ELSEVIER

Computer Physics Communications 145 (2002) 256–266

Computer Physics Communications

www.elsevier.com/locate/cpc

Numerical evaluation of Hilbert transforms for oscillatory functions: A convergence accelerator approach

Frederick W. King, Gregory J. Smethells, Geir T. Helleloid, Paul J. Pelzl

Department of Chemistry, University of Wisconsin-Eau Claire, Eau Claire, WI 54702, USA

Received 18 September 2001; accepted 4 January 2002

Abstract

The numerical evaluation of Hilbert transforms on the real line for functions that exhibit oscillatory behavior is investigated. A fairly robust numerical procedure is developed that is based on the use of convergence accelerator techniques. Several different types of oscillatory behavior are examined that can be successfully treated by the approach given. A few examples of functions whose oscillations are too extreme to deal with are also discussed.

Keywords: Hilbert transforms; Oscillatory functions; Numerical quadrature; Convergence acceleration

PACS: 02-60.Jh; 02-30.Uu; 02-60.-x

1. Introduction

A great deal of effort has been devoted to the numerical evaluation of singular integrals in general, and the Hilbert transform in particular [1–44]. Much of this work has been driven by the wide occurrence of Hilbert transforms in a diverse array of problems of considerable significance in the physical sciences, engineering, and applied mathematics [45–55]. In the present work, we are concerned with the numerical evaluation of Hilbert transforms on the real line for functions exhibiting oscillatory behavior. Some of the simplest examples are the sine and cosine functions, which occur widely in applications in the applied sciences and in engineering problems.

The Hilbert transform of f, denoted Hf, is defined by

$$(Hf)(x_0) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{f(x)\,\mathrm{d}x}{x_0 - x}. \tag{1}$$

In Eq. (1) P designates the Cauchy principal value, that is

$$(Hf)(x_0) = \frac{1}{\pi} \lim_{\varepsilon\to 0+} \left\{ \int_{-\infty}^{x_0-\varepsilon} \frac{f(x)\,\mathrm{d}x}{x_0 - x} + \int_{x_0+\varepsilon}^{\infty} \frac{f(x)\,\mathrm{d}x}{x_0 - x} \right\}. \tag{2}$$

We note that the Hilbert transform is occasionally defined with the opposite sign to that given in Eq. (1). It is well known that for a function f satisfying $f \in L^p(\mathbb{R})$ for $1 < p < \infty$, the Hilbert transform of f satisfies

0010-4655/02/$ – see front matter
PII: S0010-4655(02)00155-8

$Hf \in L^p(\mathbb{R})$ [56,57]. For the case $p = 1$, Hf exists almost everywhere, but in general is not integrable. The focus, either explicitly or implicitly, of most of the published literature on the numerical evaluation of Hilbert transforms and other singular integrals covers functions belonging to the class L^p (for $p \geqslant 1$). Some of the simple cases that we consider do not fall within the class $L^p(\mathbb{R})$, for example, the choice $f(x) = \sin ax$, with a denoting a constant, gives $\int_{-\infty}^{\infty} |\sin ax|^p \,\mathrm{d}x = \infty$ for $p > 0$.

2. Numerical evaluation technique

The methodology we present for numerically evaluating the Hilbert transform of a function with oscillatory behavior consists of breaking up the region of integration into three parts:

$$\frac{P}{\pi}\int_{-\infty}^{\infty}\frac{f(x)}{x_0 - x}\,\mathrm{d}x = \frac{1}{\pi}\int_{-\infty}^{x_0-\varepsilon}\frac{f(x)}{x_0 - x}\,\mathrm{d}x + \frac{P}{\pi}\int_{x_0-\varepsilon}^{x_0+\varepsilon}\frac{f(x)}{x_0 - x}\,\mathrm{d}x + \frac{1}{\pi}\int_{x_0+\varepsilon}^{\infty}\frac{f(x)}{x_0 - x}\,\mathrm{d}x. \tag{3}$$

The second integral on the right-hand side of Eq. (3) can be recast as

$$\frac{P}{\pi}\int_{x_0-\varepsilon}^{x_0+\varepsilon}\frac{f(x)}{x_0 - x}\,\mathrm{d}x = \frac{1}{\pi}\lim_{\tau\to 0+}\int_{\tau}^{\varepsilon}\frac{\{f(x_0-t) - f(x_0+t)\}\,\mathrm{d}t}{t} + \frac{1}{\pi}\left\{\int_{\varepsilon}^{x_0+\varepsilon}\frac{f(x_0-t)\,\mathrm{d}t}{t} - \int_{\varepsilon}^{\varepsilon-x_0}\frac{f(x_0+t)\,\mathrm{d}t}{t}\right\}. \tag{4}$$

In Eq. (4) the last two integrals are not singular since $x_0 \in (-\varepsilon, \varepsilon)$. We impose the restriction that the function f is Hölder continuous on the interval of integration $(-\varepsilon, \varepsilon)$ with exponent α, with $\alpha > 0$. That is

$$\left|f(x_0 - t) - f(x_0 + t)\right| \leqslant C|t|^{\alpha}, \quad \text{for } \alpha > 0. \tag{5}$$

In this case we can write

$$\frac{1}{\pi}\lim_{\tau\to 0+}\int_{\tau}^{\varepsilon}\frac{\{f(x_0-t) - f(x_0+t)\}\,\mathrm{d}t}{t} = \frac{1}{\pi}\int_{0}^{\varepsilon}\frac{\{f(x_0-t) - f(x_0+t)\}\,\mathrm{d}t}{t}. \tag{6}$$

The last integral is no longer a Cauchy principal value integral. The idea just outlined is well known. Sloan [3] applied it in the context of the numerical evaluation of the finite Hilbert transform with a singular point at the origin.

We have used *Mathematica* [58] to carry out the basic numerical evaluation approach. The principal value integral is evaluated directly, using the *CauchyPrincipalValue* object in *Mathematica*, over a small interval $(x_0 - \varepsilon, x_0 + \varepsilon)$, where we have chosen ε to be 10^{-3} in our implementation. This particular choice yielded high accuracy for most of the test functions examined. *Mathematica* employs the approach indicated in Eq. (4), and then applies a Gauss–Kronrod procedure—an adaptive Gaussian quadrature approach to handle the resulting integrals.

There is potential for significant loss of precision in the evaluation of Eq. (6) as $t \to 0+$. If the rounding errors for the evaluation of $f(x_0 - t) - f(x_0 + t)$ are sizable, they become magnified by the term t^{-1} as $t \to 0+$. It is therefore essential to have appropriate precision available for the evaluation of $f(x_0 - t) - f(x_0 + t)$. *Mathematica* is ideally suited for this calculation.

When the function is *not* Hölder continuous in the vicinity of the singular point, then the approach of the present work is likely to meet with limited success, or fail completely. Cases where the singularity is embedded in a region of extreme oscillatory behavior are much more difficult to deal with using a general robust algorithm.

The integrals over $(-\infty, x_0 - \varepsilon)$ and $(x_0 + \varepsilon, \infty)$ are separately partitioned into sets of integrals, for which the specific details are described below. Each integral in a set is treated as a term of a slowly converging infinite series.

Convergence acceleration techniques are then applied to obtain the desired precision. The details of the approach are now addressed.

The numerical integration over $(-\infty, x-\varepsilon)$ is achieved by first determining the oscillatory characteristics of the function in the integrand, $h(x_0, x)$, where

$$h(x_0, x) = \frac{f(x)}{\pi(x_0 - x)}. \tag{7}$$

In the following discussion we assume some fixed value of x_0 is selected and abbreviate $h(x_0, x)$ by $h(x)$. A trace is made over $h'(x)$. Initializing two variables, one to $x_0 - \varepsilon$ and one to $x_0 - \varepsilon - \delta$, where δ denotes a step-size, we scan in the direction of negative infinity, stepping by $-\delta$. Any changes in the sign of $h'(x)$ across this interval causes a call to the built-in *Mathematica* object *FindRoot*, which in turn uses Newton's method to determine the root more precisely. A maximum of two roots are gathered in this manner and the search stops once these two roots are found or if either variable goes beyond $(x_0 - \varepsilon - \Delta)$, where Δ is a hand-tuned constant acting as an upper-limit on the range of our trace. Our implementation has tuned $|\delta|$ to 10^{-1} and $|\Delta|$ to 100, based upon the general characteristics of the functions under investigation, although these values could easily be parameterized in the code. We also note that when available, *Mathematica* precision and accuracy specific options are used to aid in obtaining a user-specified precision level, $\mathcal{P}$. Specifically, the *Mathematica* options *AccuracyGoal* and *PrecisionGoal* are set to one and a half times $\mathcal{P}$ and *WorkingPrecision* is set to four times $\mathcal{P}$. These have proven to be sufficient to obtain the requested precision level based on our empirical experience. *Mathematica* objects we utilize that accept these options include *FindRoot*, *NIntegrate*, and *CauchyPrincipalValue*.

Three outcomes from our trace are possible: (a) no roots are detected, (b) one root is found, or (c) two roots are found. Although all the integrands of interest are oscillatory in nature, it is feasible that the separation between roots could be larger in magnitude than any chosen fixed value $|\Delta|$, hence the necessity of cases (a) and (b) simply for reasons of robustness in the code. An auxiliary function $g(x)$ is introduced to describe the partitioning scheme for the integration of $h(x)$. For cases (a) and (b), we take $g(x) = \sin x$, which leads to evenly spaced partition points and the desired robustness in the application of our algorithm. However, for the vast majority of integrals of interest, case (c) occurs. In this case, a check is made to ascertain whether $h(x)$ evaluates to zero at either of the two roots, and if so, $g(x)$ is taken to be $h'(x)$. This resolves the case where $h(x)$ oscillates, touching but not crossing the x-axis. When $h(x)$ does oscillate across the x-axis, $g(x)$ is taken to be $h(x)$.

Once $g(x)$ is known, a set of partition points can be determined. For cases (a), (b), and also (c), when $h(x)$ oscillates across the x-axis, the partition points are taken to be the roots of $g(x)$. However, for case (c) when $h(x)$ does *not* oscillate across the x-axis, we take the partition points to be the roots of $g(x)$ that correspond to where $h(x)$ touches the x-axis. The net result, after a trace over $g(x)$ is made, is a set of intervals over each individual oscillation in $h(x)$.

Using a similar tracing procedure to that described above, a set of n roots $\{\rho_1, \rho_2, \rho_3, \ldots, \rho_n\}$ is gathered from $g(x)$. Only one small change is made to the tracing algorithm: after two or more roots have been found, δ is set equal to the difference between adjacent roots times a small, hand-tuned fractional value. This has the advantage of permitting our trace to discover roots more closely spaced than the original δ and to dynamically adapt to any given oscillatory function. Next, we numerically integrate using the *Mathematica* object *NIntegrate*, between successive roots in the set we gathered, and obtain a set of values $\{v_1, v_2, v_3, \ldots, v_{n-1}\}$.

As these individual values may or may not represent terms of a monotonic slowly converging infinite series, we cannot treat every v_i as a term in a sequence to be summed by convergence acceleration techniques. Rather, we begin to sum the values:

$$V_m = \sum_{i=1}^{m} v_i, \tag{8}$$

while $|v_i| < |v_{i+1}|$, searching for a peak value that can be used as a valid initial term of a non-increasing series. V_m is eventually added to the result of the convergence acceleration of the sub-series $\{v_{m+1}, v_{m+2}, \ldots, v_{n-1}\}$. Once the initial value v_{m+1} is determined, partial sums are calculated,

$$S_k = \sum_{j=1}^{k} v_j, \quad \text{for } k = 1, 2, \ldots, n-1, \tag{9}$$

where v_{m+1} now plays the role of the first term, and the series is convergence accelerated to obtain a result for the integral over $(-\infty, \rho_1)$. References that we have found to be valuable for their discussion of convergence acceleration techniques and applications are [59–73]. Our implementation uses a Levin u transformation [61] to perform this acceleration:

$$u_k = \frac{\sum_{j=0}^{k}(-1)^j \binom{k}{j}(j+1)^{k-2}\frac{S_{j+1}}{A_{j+1}}}{\sum_{j=0}^{k}(-1)^j \binom{k}{j}(j+1)^{k-2}\frac{1}{A_{j+1}}}, \tag{10}$$

where $\binom{k}{j}$ is a binomial coefficient, A_j denotes the jth term in the series to be summed, and S_j designates the jth partial sum, that is,

$$S_j = \sum_{n=1}^{j} A_n. \tag{11}$$

Certain situations exist where alternative transformations give rise to improved results. Overall, the Levin u transformation has shown itself to be quite adept at accelerating the convergence of a wide range of series, including alternating and monotonic series alike. In special instances, however, it may be advantageous to use such non-linear sequence transformations as the Wynn ρ algorithm [62], one of two transformations presented by Weniger [66–68], or the Levin t transform [61].

Lastly, any residual interval $(\rho_1, x_0 - \varepsilon)$ is numerically integrated using *NIntegrate* and added to the result of the aforementioned convergence acceleration. In this manner, we obtain a result for the first integral on the right-hand side of Eq. (3). A similar approach can be applied to the integral in Eq. (3) covering the interval $(x_0 + \varepsilon, \infty)$. The only change necessary in the procedure is the obvious one of switching the direction of the trace—towards positive infinity, from $x_0 + \varepsilon$. As discussed previously, the center integral about the principal value point was done directly using the *Mathematica* object *CauchyPrincipalValue*. The latter calculation is straightforward, provided severe oscillation does not occur in very close proximity to the singular point. The summation of all three results provides a numeric answer to the Hilbert transform in Eq. (1). A copy of the code will be available at http://www.chem.uwec.edu/king/hilbert.

There is a key item about this technique that deserves special mention for the case of alternating series, which arise from an integrand which oscillates across the axis. This is handled very effectively by the choice of a Levin u transform for the convergence accelerator. The alternation of sign from the Levin u transform cancels out that of the alternating series to provide a monotonic acceleration that avoids possible numerical instabilities.

3. Test functions

We have picked test functions to illustrate the technique just described that satisfy the following basic requirements:

(a) $(Hf)(x)$ does not diverge for all x, and
(b) $(Hf)(x)$ can be evaluated in analytic form.

Table 1
A comparison of numerical quadrature values for the Hilbert transform versus exact evaluation for several oscillatory functions. The values $x = 2$, $a = 5$, and $b = 3$ have been employed

$f(x)$	$\frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{f(s)\,ds}{x-s}$ (exact result)	Numerical quadrature result	From the exact result
$\sin ax$	$-\operatorname{sgn} a \cos ax$	0.839071529076452	0.839071529076452
$\cos ax$	$\operatorname{sgn} a \sin ax$	−0.544021110889369	−0.544021110889369
$\frac{\sin^2(ax)}{x^2}$	$\frac{\operatorname{sgn} a\{2ax - \sin 2ax\}}{2x^2}$	2.385881843659046	2.385881843659046
$\cos \pi x\, e^{-x^2}$	$e^{-x^2} \operatorname{Im}\{e^{i\pi x} \operatorname{erf}(\frac{\pi}{2} + ix)\}$	0.013915590535066	0.013915590535066
$\operatorname{sinc} x$	$\frac{1-\cos \pi x}{\pi x}$	6.27595×10^{-34}	0
$\sin ax^2$ $(a > 0)$	$-\operatorname{sgn} x\{S(\sqrt{2a/\pi}\lvert x\rvert)[\cos ax^2 + \sin ax^2]$ $+C(\sqrt{2a/\pi}\lvert x\rvert)[\cos ax^2 - \sin ax^2]\}$	−0.3168301	−0.316829655318096
$\frac{\sin 2abx}{\sin bx}$ $(a > 0,$ $b > 0)$	$\frac{2\sin^2(abx)}{\sin(bx)}$	−6.98717	−6.987489931661428
$\cos x^{-1}$	$-\sin x^{-1}$	−0.479369	−0.479425538604203

Condition (b) has been selected for the obvious reason that it allows an important check to be made on the quality of the results from the numerical evaluations.

When Hilbert transforms of the form $H\{fg\}(x)$ are encountered, where f exhibits oscillatory behavior and g is a continuous function with a suitable asymptotic behavior as $x \to \pm\infty$, then a viable computational approach is to try the technique described above on Hf. This assumes that Hf does not diverge, and further assumes that it is possible to evaluate Hf analytically, so the quality of the numerical result can be evaluated. Then, depending on the smoothness properties of g, a reasonable assumption is that if Hf can be successfully evaluated numerically, the same would probably be true for $H\{fg\}(x)$. We note that when f exhibits oscillatory behavior, there are a very large number of examples that can be worked out in analytic form, however the inclusion of the function g for even some fairly simple choices, can very quickly make the solution of $H\{fg\}(x)$ in terms of standard special functions a very difficult if not impossible assignment. Series techniques may be a feasible evaluation strategy in such cases.

The specific test functions selected fall into the following categories:

(i) Simple oscillatory behavior as exemplified by the sine and cosine functions.
(ii) Simple oscillatory behavior multiplied by a slowly decaying algebraic function, for example, $x^{-1}\sin x$ and $x^{-2}\sin^2 x$.
(iii) Simple oscillatory behavior superimposed on an exponential decaying function, for example, $\cos x e^{-x^2}$.
(iv) Functions with more complex oscillatory behavior, for example, $\sin x^2$ and $(\sin bx)^{-1}\sin 2abx$, for a and b constants.
(v) Functions with extreme oscillatory behavior, for example, $\cos x^{-1}$ and $\cos(\cot x)$.

For some of these choices it is obviously necessary to pay particular attention to the behavior of the function in the vicinity of the origin, to avoid divergent Hilbert transforms. For example, the choice $f(x) = x^{-1}\cos x$ leads to a Dirac delta distribution function contribution for Hf, and such cases have been excluded in the present treatment.

4. Results

A selection of results are presented in Table 1. In the table entries the evaluation point x_0 has been simplified to x. As indicated above, we set the working precision at four times the requested precision required for the final answer. This particular scaling of the working precision could be decreased for most of the examples we examined, as the precision obtained for the numerical results exceeded what we have reported in Table 1. The following notation has been employed in the table. The Fresnel cosine and sine integrals are defined, respectively, by

$$C(z) = \int_0^z \cos\left(\frac{\pi t^2}{2}\right) \mathrm{d}t, \tag{12}$$

and

$$S(z) = \int_0^z \sin\left(\frac{\pi t^2}{2}\right) \mathrm{d}t. \tag{13}$$

Im denotes the imaginary part, erf designates the error function, defined by

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \mathrm{e}^{-s^2}\, \mathrm{d}s, \tag{14}$$

sgn denotes the signum function (sign function) given by

$$\operatorname{sgn} x = \begin{cases} 1 & \text{for } x > 0, \\ 0 & \text{for } x = 0, \\ -1 & \text{for } x < 0 \end{cases} \tag{15}$$

and the sinc function is defined by

$$\operatorname{sinc} x = \frac{\sin \pi x}{\pi x}. \tag{16}$$

Two functions that we have studied in detail are $f(x) = \cos(x^{-1})$ and $f(x) = \cos(\cot x)$. We comment in the following section on the probable reasons for the limited precision obtained for functions of this type.

5. Discussion

In Figs. 1 and 2 we show plots for the integrands $(\sin^2 5x)/(\pi x^2(2-x))$ and $(\sin 5x)/(\pi(2-x))$. The former function has a maximum magnitude of oscillatory behavior at the origin, with decreasing oscillatory behavior near the singular point at $x_0 = 2$, where there is a switch over in the sign for the oscillatory behavior. The technique described in Section 2 handles functions with this type of behavior rather well. For the second function, there is major oscillatory behavior in the vicinity of the singular point, and the oscillations die off fairly uniformly on both sides of the singularity. Again, our technique handles this situation well. The alternating sign that arises with this example is not a problem, for the reason we have addressed in Section 2.

A more difficult case is represented by the choice of functions $(\sin 2abx)/(\sin bx)$. A plot of $(\sin 2abx)/(\pi(2-x)\sin bx)$ for $a = 5$ and $b = 3$ is shown in Fig. 3. In this example we have major oscillations centered around the singular point, but as Fig. 3 clearly illustrates, there is a non-monotonic decay of the oscillatory behavior away from the singular point. The technique we employed handles this situation, but the precision obtained is not as high as that determined for the other examples reported in Table 1. Further refinements in the partitioning technique employed coupled with a different selection of convergence accelerator techniques will likely lead to improved precision. This may well be at the expense of keeping to a single robust algorithm. For the case $\sin ax^2$ we increased the working precision by a factor of four, resulting in only a very modest improvement in the accuracy obtained.

We now turn to a pair of considerably more difficult examples. The first is $f(x) = \cos(ax^{-1})$ where a is a constant, for which $(Hf)(x) = -\operatorname{sgn} a \sin(ax^{-1})$. The function $(\cos(x^{-1}))/(\pi(2-x))$ is displayed in the vicinity of the origin in Fig. 4. Inspection of this figure immediately reveals the expected difficulties to be associated with

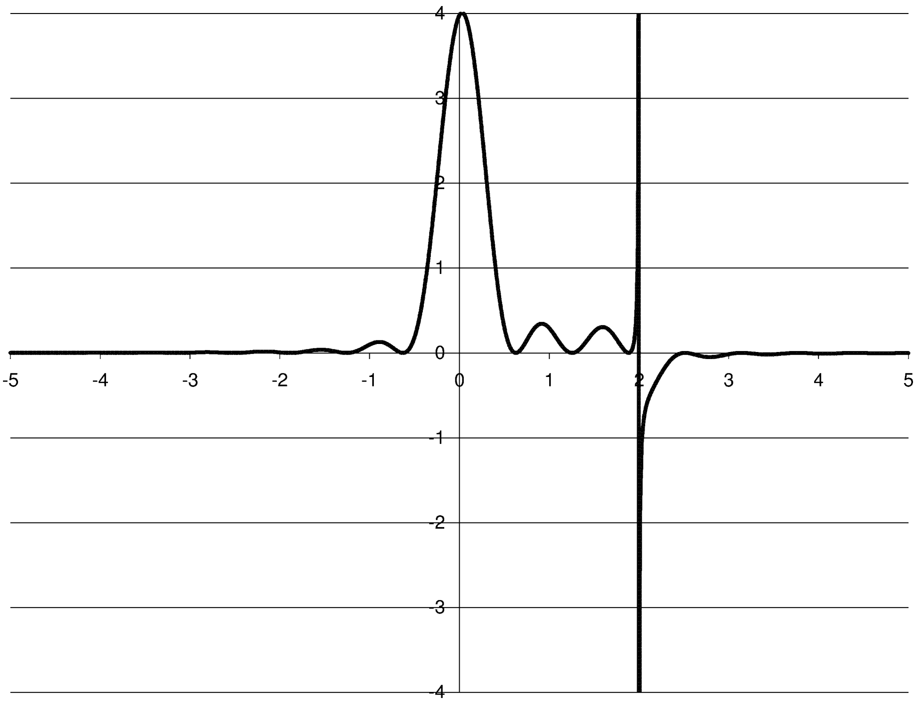

Fig. 1. Plot of the function $\{\pi x^2(2-x)\}^{-1} \sin^2 5x$ in the vicinity of the singular point at $x = 2$.

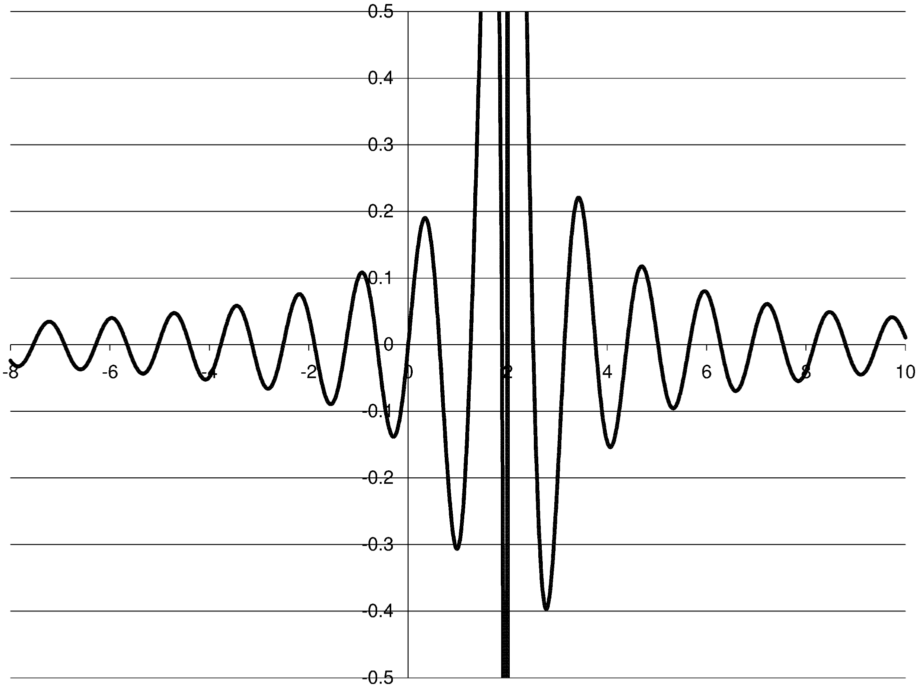

Fig. 2. Plot of the function $\{\pi(2-x)\}^{-1} \sin 5x$ in the vicinity of the singular point at $x = 2$.

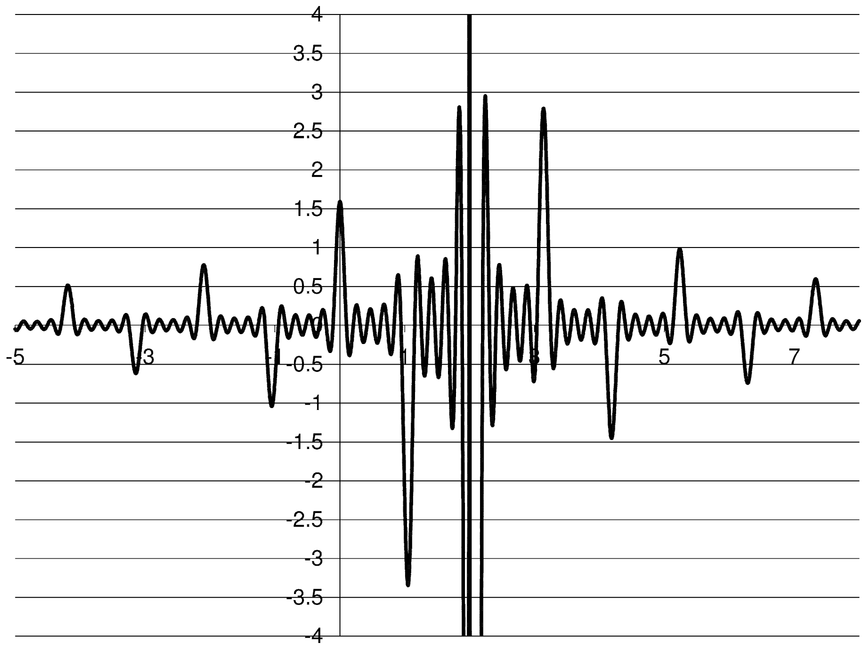

Fig. 3. Plot of the function $\{\pi(2-x)\sin 3x\}^{-1}\sin 30x$ in the vicinity of the singular point at $x=2$.

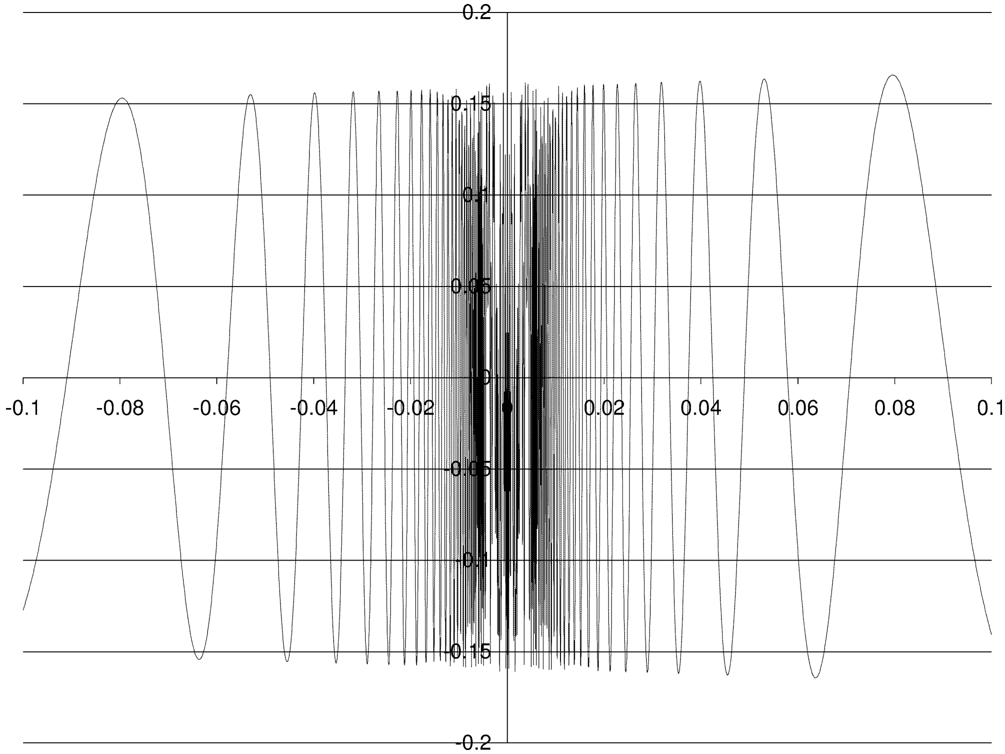

Fig. 4. Plot of the function $\{\pi(2-x)\}^{-1}\cos(x^{-1})$ in the vicinity of the origin.

this example. The extreme oscillation as $x \to 0$ makes this case rather recalcitrant to solve with a general robust type of algorithm. We applied a modified approach to deal with this example. For convenience we assume the singularity is located in $(1, \infty)$. The integration range was split into $(-\infty, -1)$, $(-1, 1)$, $(1, x_0-\varepsilon)$, $(x_0-\varepsilon, x_0+\varepsilon)$, and $(x+\varepsilon, \infty)$, and the first, third, fourth and fifth intervals treated as described in Section 2. The procedure of Section 2 was also applied to the second interval, but the origin was approached from *both* the points -1 and 1. As the center of the pocket of extreme oscillation is approached the number of roots spirals so significantly, that a cutoff in the number of determined roots must obviously be made. This approach yielded a modest level of precision as evidenced by the values reported for $\cos x^{-1}$ in Table 1. If the singularity is located in $(-\infty, -1)$ or in $(-1, 1)$, but not too close to the origin, then the scheme just outlined can be applied in a similar fashion. If the singular point is located very close to the origin, numerical difficulties are to be expected, since minute changes in the x-ordinate lead to significant changes in the value of the oscillatory component of the integrand.

The next choice considered is $f(x) = \cos(\alpha \cot x)$ where α is a constant and $\alpha > 0$, for which $(Hf)(x) = -\sin(\alpha \cot x)$. Fig. 5 illustrates a plot of $(\cos(\alpha \cot x))/(\pi(2 - x))$ for $\alpha = 1$, and it is clear that it will be extremely difficult to handle the packets of extreme oscillations that occur periodically beyond the singular point. A modification of the procedure used to deal with the case $\cos x^{-1}$ should work on functions that have periodically placed pockets of extreme oscillation, provided that the overall decay characteristics of the function are not too slow. Oscillatory functions with non-periodically placed pockets of extreme oscillation would be significantly more difficult to deal with using a general robust type of algorithm. Integrals with the singularity embedded in a region of extreme oscillation, for example, if the singular point in the present example is located very close to $x = n\pi$, for integer n, lead to a significant increase in the complexity of the problem. Such integrals are probably best dealt with on a case-by-case basis.

For all the examples presented in Table 1, an attempt was made to evaluate the integrals directly using the *Mathematica* object *CauchyPrincipalValue*. For the cases $\sin ax$, $\cos ax$, and $\sin ax^2$ the calculation returned totally inaccurate values, with a warning indicating precision loss. For the functions sinc x, $(\sin^2(ax))/x^2$, and

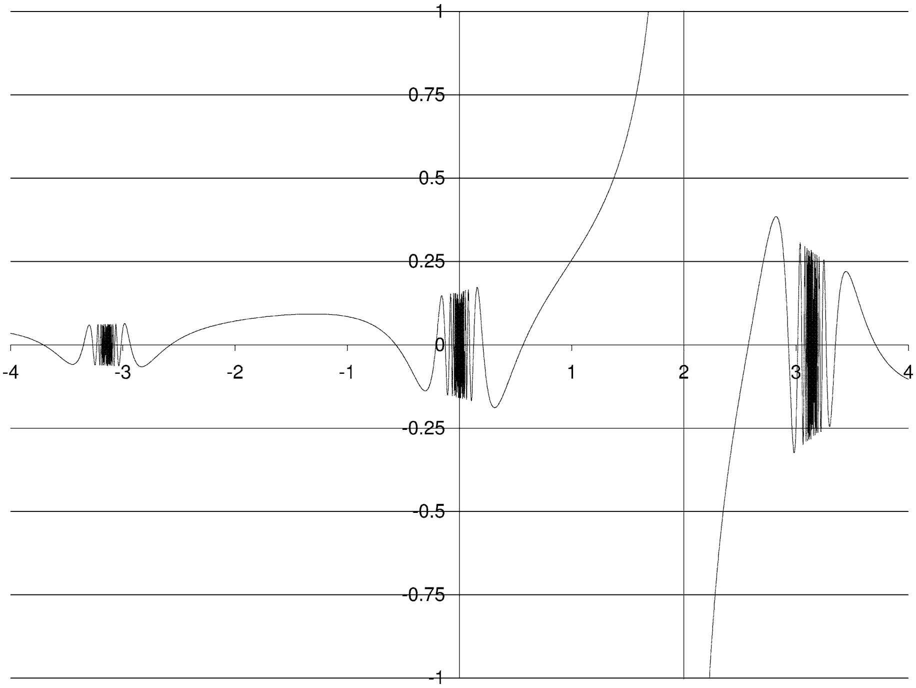

Fig. 5. Plot of the function $\{\pi(2-x)\}^{-1}\cos(\cot x)$ in the vicinity of the of the singular point at $x = 2$.

$(\sin 2abx)/(\sin bx)$ the calculation aborted without returning a value, citing infinite and indeterminate expressions encountered. Apparently, the algorithm embedded in *CauchyPrincipalValue* is not sufficiently robust to handle general oscillatory behavior of the type exhibited by the functions discussed in this work. The *Mathematica* option to deal with oscillatory integrands is rather restrictive, and in any case, did not prove to be effective.

Not surprisingly, the one example that *Mathematica* could deal with successfully was $\cos \pi x\, e^{-x^2}$. The reason for this is not difficult to see. The Gaussian component of the function dies off so quickly (and smoothly) for large values of x that the oscillatory behavior is only significant around the origin, but in this region the degree of oscillation is small, and does not lead to any significant loss of precision in the calculation. We would therefore expect a reasonably robust numerical integration routine to have little difficulty obtaining an accurate result for this particular example.

We note that the integration approach discussed in Section 2 can break down for wildly oscillatory integrands or for integrands whose asymptotic nature is not well approximated as a slowly convergent infinite series. Examples of the former were discussed above. Trouble in the latter case occurs because the partitioned $h(x)$ can provide a set of integrals that give rise to a sequence of terms that is not necessarily monotonically decreasing. This does not mean that the approach fails in such cases, but rather that the expected precision of the result will drop off. An alternative approach that may be more successful would attempt to isolate any asymptotic characteristics that are suggestive of two or more slowly convergent series. If this is the case, the principal series may be broken up into multiple infinite series and convergence accelerated individually. Here it may be possible to use the larger oscillations in a separate series from the smaller oscillations. An example where this might give improved results is the function $(\sin x)^{-1} \sin 2\alpha x$ for α a constant. However, questions of robustness of the algorithm must also be kept in mind, as we would not want changes in this new approach to curtail the high quality results already gained using the proposed algorithm.

The technique discussed in Section 2 can be utilized with very little change to cover the case of non-oscillatory continuous functions with suitable decay characteristics. We have not pursued this line of investigation in the current work, since there are many techniques available to cover this case.

Acknowledgements

Support from The Petroleum Research Fund of the American Chemical Society and the National Science Foundation are greatly appreciated.

References

[1] I.M. Longman, Math. Tables Aids Comp. 12 (1958) 205.
[2] L.M. Delves, Comput. J. 10 (1967) 389.
[3] I.H. Sloan, J. Comput. Phys. 3 (1968) 332.
[4] R. Piessens, BIT 10 (1970) 476.
[5] K. Atkinson, SIAM J. Numer. Anal. 9 (1972) 284.
[6] D.F. Paget, D. Elliott, Numer. Math. 19 (1972) 373.
[7] D.B. Hunter, Numer. Math. 21 (1973) 185.
[8] D. Elliott, D.F. Paget, Numer. Math. 23 (1975) 311.
[9] D. Elliott, D.F. Paget, Numer. Math. 25 (1976) 287.
[10] S.J. Collocott, Comp. Phys. Commun. 13 (1977) 203.
[11] P. Rabinowitz, in: Symposium on Numerical Mathematics, University of Natal, Durban, 1978, p. 53.
[12] S.J. Collocott, G.J. Troup, Comput. Phys. Commun. 17 (1979) 393.
[13] D. Elliott, D.F. Paget, Math. Comput. 33 (1979) 301.
[14] M.M. Chawla, S. Kumar, Computing 23 (1979) 67.
[15] O.E. Taurian, Comput. Phys. Commun. 20 (1980) 291.

[16] S. Kumar, J. Inst. Math. Appl. 26 (1980) 447.
[17] G. Monegato, Computing 29 (1982) 337.
[18] P.R. Rabinowitz, Math. Comput. 41 (1983) 63.
[19] D.S. Lubinsky, P. Rabinowitz, Math. Comput. 43 (1984) 219.
[20] G.A. Gazonas, Intern. J. Comput. Math. 18 (1986) 277.
[21] P. Rabinowitz, Intern. J. Comput. Math. 20 (1986) 283.
[22] G. Criscuolo, G. Mastroianni, Math. Comput. 48 (1987) 725.
[23] P. Rabinowitz, D.S. Lubinsky, Math. Comput. 53 (1989) 279.
[24] G. Criscuolo, G. Mastroianni, in: P. Nevai, A. Pinkus (Eds.), Progress in Approximation Theory, Academic Press, Boston, 1991.
[25] T. Hasegawa, T. Torii, Math. Comput. 56 (1991) 741.
[26] T. Hasegawa, T. Torii, J. Comput. Appl. Math. 51 (1994) 71.
[27] K. Diethelm, Proc. Sympos. Appl. Math. 48 (1994) 287.
[28] K. Diethelm, Z. Angew. Math. Mech. 74 (1994) T689.
[29] K. Diethelm, Computing 52 (1994) 337.
[30] K. Diethelm, J. Comput. Appl. Math. 56 (1994) 321.
[31] B.D. Vecchia, BIT 34 (1994) 346.
[32] G. Criscuolo, B.D. Vecchia, D.S. Lubinsky, G. Mastroianni, J. Math. Anal. Appl. 189 (1995) 256.
[33] L. Gori, E. Santi, Numer. Algorithms 10 (1995) 27.
[34] K. Diethelm, Approx. Theory Appl. 11 (1995) 78.
[35] K. Diethelm, J. Comput. Appl. Math. 65 (1995) 97.
[36] K. Diethelm, Z. Angew. Math. Mech. 75 (1995) S621.
[37] K. Diethelm, J. Comput. Appl. Math. 66 (1996) 167.
[38] K. Diethelm, Bolyai Soc. Math. Stud. 5 (1996) 175.
[39] K. Diethelm, Numer. Math. 73 (1996) 53.
[40] K. Diethelm, J. Comput. Appl. Math. 82 (1997) 93.
[41] K. Diethelm, J. Comput. Appl. Math. 112 (1999) 45.
[42] B. Bialecki, P. Keast, J. Comput. Appl. Math. 112 (1999) 3.
[43] K. Diethelm, P. Köhler, J. Inequal. Appl. 5 (2000) 167.
[44] F.W. King, submitted for publication.
[45] C. A Emeis, L.J. Oosterhoff, G. de Vries, Proc. Roy. Soc. London A 297 (1967) 54.
[46] H. Morawitz, J. Comput. Phys. 6 (1970) 120.
[47] D. Parris, S.J. Van Der Walt, Anal. Biochem. 68 (1975) 321.
[48] D.W. Johnson, J. Phys. A: Math. Gen. 8 (1975) 490.
[49] F.W. King, J. Phys. C: Solid State Phys. 10 (1977) 3199.
[50] F.W. King, J. Opt. Soc. Amer. 68 (1978) 994.
[51] T. Andersson, J. Johansson, H. Eklund, Math. Comput. Simul. 23 (1981) 262.
[52] H.-P. Liu, D.D. Kosloff, Geophys. J. Roy. Astr. Soc. 67 (1981) 791.
[53] K. Ohta, H. Ishida, Appl. Spectros. 42 (1988) 952.
[54] S.L. Hahn, Hilbert Transforms in Signal Processing, Artech House, Boston, 1996.
[55] S.L. Hahn, in: A.D. Poularikas (Ed.), The Transforms and Applications Handbook, CRC Press, Boca Raton, 1996, p. 463.
[56] P.L. Butzer, R.J. Nessel, Fourier Analysis and Approximation, One-Dimensional Theory, Vol. 1, Academic Press, New York, 1971.
[57] J.N. Pandey, The Hilbert Transform of Schwartz Distributions and Applications, John Wiley & Sons, New York, 1996.
[58] S. Wolfram, The Mathematica Book, 4th edn., Cambridge University Press, Cambridge, 1999.
[59] C. Brezinski, M. Redivo Zaglia, Extrapolation Methods Theory and Practice, North-Holland, Amsterdam, 1991.
[60] J. Wimp, Sequence Transformations and their Applications, Academic Press, New York, 1981.
[61] D. Levin, Int. J. Comput. Math. B 3 (1973) 371.
[62] P. Wynn, Proc. Cambridge Philos. Soc. 52 (1956) 663.
[63] D.A. Smith, W.F. Ford, SIAM J. Numer. Anal. 16 (1979) 223.
[64] D.A. Smith, W.F. Ford, Math. Comput. 38 (1982) 481.
[65] T. Fessler, W.F. Ford, D.A. Smith, ACM Trans. Math. Soft. 9 (1983) 346.
[66] E.J. Weniger, Comput. Phys. Rep. 10 (1989) 189.
[67] E.J. Weniger, Comput. Phys. Commun. 64 (1991) 19.
[68] E.J. Weniger, Int. J. Quantum Chem. 57 (1996) 265; Erratum, Int. J. Quantum Chem. 58 (1996) 319.
[69] H.H.H. Homeier, J. Comput. Appl. Math. 69 (1996) 81.
[70] H.H.H. Homeier, Numer. Algorithms 17 (1998) 223.
[71] U.D. Jentschura, P.J. Mohr, G. Soff, E.J. Weniger, Comput. Phys. Commun. 116 (1999) 28.
[72] H.H.H. Homeier, J. Comput. Appl. Math. 122 (2000) 81.
[73] E.J. Weniger, J. Comput. Appl. Math. 122 (2000) 329.