

$$\omega_n = (-1)^{n+1} \frac{e^{-nq\zeta}}{\xi_n^\alpha}. \tag{76}$$

The break points ξ_n in (75)–(76) may also be replaced by the interpolation points x_n given by (32) with the shift factor $\beta = b/q$.

Based on the above analysis, we may select the most appropriate extrapolation method for the sequence of partial sums (67). Hence, upon using (75) and (76) in (29), we readily find that $\lambda = -e^{-q\zeta}$, which indicates that the sequence (67) is linearly convergent and alternating, and thus all four variants of the Levin-type methods, corresponding to the numerical remainder estimates (38), as well as the Shanks-Wynn algorithm, are applicable. Using the *analytical* remainder estimates derived above with the Levin–Sidi and Mosig–Michalski algorithms gives rise to what may be called *a*-variants of these methods. The parameter Ω_k in the Mosig–Michalski algorithm then becomes

$$\Omega_k \equiv \frac{\omega_k}{\omega_{k-1}} = -e^{-q\zeta} \left(\frac{x_{k-1}}{x_k}\right)^\alpha, \quad k \geq 1, \tag{77}$$

with $\Omega_0 = 0$, and from (59) it follows that $\sigma = 0$, and thus $\mu = 2$. The Levin-Sidi-*a* algorithm is closely related to the *W*-transformation of Sidi,[111] while the Levin-Sidi-*d* algorithm is tantamount to his "user-friendly" *mW*-transformation.[114]

As an example, consider the integral

$$I_\nu(a, z, \rho) = \int_a^\infty e^{-\xi z} J_\nu(\xi\rho)\xi^\nu \mathrm{d}\xi = \frac{(2\nu-1)!!\,\rho^\nu}{(z^2 + \rho^2)^{\nu+1/2}} - \int_0^a e^{-\xi z} J_\nu(\xi\rho)\xi^\nu \mathrm{d}\xi, \qquad (78)$$

which follows from a well-known identity [81, Equation 6.623.1], [105] by splitting the $(0, \infty)$ integration range at an arbitrary point $a > 0$. For $\nu = 0, 1$, and 2, the integrals $I_\nu(a, z, \rho)$ resemble the static forms of the tail integrals that arise in the electromagnetic field computation in layered media.[4] With $z = 0$ and $\nu = 1, 2$, the left-hand side integrals in (78) are formally divergent, but are defined in the sense of Abel summability.[77, p.381], [21,105] As the finite-range integral on the right-hand side of (78) can readily be evaluated to machine accuracy by a numerical quadrature – such as the tanh-sinh rule discussed in Section 2, the infinite integral $I_\nu(a, z, \rho)$ may serve as a realistic test bed for the extrapolation methods, allowing in particular to examine the effect of changing the lower limit a. In view of the asymptotic form of the Bessel function [5, p.223]

$$J_\nu(\xi\rho) \sim \sqrt{\frac{2}{\pi\xi\rho}} \cos(\xi\rho - \nu\pi/2 - \pi/4), \quad z \to \infty, \qquad (79)$$

$I_\nu(a, z, \rho)$ fits in the framework of the PE method with $\zeta = z, \alpha = 1/2 - \nu$, and $q = \pi/\rho$. The natural choice of the break points ξ_i for the partial integrals u_i in (67) are in this case the consecutive zeros of $J_\nu(\xi\rho)$ (Longman partition).[102,117] Since the exact Bessel function zeros can only be found by iterative procedures,[117,121] other alternatives have been proposed, such as the asymptotic zeros based on the expansion (79) (Lyness partition),[113] or just the equidistant nodes separated by the asymptotic half-period q (Sidi partition).[111,114,122] Still another alternative is to use the asymptotic *extrema* of the Bessel function, which somewhat dampens the oscillation of the partial sums and results in a modest convergence improvement of the PE method.[113] The Sidi partition method is the simplest, but it has been found unreliable, as there are certain values of ρ for which the sequence of the partial sums (67) ceases to be strictly alternating, with the concomitant precipitous performance degradation of the PE method.[12] We have observed, however, that this problem does not occur if the lower integration limit a is always placed at the first Bessel function zero that is greater than the original a. Of course, if there is no flexibility in the choice of the lower integral limit, this shift requires an extra partial integral to bridge the gap between the original and the modified values of a. Hence, if the *n*th zero of $J_\nu(\cdot)$ is denoted by $j_{\nu n}$, in the modified Sidi partition we relocate a to $j_{\nu,m}/\rho > a$, where m is the lowest zero index for which this condition is satisfied, and set the break points according to (72), with $b = a + q$. We adopt this modified Sidi partition in the examples that follow. For $j_{\nu n}$, we use the tabulated values [123, p.409] if $n \leq 10$, and the large-*n* McMahon's expansions [123, p.371] otherwise.

In Figure 4, we show plots of the original and accelerated partial sums for the integral $I_2(5.13562, 0, 1)$, using the PE method with the Mosig–Michalski-*a* algorithm. Note that the chosen value of the lower limit a is the first zero of $J_2(\xi\rho)$, consistent with the modified Sidi partition. The original sequence S_n is divergent and alternating, while the transformed sequence S'_n converges rapidly to the exact integral value S – in this case to the so-called *antilimit*.[78], [83, p.248] In Figure 5, we compare the performance of the PE methods for the integrals $I_0(3.247, 0, 7.5)$ and $I_0(3.247, 1, 7.5)$, using the modified Sidi partition and
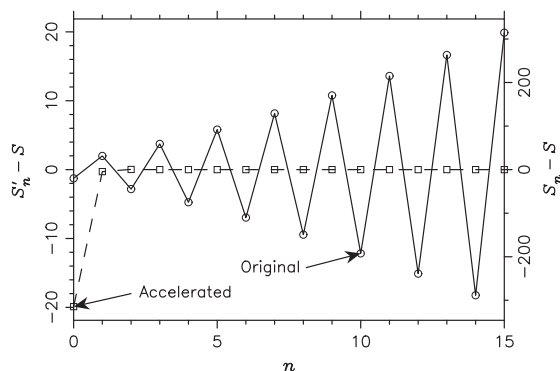
Figure 4. The original (S_n) and accelerated (S'_n) partial sums for the integral $I_2(5.13562, 0, 1)$, using the PE method with the modified Sidi partition and extrapolation by the Mosig–Michalski-a algorithm. Note that the scales of these two sequences are vastly different. Although S_n is divergent, S'_n rapidly converges to the exact value of the integral (the antilimit), $S \approx -10.07949$.
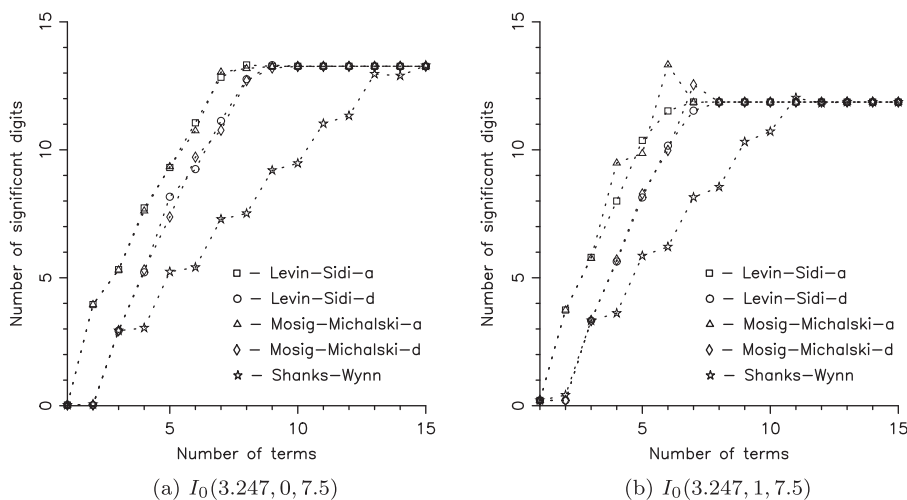

(a) $I_0(3.247, 0, 7.5)$ (b) $I_0(3.247, 1, 7.5)$

Figure 5. Performance of different PE methods for the integrals.

different extrapolation algorithms. The difference between these two integrals is the strong exponential decay exhibited by the integrand of the second. Note that the chosen value $a = 3.247$ is consistent with the Sidi partition. As the performance measure we use plots of the number of significant digits vs. the number of terms (partial integrals) evaluated. All computations are done using double precision arithmetic with approximately 15 significant digits. We present plots for the a- and d-variants of the Levin-type algorithms (the other variants produce very similar results) and for the Shanks-Wynn algorithm. It is noted that the Levin–Sidi and Mosig–Michalski perform similarly and both are more effective than the Shanks-Wynn algorithm. Furthermore, the a-variants, using analytical remainder estimates, have a slight advantage over the d-variants. However, the latter are more user-friendly, as no asymptotic analysis of the integrand function is required. In Figure 6, we plot the number of significant digits obtained after 9 extrapolation steps (at the cost of 10 partial integrals) for the PE method using the modified Sidi partition with the Levin–Sidi-t, Mosig–Michalski-t, and
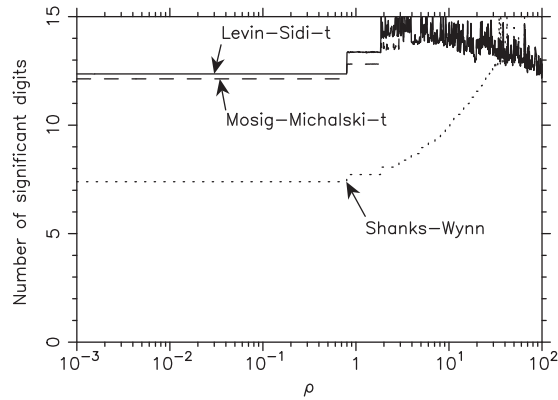
Figure 6. Accuracy of the PE method achieved after 9 extrapolation steps (10 partial integrals) for $I_0(a, 0, \rho)$ with ρ varying over a five-decade range. Here a depends on ρ consistent with the Sidi partition with the constraint $a \geq 3$.

Shanks–Wynn algorithms, applied to the integral $I_0(a, 0, \rho)$ with the threshold value of $a = 3$ and varying ρ. Note that both Levin-type methods, using numerical remainder estimates, achieve at least 12 digits of accuracy over the five-decade ρ-range and no breakdowns are observed. The Shanks-Wynn algorithm is competitive for large ρ only. These plots were generated using 250 points per decade. The staircase behavior is due to the passing of the consecutive Bessel function zeros through the threshold value a and the concomitant jumps in the lower integration limit as ρ is increased. The growing frequency of these steps is due to the logarithmic ρ-scale used.

A pseudocode for the PE method computation of tail integrals comprising Bessel functions is given in Algorithm 6, which uses the Levin-Sidi transformation implemented via the function LevinSidi – see Algorithm 3 of Section 3.

Algorithm 6 The partition-extrapolation algorithm via LevinSidi

---

    procedure PartExtrap(fun, a, tol, val)   ▷ fun is the integrand function
        global q
        parameter $kmax \leftarrow 10, \epsilon \leftarrow 1E-6$
        allocate $X(-1:kmax), A(0:kmax), B(0:kmax)$
        $X(-1) \leftarrow a$
        $s \leftarrow 0$
        for $k \leftarrow 0, kmax$ **do**
            $X(k) \leftarrow X(k-1) + q$
            TanhSinhQuad($fun, X(k-1), X(k), \epsilon, u$)
            $s \leftarrow s + u$
            $\omega \leftarrow u$ ▷ t-variant; $\omega \leftarrow X(k) * u$ gives u-variant
            $val \leftarrow$ LevinSidi(k, s, ω, X, A, B)
            if $k > 1$ **and** $abs(val - old) < tol * abs(val)$ **exit**   ▷ Convergence check
            $old \leftarrow val$
        return val ▷ Return integral value

It is convenient in the present context to declare the array $X(:)$, holding the interpolation points in Algorithm 3, to have a starting index -1, rather than 0, so that $X(-1)$ may hold the lower integration limit a. For simplicity, we use the t-variant here, which is appropriate for alternating sequences (the u-variant can be implemented as easily by modifying just one

line of the code). The modified Sidi partition is assumed, with the lower integral limit a set to a Bessel function zero and the asymptotic half-period $q = \pi/\rho$ (which is accessible as a global variable). The partial integrals are evaluated by the procedure TANHSINHQUAD of Section 2. Since the singularity-suppressing property of the tanh-sinh rule is not needed here, the Gauss-Legendre or Romberg quadrature may be used instead.[18, p.218] For regular integrand functions, the Gaussian quadrature is typically less costly than the tanh-sinh rule.[124] The transformation is applied recursively until the stopping rule, controlled by the relative error parameter tol, is satisfied. For simplicity, the implemented error estimate is computed as the difference between the current and the previous extrapolated results. However, to reduce the chance of false convergence, a much safer strategy is to use the maximum difference between the current and the previous *two* extrapolated results.[117] A pseudocode for an alternative version of the PE method, using the Mosig–Michalski Algorithm 5 of Section 3 with analytical remainder estimates (the a-variant), is given in Algorithm 7. Note that the auxiliary function OMEGA implements the coefficient Ω_k given by (77).

Algorithm 7 The partition-extrapolation algorithm via MOSIGMICHALSKI

> **procedure** PARTEXTRAP(fun, a, tol, val)   ▷ fun is the integrand function
>   **global** q, z, α
>   **parameter** $kmax \leftarrow 10, \epsilon \leftarrow 1E-6$
>   **allocate** $X(-1:kmax), R(0:kmax)$
> $\mu \leftarrow 2$
> $X(-1) \leftarrow a$
> $s \leftarrow 0$
>   **for** $k \leftarrow 0, kmax$ **do**
> $X(k) \leftarrow X(k-1) + q$
>   TANHSINHQUAD($fun, X(k-1), X(k), \epsilon, u$)
> $s \leftarrow s + u$
> $\Omega \leftarrow$ OMEGA(k, q, z, α, X)   ▷ a-variant
> $val \leftarrow$ MOSIGMICHALSKI(μ, k, s, Ω, X, R)
>   **if** $k > 1$ **and** $abs(val - old) < tol * abs(val)$ **exit**   ▷ Convergence check
> $old \leftarrow val$
>   **return** val ▷ Return integral value
> **function** OMEGA(k, q, z, α, X)
>   **if** $k = 0$ **then**
> $\Omega \leftarrow 0$
>   **else**
> $\Omega \leftarrow -exp(-q*z)*(X(k-1)/X(k))**\alpha$
>   **return** Ω

The PE method can also be used to evaluate Sommerfeld integrals, such as (2), involving a product of two Bessel functions of different orders. As an example, consider the integral [8]

$$I(z, \rho, \tau) = \int_0^\infty e^{-\xi z} J_0(\xi\rho) J_{3/2}(\xi\tau) \xi^{1/2} d\xi$$

$$= \sqrt{\frac{2}{\pi\tau^3}} \left(\text{arccot } v_+ - \frac{v_+}{v_+^2 + v_-^2} \right), \tag{80}$$
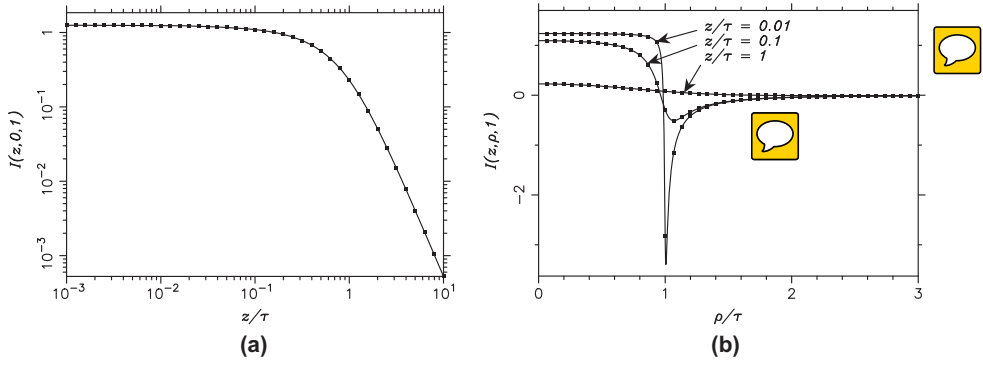
Figure 7. (a) Plot of $I(z, 0, 1)$ vs. z/τ. The integral is computed by the PE method (Algorithm 6) for $z/\tau < 1$ and by the mixed quadrature rule (Algorithm 2) for $z/\tau \geq 1$. (b) Plot of $I(z, \rho, 1)$ vs. ρ/τ, with $z/\tau = 0.01$, 0.1, and 1, computed by the PE method (Algorithm 7) with the Lucas decomposition. The integrated results are indicated by symbols and the exact results by solid lines.

where

$$v_\pm = \sqrt{\frac{d^2 \pm (z^2 + \rho^2 - \tau^2)}{2\tau^2}} , \quad d^2 = \sqrt{(z^2 + \rho^2 - \tau^2)^2 + 4z^2\tau^2}. \tag{81}$$

In the special case when $\rho = 0$, the first Bessel function in the product becomes identically 1, thus $I(z, 0, \tau)$ can still be computed by the PE method described above, where v is now half-integer. To illustrate, in Figure 7(a) we compare the integrated and exact results for $I(z, 0, 1)$ over a four-decade z/τ-range, with the former computed by the PE Algorithm 6 for $z/\tau < 1$, and by the mixed quadrature rule (Algorithm 2 of Section 2) for $z/\tau \geq 1$ (where $\tau = 1$ in the present case). The switching of the algorithms is beneficial, because in the latter range the behavior of the integrand function in (80) is dominated by the exponential decay, rather than the oscillations of the Bessel function.

In the general case, when $\rho > 0$, the integrals such as (80) are notoriously challenging to evaluate, due to the irregular oscillatory behavior of the product of the Bessel functions, which makes it difficult to devise a partition scheme that would yield a sequence of partial sums amenable to extrapolation. The method by Van Deun and Cools [105] should be mentioned here, in which the tail integral is evaluated analytically, after replacing the Bessel functions by their asymptotic expansions. In the special case where the Bessel functions have the same argument, i.e. $\rho = \tau$, the modified mW-transformation of Sidi [125] may be applicable. Hence, unless $z > \max(\rho, \tau)$, when the strong exponential decay makes the mixed rule applicable, we resort to the long-standing method of Lucas [8,106,126,127], which is based on the decomposition

$$J_v(\xi\rho)J_\sigma(\xi\tau) = \mathcal{J}_{v\sigma}^+(\xi; \rho, \tau) + \mathcal{J}_{v\sigma}^-(\xi; \rho, \tau), \tag{82}$$

with

$$\mathcal{J}_{v\sigma}^\pm(\xi; \rho, \tau) = 1/2\big[J_v(\xi\rho)J_\sigma(\xi\tau) \mp Y_v(\xi\rho)Y_\sigma(\xi\tau)\big], \tag{83}$$

where Y_v denotes the Bessel function of the second kind and order v.[5] Note the analogy with the trigonometric identity $\cos\alpha\cos\beta = 1/2[\cos(\alpha + \beta) + \cos(\alpha - \beta)]$. Since the large-argument form of Y_v is similar to that of J_v, except that the cosine in (79) is replaced

by a sine, it is readily found that

$$\mathcal{J}_{\nu\sigma}^{\pm}(\xi;\rho,\tau) \sim \frac{1}{\pi\xi\sqrt{\rho\tau}} \begin{cases} \sin[\xi(\rho+\tau)-(\nu+\sigma)\pi/2] \\ \cos[\xi(\rho-\tau)-(\nu-\sigma)\pi/2] \end{cases}, \quad \xi \gg 1. \tag{84}$$

Thus, for $\rho \neq \tau$, both terms in the decomposition (82) approach simple sinusoids with half-periods $q = \pi/|\rho \pm \tau|$, and for $\rho = \tau$, $\mathcal{J}_{\nu\sigma}^{+}$ is still asymptotically sinusoidal, but $\mathcal{J}_{\nu\sigma}^{-}$ approaches a monotonically decreasing function. The splitting (82) achieves the separation of the high- and low-frequency components in the product of the Bessel functions by adding and subtracting the term $Y_\nu(\xi\rho)Y_\sigma(\xi\tau)$. However, since Bessel functions of the second kind have large magnitudes for small arguments, to prevent the loss of accuracy due to catastrophic cancellation, the decomposition (82) should only be applied for ξ exceeding a suitable threshold value a, where the term $Y_\nu(\xi\rho)Y_\sigma(\xi\tau)$ no longer dominates. Following Lucas,[126] here we set $a = \max(y_{\nu,1}/\rho, y_{\sigma,1}/\tau)$, where $y_{\nu,1}$ denotes the first zero of $Y_\nu(\cdot)$. The initial zeros of $Y_\nu(\cdot)$ are tabulated for some integer and half-integer orders ν.[123, p.410], [123, p.467] For the first zero, an asymptotic expansion valid for large orders is also available,[123, p.371] which yields an acceptable approximation for ν as low as 1.[126] Based on these considerations, we split the integral (80) into three parts as

$$\begin{aligned} I(z,\rho,\tau) &= I_a(z,\rho,\tau) + I^+(a,z,\rho,\tau) + I^-(a,z,\rho,\tau) \\ &= \int_0^a e^{-\xi z} J_0(\xi\rho) J_{3/2}(\xi\tau)\,\xi^{1/2}\mathrm{d}\xi \\ &\quad + \int_a^\infty e^{-\xi z} \mathcal{J}_{03/2}^+(\xi;\rho,\tau)\,\xi^{1/2}\mathrm{d}\xi + \int_a^\infty e^{-\xi z}\mathcal{J}_{03/2}^-(\xi;\rho,\tau)\,\xi^{1/2}\mathrm{d}\xi. \end{aligned} \tag{85}$$

We evaluate the finite-range integral I_a by the tanh-sinh DE rule and the two tail integrals $I^{\pm}$ by the PE method – except in the special case $\rho = \tau$, when the integrand of I^- ceases to be oscillatory, and we employ the mixed quadrature rule. The use of the mixed rule is justified if the integrand has some exponential decay, i.e. $z > 0$. Otherwise, the DE rule (20)-(21) is more appropriate. Applying the PE method to $I^{\pm}(a,z,\rho,\tau)$, we use the modified Sidi partition and Algorithm 7 with $q = \pi/|\rho \pm \tau|$. To find the first zero of the integrand exceeding a, we bracket it by taking a few steps of size $q/4$ until a change in the sign of the function is detected, and apply the derivative-free Brent method.[18, p.454] The gap between the original and modified values of a is then bridged by another application of the tanh-sinh rule. Note that ρ cannot be arbitrarily close to zero with this approach, as this would push the break point a towards infinity. For very small ρ and $z > 0$, the integration range may have to be restricted, to prevent underflows.

To illustrate the performance of the PE method in conjunction with the Lucas decomposition, in Figure 7(b) we compare the integrated and exact results for $I(z,\rho,1)$, with varying ρ/τ and z/τ as a parameter (where $\tau = 1$ in this case). Note the excellent agreement, even in the vicinity of $\rho = \tau$, where the integral becomes singular as $z \to 0$. Hence, the combination of the DE quadrature and a partition method with convergence acceleration by extrapolation resulted in a powerful and efficient technique, applicable for a wide range of the parameters.

5. Application of the Ogata–Sugihara DE rule

The method of Ogata and Sugihara [128–130] which extends the earlier work of Ooura and Mori [131–133] has been successfully adopted for Sommerfeld integral tails by Golubović et al.. [15,41] Whereas the conventional DE transformations make the decay of the integrand double-exponential as the variable of integration approaches $\pm\infty$, the idea here is to make

the nodes of the quadrature formula approach the zeros of the Bessel function in (3) double-exponentially. In applying this method to the Sommerfeld tail

$$S = \int_a^\infty \tilde{G}(k_\rho; z|z') J_\nu(k_\rho \rho) k_\rho \, dk_\rho, \quad \rho > 0, \tag{86}$$

we first introduce the transformation [129]

$$k_\rho \rho = \varphi(t) = \frac{\pi t}{h} \tanh g(t) + c \, \text{sech} \, g(t), \quad g(t) = \frac{\pi}{2} \sinh t, \tag{87}$$

where $c = a\rho$ and $h > 0$ is a "mesh size" parameter, whose selection will be addressed in due course. The corresponding Jacobian is readily found as

$$\dot{\varphi}(t) = \frac{\pi}{h} \tanh g(t) + \left[\frac{\pi t}{h} - c \, \sinh g(t) \right] \text{sech}^2 g(t) \, \dot{g}(t), \quad \dot{g}(t) = \frac{\pi}{2} \cosh t, \tag{88}$$

and it clearly does not possess the rapid fall-off property of the more standard DE transformations of Section 2. The integral (86) may now be expressed as [41]

$$S = \int_0^\infty F[\varphi(t)] J_\nu[\varphi(t)] \dot{\varphi}(t) \, dt = \frac{1}{2} \int_{-\infty}^\infty \text{sgn} \, t \, F[\varphi(t)] J_\nu[\varphi(t)] \dot{\varphi}(t) \, dt, \tag{89}$$

where we have introduced

$$F(t) = \frac{t}{\rho^2} \tilde{G}\left(\frac{t}{\rho}; z|z'\right), \tag{90}$$

and where in the second integral we have used the fact that the integrand on the left is an odd function of t. The signum function is defined as

$$\text{sgn} \, t = \begin{cases} +1 & \text{if } t > 0, \\ 0 & \text{if } t = 0, \\ -1 & \text{if } t < 0. \end{cases}$$

For later reference, we note that $\varphi(0) = c$, $\dot{\varphi}(0) = 0$, and $\ddot{\varphi}(0) = (1 - hc/4)\pi^2/h$. The role of the sech part of the transformation (87) is to map the lower integral limit from $k_\rho = a$ into $t = 0$. The influence of this term vanishes quickly as $t \to \pm\infty$ and we find that $\varphi(t) \sim \pi|t|/h$ in those limits. Consequently, the integrand function in (89) has zeros that for large t approach the values

$$t_{\nu k} = \frac{h}{\pi} j_{\nu k} \tag{91}$$

double-exponentially, where $j_{\nu k}$ denotes the kth zero of the Bessel function of order ν. Hence, the second integral in (89) is amenable to the DE rule of Ogata and Sugihara [129], and we obtain

$$S \approx h \sum_{k=1}^n w_{\nu k} F[\varphi(t_{\nu k})] J_\nu[\varphi(t_{\nu k})] \dot{\varphi}(t_{\nu k}) + \delta_\nu, \tag{92}$$

where

$$w_{\nu k} = \frac{Y_\nu(j_{\nu k})}{J_{\nu+1}(j_{\nu k})} = \frac{2}{\pi j_{\nu k} J_{\nu+1}^2(j_{\nu k})} \tag{93}$$

and

$$\delta_\nu = \frac{h}{2\pi} \sum_{\lambda=0}^{\nu-1} C_\lambda^{(\nu)} \left(\frac{h}{\pi}\right)^{2\lambda+1} \frac{d^{2\lambda+1}}{dt^{2\lambda+1}} \{F[\varphi(t)] J_n[\varphi(t)] \dot{\varphi}(t)\} \Big|_{t=0}, \tag{94}$$

with

$$C_\lambda^{(\nu)} = \frac{1}{(2\lambda+1)!} \sum_{m=0}^{\nu-\lambda-1} \frac{(\nu-m-1)!}{m!} \, 2^{\nu-2m} \, b_{\nu-\lambda-m-1}^{(\nu)}, \qquad (95)$$

where $b_m^{(\nu)}$ are the coefficients in the Laurent series expansion of $1/J_\nu(z)$ around $z=0$,

$$\frac{1}{J_\nu(z)} = \sum_{m=0}^{\infty} b_m^{(\nu)} \, z^{2m-\nu}. \qquad (96)$$

In (93), Y_ν denotes the Neumann function and the second expression follows upon invoking the Wronskian of J_ν and Y_ν.[5, Equation 10.5.2] The zeros $j_{\nu k}$ in (91) and (93) can be obtained using available software packages [134] or may be computed by a few Newton-Raphson iterations, using the asymptotic values as starting points. From (94)–(96), we readily find that $\delta_0 = 0$ and

$$\delta_1 = \frac{h}{2}(4-hc)F(c)J_1(c), \qquad (97)$$

with the higher order δ_ν given by expressions rapidly increasing in complexity, which we omit to conserve space. In (97), we have used the fact that $b_0^{(1)} = 2$.

We still need to address the selection of the mesh size h and the truncation index n in (92). For Sommerfeld integral tails, Golubović et al.[41] recommend $h = 1/32$ as the best trade-off between accuracy and computational cost. However, since it is unlikely that this particular mesh size should be suitable for all integrand functions, an algorithm for its automatic selection is desirable. If $I(h)$ denotes the quadrature rule (92) with mesh size h, it can be shown that the discretization error σ is $\mathcal{O}(e^{-d/h})$, where d is a constant. Consequently, if the mesh size is halved, the error of $I(h/2)$ will be approximately $\sqrt{\sigma}$. Hence, we may compute a sequence of approximations $I(1/2), I(1/4), I(1/8), \ldots$, until $|I(h/2)-I(h)| < \sqrt{\epsilon}$, where ϵ is the desired error tolerance. Then, since the error of $I(h/2)$ will be approximately $\sqrt{\epsilon}$, we conclude that $\sigma \leq \epsilon$.[135] Although this adaptive strategy is similar to that implemented for the standard DE rules in Section 2, the Ogata–Sugihara quadrature is non-progressive, i.e. it cannot reuse the previously-computed function values each time h is halved, because the Bessel function zeros are not equidistant.

The truncation error, which depends on the truncation index n, should be kept smaller than the discretization error for any value of h. Unfortunately, it does not seem possible to derive a simple a priori formula for n satisfying this condition. A simple expedient adopted here is to continue the summation in (92) until the magnitude of the nth term falls below a small fraction, say 10^{-6}, of the already accumulated sum. We have found that double precision arithmetic (with approximately 15 significant digits) is not sufficient for some spectral domain Green functions $\tilde{G}$ that grow with k_ρ, in which case, when ρ is small, the magnitude of $J_\nu[\varphi(t_{\nu k})]$ may reach a floor value before the sum in (92) converges. On the other hand, for rapidly decreasing $\tilde{G}$, its magnitude may become exceedingly small even before the first zero of the Bessel function is reached, in which case a very small mesh size h may be required to achieve the desired accuracy of the quadrature. These caveats notwithstanding, the implementation of the Ogata–Sugihara rule (92) is straightforward, hence we omit the detailed algorithm for brevity. Examples of the application of the Ogata–Sugihara rule to Sommerfeld integral tails are given in the next section.
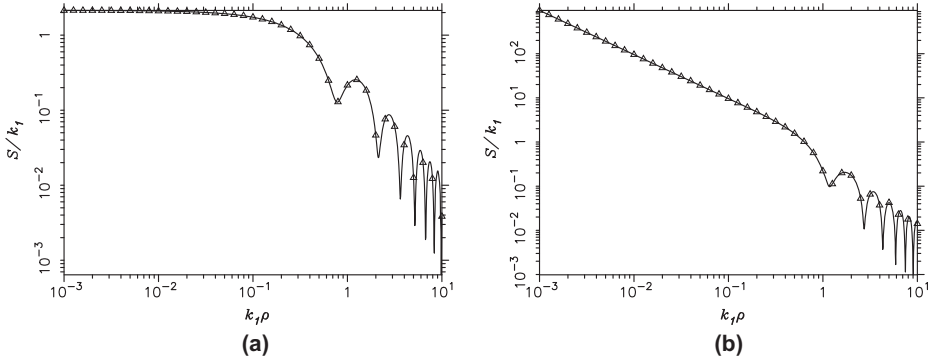
Figure 8. Sommerfeld tails integrated by the automatic Ogata–Sugihara quadrature (solid line) and by the PE method (triangles) for (a) the kernel given in (98) and (b) the kernel given in (99).

6. Examples of automatic Sommerfeld integral tail computation

We now illustrate the application of the methods described in this paper to the automatic computation of the Sommerfeld integral tail (89) with the kernels

$$\tilde{G}(k_\rho; z|z') = \frac{k_1}{jk_{z1}} \overleftarrow{\Gamma}_1^h \, e^{-jk_{z1}(z+z')} , \quad \nu = 0, \tag{98}$$

and

$$\tilde{G}(k_\rho; z|z') = \frac{k_1}{k_\rho} \left(\overleftarrow{\Gamma}_1^e - \overleftarrow{\Gamma}_1^h \right) e^{-jk_{z1}(z+z')}, \quad \nu = 1, \tag{99}$$

where

$$\overleftarrow{\Gamma}_1^h = -\frac{k_{z2}-k_{z1}}{k_{z2}-k_{z1}}, \quad \overleftarrow{\Gamma}_1^e = \frac{k_{z2}/\varepsilon_2 - k_{z1}}{k_{z2}/\varepsilon_2 - k_{z1}}, \tag{100}$$

with

$$k_{z1} = \sqrt{k_1^2 - k_\rho^2}, \quad k_{z2} = \sqrt{k_1^2 \varepsilon_2 - k_\rho^2}. \tag{101}$$

These spectral domain kernels arise in the computation of the horizontal and vertical components of the vector-potential Green functions due to a horizontal dipole radiating in air (medium 1) above a lossy halfspace (medium 2).[55] In the above, $\overleftarrow{\Gamma}_1^h$ and $\overleftarrow{\Gamma}_1^e$ are the reflection coefficients for TE and TM waves, respectively, looking into the lower halfspace, k_1 is the wavenumber of medium 1 and ε_2 is the dielectric constant of medium 2. The complex square roots with negative imaginary parts should be selected in (101), to satisfy the radiation conditions at infinity. In the numerical results that follow, we use $a = 2k_1$, $\zeta = z + z' = 0$, and $\varepsilon_2 = 10 - j18$, where the latter is a representative value for wet soil at 10 GHz. For the two kernels above, in Figure 8 we compare the results obtained by the automatic Ogata–Sugihara quadrature of Section 5 and the Levin-Sidi partition-extrapolation (PE) method of Section 4. The maximum discrepancy of the results is 0.82% and 0.12% in the cases of (98) and (99), respectively. In the case of (98), where $\tilde{G}$ is rapidly decreasing as a result of the vanishing $\overleftarrow{\Gamma}_1^h$ for $k_\rho \to \infty$, the mesh size h as small as $1/2^{14}$ was required in the Ogata–Sugihara rule near the lower $k_1\rho$ limit to satisfy the convergence criterion with $\epsilon = 10^{-6}$. This example illustrates that, in view of the non-progressive nature of the Ogata–Sugihara rule, many function evaluations can be wasted in some cases before the required quadrature level is reached. The PE method does not suffer from this drawback and should thus be preferred for automatic integration of Sommerfeld tails.

7. Conclusion

A review has been presented of the most effective methods and algorithms for the computation of Sommerfeld integral tails that commonly arise in the Green functions for plane-stratified media. The mathematical foundations of these methods are discussed in detail and their performance is illustrated by relevant numerical examples. In the most challenging case of oscillatory tails, the partition-extrapolation method is the recommended approach for efficient and reliable computation of such integrals. We demonstrate the implementation of this method as a fully automatic procedure and present the associated algorithms in pseudocode, which even a novice can readily convert into a working computer program. Hence, the computation of Sommerfeld integral tails should no longer be a "cauchemar" of engineers and scientists engaged in various areas of applied electromagnetics.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Krzysztof A. Michalski http://orcid.org/0000-0001-5753-7333

References

[1] Tyras G. Radiation and propagation of electromagnetic waves. New York (NY): Academic Press; 1969.
[2] Piessens R. Chapter 9, Hankel transforms. In: Poularikas AD, editor. Transforms and applications handbook. 3rd ed. Boca Raton (FL): CRC Press; 2010. p. 719–745.
[3] Michalski KA, Mosig JR. Multilayered media Green's functions in integral equation formulations (Invited review paper). IEEE Trans. Antennas Propag. 1997;45:508–519.
[4] Michalski KA. Electromagnetic field computation in planar multilayers. In: Chang K, editor. Encyclopedia of RF and microwave engineering. Vol. 2. Hoboken (NJ): Wiley-Interscience; 2005. p. 1163–1190.
[5] Olver FWJ, Lozier DW, Boisvert RF, Clark CW, editors. NIST Handbook of mathematical functions. New York (NY): Cambridge University Press; 2010.
[6] Singh NP, Mogi T. Electromagnetic response of a large circular loop source on a layered earth: a new computation method. Pure Appl. Geophys. 2005;162:181–200.
[7] Michalski KA. Spectral domain analysis of a circular nano-aperture illuminating a planar layered sample. Prog. Electromagn. Res. B. 2011;28:307–323.
[8] Michalski KA, Mosig JR. On the plane wave-excited subwavelength circular aperture in a thin perfectly conducting flat screen. IEEE Trans. Antennas Propag. 2014;62:2121–2129.
[9] Michalski KA, Mosig JR. Analysis of a plane wave-excited subwavelength circular aperture in a planar conducting screen illuminating a multilayer uniaxial sample. IEEE Trans. Antennas Propag. 2015;63:2054–2063.
[10] Gay-Balmaz P, Mosig JR. Three dimensional planar radiating structures in stratified media. Int. J. Microwave Millimeter-Wave Comput-Aided Eng. 1997;7:330–343.
[11] Bunger R, Arndt F. Efficient MPIE approach for the analysis of three-dimensional microstrip structures in layered media. IEEE Trans. Microwave Theory Tech. 1997;45:1141–1153; erratum, ibid., 1998;46:202.
[12] Michalski KA. Extrapolation methods for Sommerfeld integral tails (Invited review paper). IEEE Trans. Antennas Propag. 1998;46:1405–1418.
[13] Paulus M, Gay-Balmaz P, Martin OJF. Accurate and efficient computation of the Green's tensor for stratified media. Phys. Rev. E. 2000;62:5797–5807.

[14] Simsek E, Liu QH, Wei B. Singularity subtraction for evaluation of Green's functions for multilayer media. IEEE Trans. Microwave Theory Tech. 2006;54:216–225.

[15] Golubovic R, Polimeridis AG, Mosig JR. Efficient algorithms for computing Sommerfeld integral tails. IEEE Trans. Antennas Propagat. 2012;60:2409–2417.

[16] Koufogiannis ID, Mattes M, Mosig JR. On the development and evaluation of spatial-domain Green's functions for multilayered structures with conductive sheets. IEEE Trans. Microwave Theory Tech. 2015;63:20–29.

[17] Elhay S, Kautsky J. ALGORITHM 655 IQPACK: FORTRAN subroutines for the weights of interpolatory quadratures. ACM Trans. Math. Softw. 1987;13:399–415.

[18] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes: the art of scientific computing. 3rd ed. New York (NY): Cambridge University Press; 2007.

[19] Takahasi H, Mori M. Double exponential formulas for numerical integration. Publ. RIMS, Kyoto Univ. 1974;9:721–741.

[20] Evans GA, Forbes RC, Hyslop J. The tanh transformation for singular integrals. Int. J. Comput. Math. 1984;15:339–358.

[21] Mosig JR. Weighted averages and double exponential algorithms (Invited paper). IEICE Trans. Commun. 2013;E96-B:2355–2363.

[22] Johnson WA, Dudley DG. Real axis integration of Sommerfeld integrals: Source and observation points in air. Radio Sci. 1983;18:175–186.

[23] Katehi PB, Alexopoulos NG. Real axis integration of Sommerfeld integrals with applications to printed circuit antennas. J. Math. Phys. 1983;24:527–533.

[24] Mosig JR, Sarkar TK. Comparison of quasi-static and exact electromagnetic fields from a horizontal electric dipole above a lossy dielectric backed by an imperfect ground plane. IEEE Trans. Microwave Theory Tech. 1986;MTT-34:379–387.

[25] Tsai M, Chen C, Alexopoulos NG. Sommerfeld integrals in modeling interconnects and microstrip elements in multi-layered media. Electromagnetics. 1998;18:267–288.

[26] Mittra R, Lee SW. Analytical techniques in the theory of guided waves. New York (NY): Macmillan; 1971.

[27] Michalski KA, Butler CM. Evaluation of Sommerfeld integrals arising in the ground stake antenna problem. IEE Proc. Part H. 1987;134:93–97.

[28] Lambot S, Slob E, Vereecken H. Fast evaluation of zero-offset Green's function for layered media with application to ground-penetrating radar. Geophys. Res. Lett. 2007;34:L21405-1–6.

[29] Slob E, Lambot S. Chapter 15, Direct determination of electric permittivity and conductivity from air-launched GPR surface-reflection data. In: Miller RD, Bradford JH, Holliger K, editors. Advances in near-surface seismology and ground-penetrating radar. Tulsa (OK): Society of Exploration Geophysicists; 2010. p. 251–261.

[30] Jiménez E, Cabrera FJ, Cuevas del Río JG. Sommerfeld: a FORTRAN library for computing Sommerfeld integrals. In: Digest IEEE AP-S international symposium; July; Vol. 2; Baltimore (MD): 1996. p. 966–969

[31] Lytle RJ, Lager DL. Numerical evaluation of Sommerfeld integrals. Livermore (CA): Lawrence Livermore Laboratory; 1974. Rep. UCRL-51688.

[32] Tsang L, Brown R, Kong JA, et al. Numerical evaluation of electromagnetic fields dues to dipole antennas in the presence of stratified media. J. Geophys. Res. 1974;79:2077–2080.

[33] Bubenik DM. A practical method for the numerical evaluation of Sommerfeld integrals. IEEE Trans. Antennas Propag. 1977;AP-25:904–906.

[34] Kuo WC, Mei KK. Numerical approximations of the Sommerfeld integral for fast convergence. Rad. Sci. 1978;13:407–415.

[35] Kuester EF, Chang DC. Evaluation of Sommerfeld integrals associated with dipole sources above earth. Scientific electromagnet lab report 43. Boulder (CO): Department of Electrical Engineering, University of Colorado; 1979.

[36] Mohsen A. On the evaluation of Sommerfeld integrals. IEE Proc. Part H. 1982;129:177–182.

[37] Mosig JR, Gardiol FE. Analytic and numerical techniques in the Green's function treatment of microstrip antennas and scatterers. IEE Proc. Part H. 1983;130:175–182.

[38] Barkeshli S, Pathak PH. Radial propagation and steepest-descent path integral representations of the planar microstrip dyadic Green's function. Radio Sci. 1990;25:161–174.

[39] Dvorak SL. Application of the Fast Fourier Transform to the computation of the Sommerfeld integral for a vertical electric dipole above a half-space. IEEE Trans. Antennas Propag. 1992;40:798–805.

[40] Firouzeh ZH, Vandenbosch GAE, Moini R, et al. Efficient evaluation of Green's functions for lossy half-space problems. Prog. Electromagn. Res. 2010;109:139–157.

[41] Golubović Nićiforović R, Polimeridis AG, Mosig JR. Fast computation of Sommerfeld integral tails via direct integration based on double exponential-type quadrature formulas. IEEE Trans. Antennas Propag. 2011;59:694–699.

[42] Volskiy V, Vandenbosch GAE, Golubović Nićiforović R, et al. Numerical integration of Sommerfeld integrals based on singularity extraction techniques and double exponential-type quadrature formulas. In: Proceedings of 6th European conference on antennas and propagation (EUCAP); March; Prague, Czech Republic; 2012. p. 3215–3218.

[43] Chatterjee D, Rao SM, Kluskens MS. Improved evaluation of Sommerfeld integrals for microstrip antenna problems. In: Proceedings of 2013 URSI commission B international symposium on electromagnetic theory; May; Hiroshima, Japan; 2013. p. 981–984.

[44] Sainath K, Teixeira FL, Donderici B. Robust computation of dipole electromagnetic fields in arbitrarily anisotropic, planar-stratified medium. Phys. Rev. E. 2014;89:013312-1-18

[45] Karabult EP, Aksun MI. A novel approach for the efficient and accurate computation of Sommerfeld integral tails. In: Proceedings of international conference on electromagnetics in advanced applications (ICEAA); September; Torino, Italy; 2015. p. 646–649.

[46] Goldman M. Forward modelling for frequancy domain marine electromagnetic systems. Geophys. Prosp. 1987;35:1042–1064.

[47] Mosig JR, Melcón AA. Green's functions in lossy layered media: Integration along the imaginary axis and asymptotic behavior. IEEE Trans. Antennas Propag. 2003;51:3200–3208.

[48] Kourkoulos VN, Cangellaris AC. Accurate approximation of Green's functions in planar stratified media in terms of a finite sum of spherical and cylindrical waves. IEEE Trans. Antennas Propag. 2006;54:1568–1576.

[49] Yuan M, Sarkar TK. Computation of the Sommerfeld integral tails using the matrix pencil method. IEEE Trans. Antennas Propag. 2006;54:1358–1362.

[50] Alparslan A, Aksun MI, Michalski KA. Closed-form Green's functions in planar layered media for all ranges and materials. IEEE Trans. Microwave Theory Tech. 2010;58:602–613.

[51] Kurup DG. Analytical expressions for spatial domain Green's functions in layered media. IEEE Trans. Antennas Propag. 2015;63:1–7.

[52] Michalski KA. On the efficient evaluation of integrals arising in the Sommerfeld halfspace problem. In: Hansen RC, editor. Moment methods in antennas and scatterers. Boston: Artech House; 1990. p. 325–331.

[53] Wu B, Tsang L. Fast computation of layered medium Green's functions of multilayers and lossy media using fast all-modes method and numerical modified steepest descent path method. IEEE Trans. Microwave Theory Tech. 2008;56:1446–1454.

[54] Hochman A, Leviatan Y. A numerical methodology for efficient evaluation of 2D Sommerfeld integrals in the dielectric half-space problem. IEEE Trans. Antennas Propag. 2010;58:413–431.

[55] Michalski KA, Mosig JR. The Sommerfeld halfspace problem revisited: from radio frequencies and Zenneck waves to visible light and Fano modes (Invited review article). J. Electromagn. Waves Appl. 2016;30:142. Available from: http://dx.doi.org/10.1080/09205071.2015.1093964

[56] Markus A. Modern Fortran in practice. Cambridge (UK): Cambridge University Press; 2012.

[57] Schwarz HR. Numerical analysis, a comprehensive introduction. New York (NY): Wiley; 1989.

[58] Kythe PK, Schäferkotter MR. Handbook of computational methods for integration. Boca Raton (FL): Chapman & Hall/CRC Press; 2005.

[59] Goodwin ET. The evaluation of integrals of the form $\int_{-\infty}^{\infty} f(x)e^{-x^2}dx$. Proc. Cambridge Philos. Soc. 1949;45:241–245.

[60] Schwartz C. Numerical integration of analytic functions. J. Comp. Phys. 1969;4:19–29.

[61] Evans G. Practical numerical integration. New York (NY): Wiley; 1993.

[62] Krommer AR, Ueberhuber CW. Computational integration. Philadelphia (PA): SIAM; 1998.

[63] Trefethen LN, Weideman JAC. The exponentially convergent trapezoidal rule. SIAM Rev. 2014;56:385–458.

[64] Stenger F. Integration formulae based on the trapezoidal formula. J. Inst. Math. Appl. 1973;12:103–114.

[65] Takahasi H, Mori M. Quadrature formulas obtained by variable transformation. Numer. Math. 1973;21:206–219.

[66] Squire W. A quadrature method for finite intervals. Int. J. Numer. Methods Eng. 1976;10:708–712.

[67] Haber S. The tanh rule for numerical integration. SIAM J. Numer. Anal. 1977;14:668–685.

[68] Mori M. Quadrature formulas obtained by variable transformation and the DE-rule. J. Comput. Appl. Math. 1985;12&13:119–130.

[69] Mori M. Discovery of the double exponential transformation and its developments. Publ RIMS, Kyoto Univ. 2005;41:897–935.

[70] Bailey DH, Jeyabalan K, Li XS. A comparison of three high-precision quadrature schemes. Exp. Math. 2005;3:317–329.

[71] Bailey DH, Borwein JM. Effective error bounds in Euler-Maclaurin-based quadrature schemes. In: Proceedings of 20th international symposium on high-performance computing in an advanced collaborative environment (HPCS'06); May; St John's (NL), Canada: IEEE Computer Society; 2006. p. 1–7.

[72] Koufogiannis ID, Polimeridis AG, Mattes M, et al. Real axis integration of Sommerfeld integrals with error estimation. In: Proceedings of 6th European conference on antennas and propagation (EUCAP); March; Prague, Czech Republic; 2012. p. 719–723.

[73] Borwein JM, Bailey DH, Girgensohn R. Experimentation in mathematics, computational path to discovery. Natick (MA): A. K. Peters; 2004.

[74] Ooura T. DE-quadrature package [2006 December]. Available from: http://www.kurims.kyoto-u.ac.jp/ooura/intde.html.

[75] Ye L. Numerical quadrature: theory and computation [master's thesis]. Halifax, Nova Scotia: Dalhousie University, Faculty of Computer Science; 2006.

[76] Squire W. An efficient iterative method for numerical evaluation of integrals over a semi-infinite range. Int. J. Numer. Methods Eng. 1976;10:478–484.

[77] Brezinski C, Redivo Zaglia M. Extrapolation methods, theory and practice. New York (NY): North-Holland; 1991.

[78] Weniger EJ. Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. Comput. Phys. Rep. 1989;10:189–371.

[79] Wimp J. Sequence transformations and their applications. New York (NY): Academic Press; 1981.

[80] Levin D. Development of non-linear transformations for improving convergence of sequences. Int. J. Comput. Math., Sect. B. 1973;3:371–388.

[81] Gradshteyn IS, Ryzhik IM. Table of integrals, series, and products. 7th ed. New York (NY): Academic Press; 2007.

[82] Salzer HE. A simple method for summing certain slowly convergent series. J. Math. Phys. 1955;33:356–359.

[83] Laurie D. Appendix A: convergence acceleration. In: Bornemann F, Laurie D, Wagon S, Waldvogel J, editors. The SIAM 100-digit challenge. Philadelphia (PA): SIAM; 2004. p. 225–258.

[84] Smith DA, Ford WF. Acceleration of linear and logarithmic convergence. SIAM J. Numer. Anal. 1979;16:223–240.

[85] Homeier HHH. Scalar Levin-type sequence transformations. J. Comput. Appl. Math. 2000;122:81–147.

[86] Smith DA, Ford WF. Numerical comparison of nonlinear convergence accelerators. Math. Comput. 1982;38:481–499.

[87] Fessler T, Ford WF, Smith DA. HURRY: an acceleration algorithm for scalar sequence and series. ACM Trans. Math. Software. 1983;9:346–354.

[88] Sidi A. An algorithm for a special case of a generalization of the Richardson extrapolation process. Numer. Math. 1982;38:299–307.

[89] Davis PJ, Rabinowitz P. Methods of numerical integration. 2nd ed. New York (NY): Academic Press; 1984.

[90] Sidi A. Practical extrapolation methods. Theory and applications. Cambridge: Cambridge University Press; 2003.

[91] Shanks D. Non-linear transformations of divergent and slowly convergent sequences. J. Math. Phys. 1955;34:1–42.

[92] Wynn P. On a device for computing the $e_m(S_n)$ transformations. Math. Tables Aids Comput. 1956;10:91–96.

[93] Piessens R, deDoncker-Kapenga E, Überhauer CW, et al. QUADPACK: a subroutine package for automatic integration. New York (NY): Springer-Verlag; 1983.

[94] Bhowmic S, Bhattacharya R, Roy D. Iterations of convergence accelerating nonlinear transforms. Comput. Phys. Commun. 1989;54:31–46.

[95] Homeier HHH. A hierarchically consistent, iterative sequence transformation. Numer. Algorithms. 1994;8:47–81.

[96] Mosig JR, Gardiol FE, Hawkes PW, editors. A dynamical radiation model for microstrip structures. Vol. 59, Advances in electronics and electron physics. New York (NY): Academic Press; 1982. p. 139–237

[97] Mosig JR. Integral equation technique. In: Itoh T, editor. Numerical techniques for microwave and millimeter-wave passive structures. New York (NY): Wiley; 1989. p. 133–213.

[98] Drummond JE. Convergence speeding, convergence and summability. J. Comput. Appl. Math. 1984;11:145–159.

[99] Scraton RE. A note on the summation of divergent power series. Proc. Cambridge Philos. Soc. 1969;66:109–114.

[100] Wynn P. A note on the generalized Euler transformation. Comput. J. 1971;14:437–441.

[101] Scraton RE. The generalized Euler transformation. Comput. J. 1972;15:258.

[102] Longman IM. Note on a method for computing infinite integrals of oscillatory functions. Proc. Cambridge Philos. Soc. 1956;52:764–768.

[103] Goodwin ET, editor. Modern computing methods. Philosophical Library: New York (NY); 1961.

[104] Polimeridis AG, Golubović Nićiforović RM, Mosig JR. Acceleration of slowly convergent series via the generalized weighted-averages method. Prog. Electromagn. Res. M. 2010;14:233–245.

[105] Van Deun J, Cools R. Integrating products of Bessel functions with an additional exponential or rational factor. Comput. Phys. Commun. 2008;178:578–590.

[106] Golubović R, Polimeridis AG, Mosig JR. The weighted averages method for semi-infinite range integrals involving products of Bessel functions. IEEE Trans. Antennas Propag. 2013;61:5589–5596.

[107] Krogh FT, Snyder WV. Algorithm 699: A new representation of Patterson's quadrature formulae. ACM Trans. Math. Softw. 1991;17:457–460.

[108] Trefethen LN, Weideman JAC. Is Gauss quadrature better than Clenshaw-Curtis? SIAM Rev. 2008;50:67–87.

[109] Alaylioglu A, Evans GA, Hyslop J. The evaluation of oscillatory integrals with infinite limits. J. Comp. Phys. 1973;13:433–438.

[110] Squire W. Partition-extrapolation methods for numerical quadrature. Int. J. Comput. Math. Sect. B. 1975;55:81–91.

[111] Sidi A. The numerical evaluation of very oscillatory infinite integrals by extrapolation. Math. Comput. 1982;38:517–529.

[112] Chave AD. Numerical integration of related Hankel transforms by quadrature and continued fraction expansion. Geophysics. 1983;48:1671–1686.

[113] Lyness JN. Integrating some infinite oscillating tails. J. Comput. Appl. Math. 1985;12–13:109–117.

[114] Sidi A. A user-friendly extrapolation method for oscillatory infinite integrals. Math. Comput. 1988;51:249–266.

[115] Rabinowitz P. Extrapolation methods in numerical integration. Numer. Algorithms. 1992;3:17–28.

[116] Espelid TO, Overholt KJ. DQAINF: An algorithm for automatic integration of infinite oscillating tails. Numer. Algorithms. 1994;8:83–101.

[117] Lucas SK, Stone HA. Evaluating infinite integrals involving Bessel functions of arbitrary order. J. Comput. Appl. Math. 1995;64:217–231.

[118] Sauter T. Computation of irregularly oscillating integrals. Appl. Numer. Math. 2000;35:245–264.

[119] Slevinsky RM, Safouhi H. A comparative study of numerical steepest descent, extrapolation, and sequence transformation methods in computing semi-infinite integrals. Numer. Algorithms. 2012;60:315–337.

[120] Mosig JR. The weighted averages algorithm revisited. IEEE Trans. Antennas Propag. 2012;60:2011–2018.

[121] Temme NM. An algorithm with ALGOL 60 program for the computation of the zeros of ordinary Bessel functions and those of their derivatives. J. Comput. Phys. 1979;32:270–279.

[122] Hasegawa T, Sidi A. An automatic integration procedure for infinite range integrals involving oscillatory kernels. Numer. Algorithms. 1996;13:1–19.

[123] Abramowitz M, Stegun IA, editors. Handbook of Mathematical Functions. New York (NY): Dover; 1965.

[124] Bailey DH, Borwein JM. Hand-to-hand combat with thousand-digit integrals. J. Comput. Sci. 2012;3:77–86.

[125] Sidi A. A user-friendly extrapolation method for computing infinite range integrals of products of oscillatory functions. IMA J. Numer. Anal. 2012;32:602–631.

[126] Lucas SK. Evaluating infinite integrals involving products of Bessel functions of arbitrary order. J. Comput. Appl. Math. 1995;64:269–282.

[127] Ratnanather JT, Kim JH, Zhang S. Algorithm 935: IIPBF, a MATLAB toolbox for infinite integral of products of two Bessel functions. ACM Trans. Math. Softw. 2014;40:14.1–14.12.

[128] Ogata H, Sugihara M. Interpolation and quadrature formulae whose abscissae are the zeros of the Bessel functions (in Japanese). Trans. Jpn. Soc. Indus. Appl. Math. 1996;6:39–66.

[129] Ogata H, Sugihara M. Quadrature formulae for oscillatory infinite integrals involving the Bessel functions (in Japanese). Trans. Jpn. Soc. Indus. Appl. Math. 1998;8:223–256.

[130] Ogata H. A numerical integration formula based on the Bessel functions. Publ. RIMS, Kyoto Univ. 2005;41:949–970.

[131] Ooura T, Mori M. Double exponential formulas for oscillatory functions over the half infinite interval. J. Comput. Appl. Math. 1991;38:353–360.

[132] Mori M, Ooura T. Double exponential formulas for Fourier type integrals with a divergent integrand. In: Agarwal RP, editor. Contributions in numerical mathematics. New York (NY): World Scientific; 1993. p. 301–308.

[133] Ooura T, Mori M. A robust double exponential formula for Fourier-type integrals. J. Comput. Appl. Math. 1999;112:229–241.

[134] Vrahatis MN, Ragos O, Skiniotis T, et al. RFSFNS: A portable package for the numerical determination of the number and the calculation of roots of Bessel functions. Comput. Phys. Commun. 1995;92:252–266.

[135] Ogata H. Private communication. 2015.