Fourier transforms of slowly converging functions exemplified by electromagnetic wave propagation in evanescent structures

1999 J. Phys. A: Math. Gen. 32 8173

(http://iopscience.iop.org/0305-4470/32/46/313)

J. Phys. A: Math. Gen. **32** (1999) 8173–8183. Printed in the UK PII: S0305-4470(99)07140-1

Fourier transforms of slowly converging functions exemplified by electromagnetic wave propagation in evanescent structures

Thilo Sauter

Vienna University of Technology, Institute of Computer Technology, Gußhausstraße 27/E384, A-1040 Vienna, Austria

E-mail: sauter@ict.tuwien.ac.at

Received 23 August 1999

Abstract. Fourier-type integrals often balk at numerical evaluation with simple quadrature algorithms. A suitable strategy to cope with slowly decaying oscillating integrands over unbounded integration intervals is to subdivide the interval and extrapolate the sequence of partial sums. This paper, supported by numerical examples, presents guidelines for the choice of the partition points. It will be shown that the first subdivision point must be selected with particular care in order to obtain reliable extrapolation results. As a practical example, we explore the propagation of an electromagnetic wavefront in a dispersive, evanescent medium, which should—despite recent speculations on superluminal signal transmission—travel with exactly the speed of light. It appears that the partition extrapolation strategy correctly computes the behaviour of the wave, whereas other methods fail to give satisfying answers. What is particularly appealing about the proposed method is that it requires only moderate analysis of the integrand and can be composed from standard numerical algorithms.

1. Introduction

Many problems in computational physics require the evaluation of Fourier transforms of the type $\int_{-\infty}^{\infty} g(\omega)\exp(\mathrm{i}\omega)\,\mathrm{d}\omega$. If appropriate, such integrals are often split at $\omega = 0$, so that the exponential functions of the two parts can be combined to a sine or cosine function. In fact, standard Fourier integrals are a special case of a much wider class of semi-infinite integrals with oscillatory integrands. In a fairly general form, such integrals can be written as

$$I = \int_a^{\infty} g(x)\mathrm{e}^{\mathrm{i}p(x)}\,\mathrm{d}x \tag{1}$$

where the complex exponential function may, of course, equally well be replaced by its real or imaginary part. In ordinary Fourier integrals, $g(x)$ is a comparatively smooth spectral density function and the phase function $p(x)$ is linear. This is, however, only a special case. When we consider wave propagation phenomena, for example, we encounter phase functions of the type

$$p(\omega) = k(\omega)z - \omega t \tag{2}$$

which may be anything but linear. Here $k(\omega)$ denotes the dispersion relation in the medium, i.e. the relation of wavenumber and the angular frequency. The integration variable is ω, whereas t and z are the time and spatial coordinates. In non-dispersive media, which are characterized by a constant phase velocity, $k(\omega)$ remains linear. If the medium is dispersive, however, such that the phase velocity depends upon the frequency, the dispersion relation is nonlinear.

0305-4470/99/468173+11$30.00

In most practical cases, it is impossible to find analytical solutions to integrals of the form (1). Therefore, one must resort to numerical techniques. Unfortunately, the oscillatory nature of the integrand prevents a straightforward and unthinking application of standard quadrature schemes.

The problem is alleviated if the function $g(x)$ decays fast enough (i.e. exponentially) to justify a truncation of the integration interval. Many methods have been devised to tackle the quadrature of oscillating functions over finite intervals. They mostly rely on transformation techniques [1] or approximation of the integrand with simple functions [2–4]. To achieve the desired accuracy of the result, they frequently also employ adaptive subdivision strategies [2, 4, 5].

When the decay of the integrand is too slow to permit truncation, a suitable strategy is to subdivide the interval and compute a series of partial integrals that are alternating in sign,

$$I_i = \int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \qquad I = \sum_{i=0}^{\infty} I_i \tag{3}$$

with $x_0 = a$. The choice of the subdivision points is essentially a matter of taste. For the sake of convenience and provided we have a real-valued integrand, most authors choose the zeros of the integrand. From the partial integrals, we can form a sequence of partial sums,

$$S_i = \sum_{j=0}^{i} I_j \tag{4}$$

which then oscillates about the exact value of the integral. Since this sequence usually converges only slowly, its limit can be determined numerically by extrapolation. Quite a number of such partition extrapolation methods [6] have been developed. Unfortunately, most of them are restricted to cases where the integrand has equidistant zeros at least for $x \to \infty$, which is tantamount to an asymptotically linear phase function $p(x)$. Lyness [7] considered integrals of the type $\int_a^\infty f(x)\mathrm{d}x$ with $f(x)$ slowly decaying and oscillating. The zeros are required to be asymptotically equidistant as in $f(x) = g(x)j(x)$, where $j(x)$ is a circular or Bessel function and $g(x)$ is positive over the entire integration interval. The resulting series is then accelerated with the Euler transformation, which is very efficient as long as the sequence $\{I_i\}$ is alternating in sign. This idea was later used by Lyness and Hines [8] as the basis of a quadrature routine, and it was further extended by Espelid and Overholt [9]. Finally, the QUADPACK package [10] comprises several widely used quadrature routines tailored to Fourier transforms. Therefore, the integrands must have the form $g(x)\sin\omega x$ or $g(x)\cos\omega x$. Integration is carried out between the zeros of the oscillating factor using a Clenshaw–Curtis scheme with adaptive subdivision, so that there is practically no restriction with respect to the function $g(x)$. The sequence of partial sums is accelerated with the ϵ-algorithm [11].

It is surprising to see that all computer routines available in standard software libraries are only capable of integrating functions with an ultimately constant period. There are, however, also cases where the integrand has increasingly rapid oscillations for $x \to \infty$. Electron-wave dispersion is an example for such a behaviour. Sidi [12] was one of the few who addressed this problem. He also used subdivision at the zeros of the integrand and developed an algorithm, the W-transformation, that can accelerate the convergence of the sequence of partial sums. More importantly, he proved that there is no need to determine the zeros of the integrand exactly, which can be a computationally expensive procedure. Instead, it is sufficient to regard the polynomial part of the phase function to create an asymptotic partition, which will eventually also produce an alternating series. However, this algorithm has not found its way into common collections of computer programs.

2. How to partition the interval

The strategy of partitioning the integration interval and using convergence acceleration to find the limit of the sequence of partial sums is so intuitive that we can readily apply it even without having specialized computer routines at hand. Taking successive zeros as subdivision points, we obtain handy pieces that can be treated with any quadrature scheme suitable for smooth integrands. Instead of the zeros, we can also use the extrema to break up the integration range. Which points we choose is actually of very little significance [8, 9]. What matters is that the resulting sequence $\{I_i\}$ is alternating. We can even go one step further and calculate the partition points only asymptotically [12]. If the phase function $p(x)$ has an asymptotic expansion

$$p(x) \sim x^{\gamma} \sum_{i=0}^{\infty} \frac{a_i}{x^i} \tag{5}$$

we can extract its polynomial part,

$$\bar{p}(x) = \sum_{i=0}^{\gamma} a_i x^{\gamma - i}. \tag{6}$$

The idea behind this is that for very large x, $p(x)$ behaves like $\bar{p}(x)$. In the limit $x \to \infty$, both functions are identical. Consequently, the zeros of $\sin p(x)$ approach those of $\sin \bar{p}(x)$, and in either case the sequence of partial integrals (3) is at least ultimately alternating. The sequence of partial sums (4) is then treated with any extrapolation algorithm that can handle alternating sequences. The benefit of an asymptotic partition is that the computation of the partition points is reduced to the simpler problem of solving a polynomial equation. This can significantly improve the performance of the entire computation.

The most crucial part of the partitioning strategy is the choice of the sequence member S_I up to which the summation is carried out explicitly and where the extrapolation process is started. In most cases and for the sake of simplicity, this will be the first subdivision point x_1. To underline the importance of a proper choice, we shall examine a simple example. Consider the integral

$$\int_0^{\infty} \frac{1}{x+1} \sin\left(x + 4\sqrt{(x-25)^2 + 1}\right) \mathrm{d}x. \tag{7}$$

Figure 1 shows the phase function together with all solutions of the equation $p(x) = n\pi$, which mark the zeros of the integrand. If we subdivide the integration range at these points and set up the sequence of partial sums, we obtain the result shown in figure 2. The pronounced step in the sequence stems, of course, from the minimum of $p(x)$ at $x = 25$. In the vicinity of this point, the distance between the zeros increases, and the respective partial integral makes an excessive contribution to the sequence of partial sums. The fact that a region where $p'(x) = 0$ dominates the value of the integral is, in a way, nothing but a manifestation of the well known method of stationary phase. Consequently, to compute the correct limit of the sequence, we must ensure that we start the extrapolation *beyond* the extremum of $p(x)$.

To avoid these obvious problems, Sidi suggested selecting x_1 as the first zero of $\sin \bar{p}(x)$ greater than the lower integration limit a. Thus x_1 satisfies the polynomial equation $\bar{p}(x) = n\pi$ for some integer n. The subsequent partition points x_i, $i > 1$, are then determined to be the largest positive solution of $\bar{p}(x) = (n + i - 1)\pi$ or $\bar{p}(x) = (n - i + 1)\pi$, depending on whether $p(x)$ is increasing or decreasing. While the selection of x_1 works well for many problems, it would fail in our particular case because the polynomial part of the phase function is $\bar{p}(x) = 5x - 100$, which is the right asymptote of the hyperbola. Since $\bar{p}(x)$ is a linear

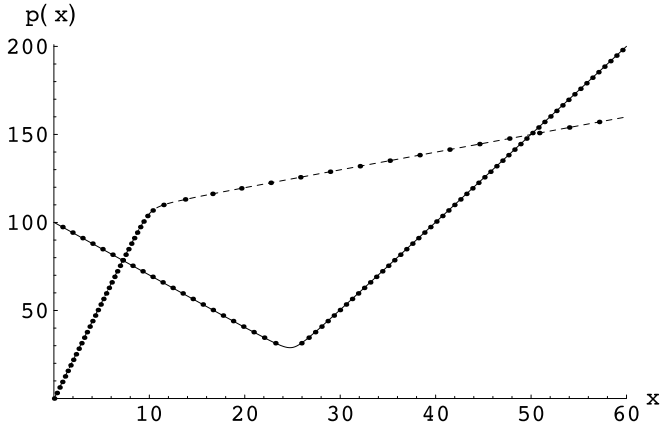

Figure 1. Phase functions $p(x) = x+4\sqrt{(x-25)^2+1}$ (——) and $p(x) = 6x-5\sqrt{(x-10)^2+1}+50$ (- - - -) together with partition points.

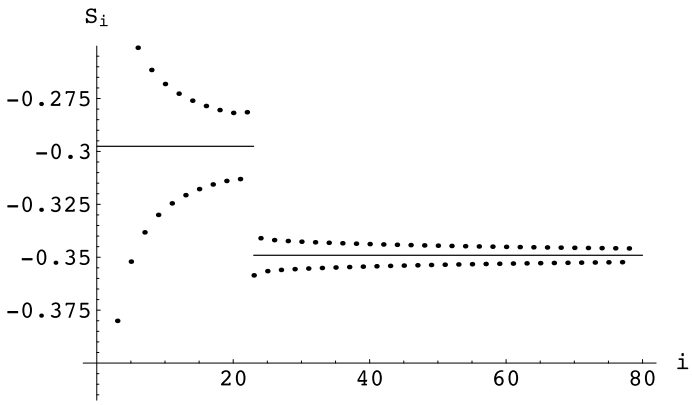

Figure 2. Sequence of partial sums for (7) based on the partition points shown in figure 1.

function, we would obtain a comparable set of partition points as the one given in figure 1 with $x_1 \approx 0$. Hence we would probably not detect the existence of the extremum and would get a completely wrong extrapolation result.

The simplest way out of this dilemma is to hinge the partition points on the original phase function $p(x)$. With the same algorithm as before, we find $x_1 \approx 40$ and run no risk. The only deficiency is that the interval $[x_0, x_1]$ for the first integral becomes unnecessarily large, which will affect the computing time. Yet another potential problem is lurking in Sidi's approach, as the next example will demonstrate. If we change $p(x)$ in (7) only slightly,

$$\int_0^\infty \frac{1}{x+1} \sin\left(6x - 5\sqrt{(x-10)^2+1} + 50\right) \mathrm{d}x \tag{8}$$

we get a phase function that has no extremum, but a change of its slope at $x = 10$, as figure 1 shows. Partitioning the integrand at its zeros, we obtain the sequence of partial sums depicted in figure 3. Despite the absence of extrema in $p(x)$, there is a discontinuity in the sequence. Like before, the sequence seems to converge on either side. Hence any extrapolation algorithm will find a computationally reliable limit when being fed with the first or the last few sequence members. As the figure also shows, these limits are different, and the only way to pick the

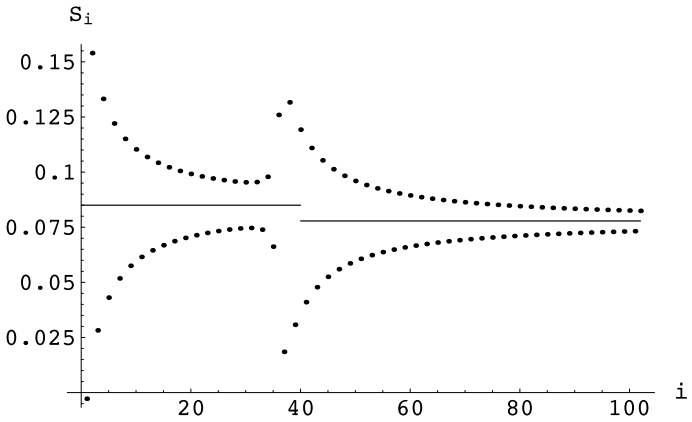

Figure 3. Sequence of partial sums for the integral (8) based on the subdivision shown in figure 1.

right one is to start the extrapolation late enough.

Needless to emphasize that the aforementioned problems are not restricted to hyperbolic phase functions. They apply equally well to polynomial phase functions or any other nonlinear function with extrema or sharp bends. Based on the examples and considerations given, we suggest the following heuristic algorithm for the partitioning of the integration interval:

- Determine all solutions of $p'(x) = 0$. If they can be found analytically, only the real part is of interest. Then take the maximum of this set,

$$x_m = \max\{\mathrm{Re}\, x_i\} | p'(x_i) = 0. \tag{9}$$

  This ensures that the first partition point is greater than any extremum or saddle point of the phase function. If $p'(x) = 0$ has no solution, the condition for inflexion points, $p''(x) = 0$, must be taken instead. For polynomial $p(x)$, these two criteria are already sufficient. If the phase function is more general (e.g. hyperbolic), inflexion points may not exist either. Then it is reasonable to look for extrema in the second derivative, i.e. $p'''(x) = 0$, which mark persistent changes of the slope of $p(x)$.
- Let x_1 be the first zero of $\sin p(x)$ greater than x_m and the lower integration limit a. Then x_1 is a solution of $p(x_1) = n\pi$ for some integer n.
- The subsequent partition points $\{x_i\}$, $i > 1$ are then found by solving the equation

$$\begin{aligned} p(x) &= (n + i - 1)\pi \qquad \text{for} \quad \lim_{x\to\infty} p(x) = \infty \\ p(x) &= (n - i + 1)\pi \qquad \text{for} \quad \lim_{x\to\infty} p(x) = -\infty. \end{aligned} \tag{10}$$

  If the equation has more than one solution for a given i, the largest solution shall be taken.

The algorithm for calculating the subdivision points given above is based on the original phase function. We can, however, apply it to the polynomial approximation $\bar{p}(x)$ of the phase function as well, which yields an asymptotic partition. In this case, the determination of x_m would be easier. However, to select the first subdivision point, it is still imperative to analyse $p(x)$ itself, since the approximation might be too coarse (as in our earlier example). The calculation of the further subdivision points can then safely be carried out with $\bar{p}(x)$.

So far, we have only considered the behaviour of the integrand's oscillating factor. Another source of troubles that may lead the extrapolation algorithm astray is the non-oscillating factor $g(x)$ in (1), which should vary only slowly compared with the oscillations. In particular,

changes of the sign of $g(x)$ can be harmful if they occur abruptly. The smoother $g(x)$ is, the less pronounced is this effect. So, in order to select the first partition point properly, we must apply the same considerations to $g(x)$ as to $p(x)$.

3. Thoughts on efficient computation of the integrals

The selection of an appropriate partition is a prerequisite for a reliable extrapolation result. It exerts only a minor influence on the computational performance of the overall algorithm. The way we compute the individual integrals is much more relevant in this respect. We should thus take a closer look at possible quadrature methods. The integrals $I_i = \int_{x_i}^{x_{i+1}} f(x)\mathrm{d}x$ to the right of the first partition point x_1 are computed between successive zeros and therefore have smooth integrands with roughly the shape of half a sine wave. The computation of these partial integrals is thus no problem, and nearly any quadrature algorithm will yield satisfying results. A scheme that is particularly suited to such well-behaved integrands is the Gauss rule. It has the additional benefit that it is available in many shades and in many program libraries.

While the partial integrals I_i for $i > 1$ look fairly similar for all possible integrands, the integration range of the first integral $I_0 = \int_a^{x_1} f(x)\mathrm{d}x$ may comprise more than just two zeros of the integrand. In fact, the integrand can be either smooth or strongly oscillating, which might depend also on additional parameters. In wave propagation problems, the choice of z and t in (2) can change the behaviour of the integrand completely. In principle, the most appropriate quadrature algorithm for the first integral could be selected based on an *a priori* investigation of the integrand. This is, however, a viable strategy for a single evaluation of an individual integrand and is not really appropriate for parameter-dependent integrals. It is thus preferable to seek a robust method able to cope with a large variety of functions, in particular with oscillatory ones. Such methods often transform the integrand prior to quadrature. One example is the double-exponential rule, which has been quite successful in numerical tests.

A performance comparison of several quadrature methods for a particular example will be given in the next section. As far as the extrapolation is concerned, any algorithm suitable for a convergence acceleration of alternating series can be applied. A good choice is, for example, the ϵ-algorithm [11]. Being a general-purpose algorithm, it has fair acceleration properties and can be found in many implementations.

4. A practical example: the propagation of a wavefront

Let us now apply the results of the previous sections to a real physical example. There has been a fierce controversy in the past few years about experiments that seemed to prove superluminal signal transmission. Enders and Nimtz [13] studied the propagation of evanescent modes in a hollow waveguide and found that the maximum of a Gaussian pulse traverses this 'tunnel' in almost no time. Their conclusion was that signal transmission faster than light is possible, which sparked a still ongoing debate [14, 15]. Gaussian pulses are very popular in theory, the only problem with this concept is that they have an infinite duration and are thus inappropriate for the transmission of information. It seems therefore reasonable to study the behaviour of a pulse with rectangular envelope, which is the prototype building block of all digital signals.

As a medium we choose a lossless electron plasma. We then have the same dispersion relation as with a hollow waveguide, but we can restrict our analysis to TEM waves. This makes life a little bit easier since we have to take care of only one spatial component. A practical example for this one-dimensional model is an electromagnetic transmission line with an appropriate dispersive dielectric. If we consider this line as infinitely long or properly

terminated, we can even neglect reflections. The wave at any point along the line can be described by a superposition of monochromatic plane waves,

$$u(x,t) = \int_{-\infty}^{\infty} U_0(\omega) \mathrm{e}^{\mathrm{i}(\omega t - k(\omega)x)}\,\mathrm{d}\omega \tag{11}$$

which is in fact a Fourier integral with $U_0(\omega)$ being the spectrum of the pulse at $x = 0$. The dispersion relation of the medium is given by

$$k(\omega) = \frac{1}{c}\sqrt{\omega^2 - {\omega_p}^2} \tag{12}$$

with the velocity of light c and the plasma frequency ω_p. Below this cutoff frequency, the wavenumber $k(\omega)$ becomes imaginary, and we have evanescence, which prohibits normal wave propagation and yields exponential attenuation with total, non-dissipative reflection of the signal's energy. To examine this behaviour, we set the excitation at $x = 0$ to be a carrier below cutoff ($\omega_0 < \omega_p$) being switched on at $t = 0$ and off at $t = \tau$,

$$u(0,t) = \begin{cases} \cos(\omega_0 t) & \text{if} \quad 0 \leqslant t \leqslant \tau \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

which means that we have steps in the signal. Since the medium is causal, we expect these wavefronts to travel through the plasma at exactly the speed of light [16]. The spectrum of the initial pulse is found to be

$$\begin{aligned} U_0(\omega) &= \frac{1}{2\pi}\int_0^{\tau} \cos(\omega_0 t)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}t \\ &= \frac{1}{2\pi}\left[\frac{\sin\frac{\tau}{2}(\omega-\omega_0)}{\omega-\omega_0}\mathrm{e}^{-\mathrm{i}(\omega-\omega_0)\frac{\tau}{2}} + \frac{\sin\frac{\tau}{2}(\omega+\omega_0)}{\omega+\omega_0}\mathrm{e}^{-\mathrm{i}(\omega+\omega_0)\frac{\tau}{2}}\right]. \end{aligned} \tag{14}$$

When we now assemble (11) to determine the evolution of the wave, we must be careful with the expression of the dispersion relation in the individual frequency ranges. Only if we set

$$k(\omega) = \begin{cases} -\dfrac{1}{c}\sqrt{\omega^2 - {\omega_p}^2} & \text{if} \quad \omega \leqslant -\omega_p \\ -\dfrac{\mathrm{i}}{c}\sqrt{\omega^2 - {\omega_p}^2} & \text{if} \quad |\omega| \leqslant \omega_p \\ \dfrac{1}{c}\sqrt{\omega^2 - {\omega_p}^2} & \text{if} \quad \omega \geqslant \omega_p \end{cases} \tag{15}$$

do we obtain correct results, i.e. a real-valued system response and attenuation for the evanescent modes with increasing x. Without loss of generality, we also set $\tau\omega_0 = 2n\pi$, so that the input signal lasts exactly for an integer multiple n of periods, which allows for a convenient simplification of the Fourier integral. Taking everything together, we obtain for (11)

$$\begin{aligned} u(x,t) = \frac{2}{\pi}\int_0^{\omega_p} \frac{\omega}{\omega^2 - {\omega_0}^2}\mathrm{e}^{-\frac{x}{c}\sqrt{{\omega_p}^2-\omega^2}}\sin(n\pi\omega/\omega_0)\cos(\omega t - n\pi\omega/\omega_0)\,\mathrm{d}\omega \\ + \frac{2}{\pi}\int_{\omega_p}^{\infty} \frac{\omega}{\omega^2 - {\omega_0}^2}\sin(n\pi\omega/\omega_0)\cos\left(\omega t - \frac{x}{c}\sqrt{\omega^2 - {\omega_p}^2} - n\pi\omega/\omega_0\right)\mathrm{d}\omega \end{aligned} \tag{16}$$

where we have split the frequency range in the evanescent part and the pass band. To prepare the integrals for numerical treatment, it is advisable to introduce normalized variables. We set

Table 1. Computing times in seconds needed for the evaluation of (17) at 1051 points with different quadrature algorithms on a PC with a 300 MHz Pentium II processor. The ϵ-algorithm was used for extrapolation.

	Gauss–Kronrod	Double exponential	Subintervals
Finite part of (17)	293	113	1
Infinite integral in (17)			
truncation	5670	2035	1
extrapolation of (17)	829	2283	120
extrapolation of (18)	598	2084	28

$\xi = \omega/\omega_p$ for the integration variable, $\Omega = \omega_0/\omega_p$ for the carrier frequency, and $X = \omega_p x/c$ and $T = \omega_p t$ for the space and time coordinates, respectively. Then the Fourier integral reads

$$u(X,T) = \frac{2}{\pi}\int_0^1 \frac{\xi}{\xi^2-\Omega^2} \mathrm{e}^{-X\sqrt{1-\xi^2}} \sin(n\pi\xi/\Omega)\cos(\xi T - n\pi\xi/\Omega)\,\mathrm{d}\xi + \frac{2}{\pi}\int_1^\infty \frac{\xi}{\xi^2-\Omega^2} \sin(n\pi\xi/\Omega)\cos(\xi T - X\sqrt{\xi^2-1} - n\pi\xi/\Omega)\,\mathrm{d}\xi. \tag{17}$$

The first integral is finite and can be computed directly. For the numerical results given below, we tried several algorithms. The performance results are given in table 1. As could have been expected, the double-exponential rule needs the least computing time. The evaluation of the second integral, however, is much more interesting. Note that the spectrum converges only like $1/\xi$. Since the integrand is a product of two oscillating functions, it is sound to rewrite the integral as

$$u_2(X,T) = \frac{1}{\pi}\int_1^\infty \frac{\xi}{\xi^2-\Omega^2} \sin(\xi T - X\sqrt{\xi^2-1})\,\mathrm{d}\xi + \frac{1}{\pi}\int_1^\infty \frac{\xi}{\xi^2-\Omega^2} \sin(\xi T - X\sqrt{\xi^2-1} - 2n\pi\xi/\Omega)\,\mathrm{d}\xi. \tag{18}$$

The integrals now have the structure given in (1) with $p(\xi) = \xi T - X\sqrt{\xi^2-1}$ and $p(\xi) = \xi T - X\sqrt{\xi^2-1} - 2n\pi\xi/\Omega$, which actually describes hyperbolas. For large ξ, the oscillations have a nearly constant period, although the frequency strongly depends on the parameters X and T. Therefore, we could also use equidistant subdivision points, rather than calculating the zeros explicitly. The choice of the first subdivision point is uncritical except for the case when in the first integral, X is slightly smaller than T. The same applies for the second integral when X is slightly smaller than $T - 2n\pi/\Omega$. Under these circumstances, distinct minima appear in $p(\xi)$ that must be considered. We thus have the same situation as in section 2 and must be wary of steps in the sequence of partial sums. Particular care is also required when $X = T$, because then the first integral in (18) is no longer ultimately oscillating, and an extrapolation method is not applicable. In this special case, a different treatment is needed, as for instance the application of the double-exponential rule.

To examine the propagation of the signal, we compute the turn-on of the wave at $X = 10$ for the parameters $\Omega = 0.2$ and $n = 2$. To achieve a sufficient resolution, we evaluate the integrals at 1051 points in the interval $[9.5, 11.6]$. The results are shown in figure 4. We see that the wavefront indeed arrives at $T = X$, which in our normalized variables means a propagation velocity equal to the speed of light. The high-pass filter characteristic of the plasma has turned the wavefront into a sharp needle, but its height remains undistorted due to the absence of losses. Before the arrival of the pulse, the medium is completely at rest, which is also correctly reflected in the numerical results.

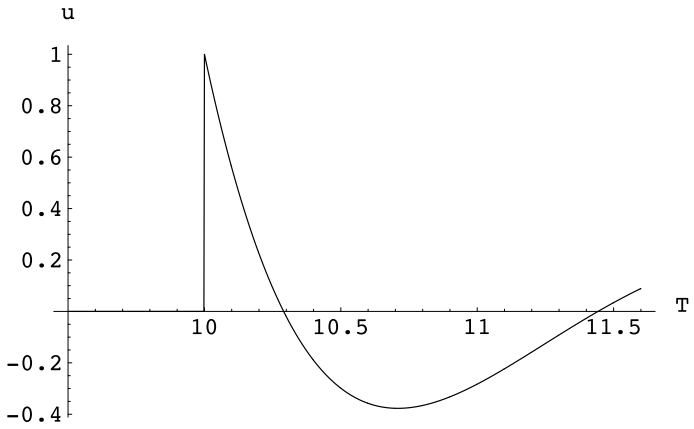

Figure 4. Numerical results of (17) for $\Omega = 0.2$, $n = 2$, and $X = 10$ obtained with the partition extrapolation approach applied to (18).

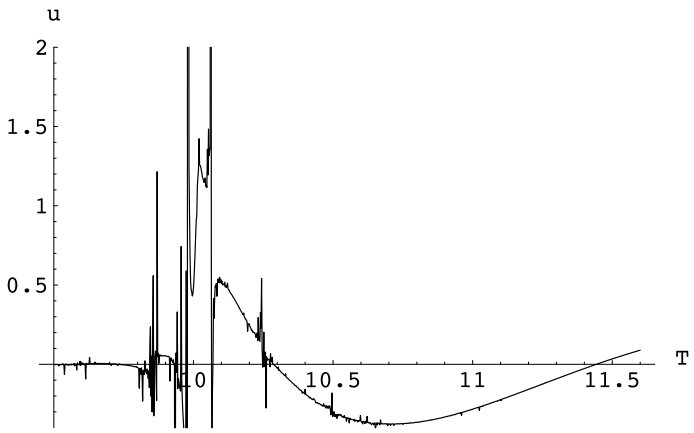

Figure 5. Numerical results of (17) for $\Omega = 0.2$, $n = 2$, and $X = 10$ obtained with a QUADPACK-like partition extrapolation strategy.

For comparison, we also compute (17) using two other strategies. For the first one we do not split the second integral in (17) but integrate between the zeros of the factor $\sin(n\pi\xi/\Omega)$ instead. The resulting sequence of partial sums is then extrapolated as usual. This is the basic idea of the QUADPACK algorithms [10]. For our experiment, however, we use the same partition extrapolation implementation as above with a different integrand and phase function. The outcome of the computation is shown in figure 5. We see that the result is particularly bad around the wavefront. A glance at (18) reveals the reason. When X is about equal to T, we no longer have a high-frequency oscillation modulated by a low-frequency one, as (17) would suggest. Rather, the integrand is composed from a superposition of a high- and a very low-frequency oscillation. Consequently, the sequence of partial sums does not oscillate symmetrically about its limit, which makes it unsuitable for extrapolation. As X and T grow more different, the integrand shows a more 'wavepacket'-like behaviour, and extrapolation becomes easier and more reliable. Hence, this strategy is applicable only for parts of the wave away from the wavefront.

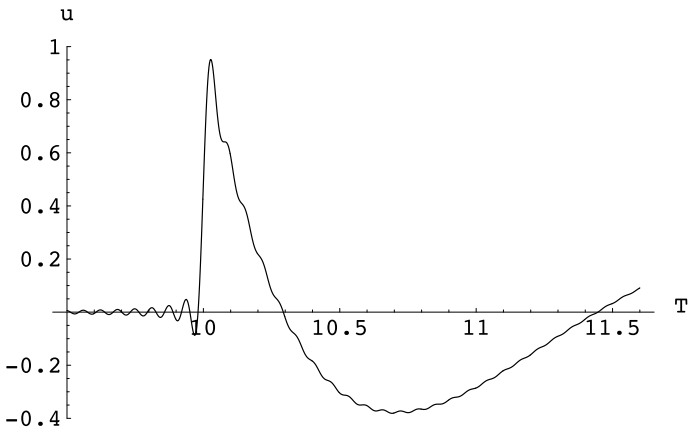

Figure 6. Numerical results of (17) for $\Omega = 0.2$, $n = 2$, and $X = 10$ obtained with direct computation and truncation of the integration interval at $\xi = 100$.

The second alternative we investigate is the conventional truncation method. Here we also compute the original integral (17) directly. To minimize the error, we truncate the integration range at a zero of $\sin(n\pi\xi/\Omega)$. Unfortunately, as figure 6 shows, this does not help. Just around the wavefront, the values are completely wrong. Obviously the truncation at $\xi = 100$ is still far too coarse for the slowly decaying integrand. What makes the situation particularly evil is the 'smooth' look of the signal. While we would surely suspect numerical problems in figure 5 (provided we chose a sufficiently fine resolution), we could be led to accept the outcome in figure 6 because of its appearance. In this case, we would have to interpret the oscillations ahead of the wavefront as superluminal precursors, although they are actually a numerical artifact. From a different viewpoint, truncating the integration interval comes down to limiting the spectrum of the signal, and what we get is nothing but the well known Gibb's phenomenon.

Table 1 gives a performance summary of the numerical algorithms implemented in Mathematica [17]. The computations were carried out on a Pentium II processor, and a series of trials on a Sun workstation yielded comparable results. The absolute figures are not so important, it is their relations that reveal several remarkable properties. First of all, we note that for single integrals comprising a large number of zeros of the integrand, an algorithm that transforms the integrand prior to quadrature (such as the double-exponential rule) is superior to a simple Gauss scheme. The latter is, on the other hand, unbeatable if we integrate between successive zeros. Hence the results confirm the suggestions of the previous section. We also see that the QUADPACK-like extrapolation approach is only slightly slower than the 'proper' extrapolation strategy due to the larger number of subintervals (i.e. members of the sequence of partial sums). Had we taken the same number as with the split integrand in (18), we would have been considerably faster, which is not surprising as there would have been only one integral to compute and not two. Alas, this brings no advantage, since the result is unacceptable anyway, and it does not improve even if we take more sequence members into account. The last lesson we learn from table 1 is the also not astonishing fact that the truncation approach is bedevilled by extremely long computing times. The only chance to ameliorate the results of figure 6 is to carry the integration further than $\xi = 100$. This in turn would linearly increase the computing time. Obviously, the truncation strategy is completely inappropriate for our example.

5. Conclusions

The numerical computation of semi-infinite Fourier-type integrals often poses problems if the integrand is slowly decaying, so that a simple truncation of the integration interval is impossible without introducing significant errors. For such cases, it is advisable to partition the integration interval and determine the limit of the sequence of partial sums by means of extrapolation. Although this is essentially a straightforward method, the subdivision points must be chosen with care. Fortunately, this requires only a simple analysis of the oscillating factor. We presented criteria for the partition strategy and the evaluation of the partial integrals.

As a practical example, we considered a wave propagation problem, namely the motion of a TEM shock wave in a lossless plasma. The associated Fourier integrals are both nonlinearly oscillating and slowly decaying, which calls for a partition extrapolation method. Since this investigation was stimulated by the recent discussion on superluminal signal transmission, a proper evaluation of the signal around the wavefront was of eminent importance. By means of the proposed method, we could verify that the wavefront moves exactly with the speed of light, which is in accordance with electrodynamic theory. We also found that other, less appropriate computation methods are unable to give correct results and may even lead to false interpretations.

As an additional benefit, it should be noted that the method presented here does not require the implementation of new and sophisticated numerical algorithms. The entire numerical work can be accomplished by standard algorithms to be taken from standard software libraries. So all that is actually needed is a reasonable combination of readily available building blocks.

Acknowledgment

I am grateful to Fritz Paschke for many discussions and suggestions.

References

[1] Levin D 1982 *Math. Comput.* **38** 531
[2] Xu P and Mal A 1985 *Wave Motion* **7** 235
[3] Ehrenmark U T 1988 *J. Comput. Appl. Math.* **21** 87
[4] Evans G A 1994 *Appl. Numer. Math.* **14** 383
[5] Haider Q and Liu L C 1992 *J. Phys. A: Math. Gen.* **25** 6755
[6] Rabinowitz P 1992 *Numer. Algorithms* **3** 17
[7] Lyness J N 1985 *J. Comput. Appl. Math.* **12&13** 109
[8] Lyness J and Hines G 1986 *ACM Trans. Math. Softw.* **12** 24
[9] Espelid T O and Overholt K J 1994 *Numer. Algorithms* **8** 83
[10] Piessens R, de Doncker-Kapenga E, Überhuber C W and Kahaner D K 1983 *QUADPACK, A Subroutine Package for Automatic Integration* (Berlin: Springer)
[11] Davis P J and Rabinowitz P 1984 *Methods of Numerical Integration* (Orlando, FL: Academic)
[12] Sidi A 1988 *Math. Comput.* **51** 249
[13] Enders A and Nimtz G 1993 *Phys. Rev.* E **48** 632
[14] Thoma P and Weiland T 1994 *Phys. Bl.* **50** 313
[15] Nimtz G and Heitmann W 1997 *Prog. Quantum Electron.* **21** 81
[16] Diener G 1996 *Phys. Lett.* A **223** 327
[17] Wolfram S 1991 *Mathematica, A System for Doing Mathematics by Computer* (Reading, MA: Addison-Wesley)