JOURNAL OF COMPUTATIONAL PHYSICS 65, 344–369 (1986)

A Dual Algorithm for Fast Calculation of the H_0^1-Transform*

SEBASTIEN M. CANDEL,† ERIC BOUSSARIE,‡
JEAN-MARC LOESCH,§ AND ANNIE LELARGE″

Office National d'Etudes et de Recherches Aérospatiales (ONERA), BP 72, 92322 Châtillon Cedex, Ecole Centrale des Arts et Manufactures, 92290 Chatenay-Malabry, and SINTRA-ALCATEL, 94117 Arcueil, France

Received November 7, 1984; revised October 30, 1985

An integral transform which involves the zeroth-order Hankel function of the first kind, H_0^1, is commonly encountered in wave analysis and with particular reference to studies of underwater sound, seismology, and electromagnetic phenomena. This paper presents new methods for the numerical estimation of this transform based on the "dual" procedure concept. The computation involves two matched algorithms, the first of which provides the lower-order components while the second yields intermediate and higher-order estimates of the transform. It is shown that a procedure constructed along these lines generates accurate transform samples and requires a computation time of the order of that of a few fast Fourier transform operations.

1. INTRODUCTION

Considerable progress has been accomplished in the last 20 years with the development of fast transform algorithms like the fast Fourier and Walsh–Hadamard transforms (respectively designated FFT and FWHT). A number of more recent studies are concerned with the calculation of Fourier–Bessel (Hankel) transforms defined by

$$F_k(r) = \int_0^\infty J_k(\zeta r)\, \zeta f(\zeta)\, d\zeta \qquad \text{for } k = 0,\ldots, N \tag{1}$$

where the transform kernel involves the Bessel function J_k with argument ζr. Most of this work has concentrated on the zeroth-order transform and a variety of

* Research sponsored in part by the Direction des Recherches et Etudes Techniques (DRET).
† Research scientist at ONERA and Professor of Engineering at Ecole Centrale.
‡ Present address: CNES, 91000 Evry, France.
§ Present address: SDP, 92150 Suresnes, France.
″ Present address: SINTRA-ALCATEL, 94117 Arcueil, France.

0021-9991/86 $3.00

methods have been proposed. Siegman [1], Sheng [2], and Talman [3] use the "Gardner transform" to convert expression (1) into a correlation which is then evaluated by performing three FFT operations. Brunol and Chavel [4], Cavanagh and Cook [5], Nachamkin and Maggiore [6], and Miraglia et al. [7] base their methods on various series expansions. A linear filtering concept is proposed by Johansen and Sorensen [8]. Oppenheim et al. [9] show that it is possible to calculate the kth-order Fourier–Bessel integral by taking the one-dimensional Fourier transform of the projection $p(x)$ of the function under transformation onto the real axis.

Another related technique exploited independentiy by Candel [10] involves a single one-dimensional Fourier transform followed by repeated summations of preselected Fourier components. An extension of this technique presented in Ref. [11] allows the determination of the first N Fourier–Bessel transforms of a given function. It involves a basic Fourier transform, repeated selection of suitable Fourier components, and successive evaluation of Fourier series coefficients. The separate estimation of even and odd transform sets is considered in Ref. [12] and an error analysis is performed in Ref. [13]. The technique proposed in Ref. [10] is accurate but not fully efficient. Considerable improvement of this aspect is obtained by making use of the "dual" procedure derived in Ref. [14]. The computation involves two matched algorithms. The first derived in Ref. [10] provides the lower-order components, i.e., the components corresponding to low values of the transform argument. The second algorithm uses asymptotic formulations and yields intermediate and higher-order components.

Switching from the first to the second algorithm takes place when the results of both are in agreement to a certain acceptable error. The advantage of the dual procedure is that the computational load rests mainly on the asymptotic algorithm and as a consequence the number of operations is of the order of that required by three to five FFT. Furthermore this concept may be extended to other cases and it is used in the present paper to enhance the computational efficiency.

The latest addition to the literature dealing with the Fourier–Bessel transform is due to Mook [15]. His method involves an Abel transform evaluated as a convolution followed by a Fourier transform. This interesting method is efficient but it may be quire inaccurate in certain cases.

We consider in this paper a definite integral

$$F(r) = \int_{-\infty}^{+\infty} \zeta H_0^1(\zeta r) f(\zeta)\, d\zeta \tag{2}$$

designated as the H_0^1-transform of the function $f(\zeta)$, that features the zeroth-order Hankel function of the first kind, $H_0^1(z) = J_0(z) + iY_0(z)$. The H_0^1-transform (2) contrasts with the specific Fourier–Bessel transform

$$F_0(r) = \int_0^{\infty} \zeta J_0(\zeta r) f(\zeta)\, d\zeta \tag{3}$$

with respect to the choice of cylinder kernel function and range of interpretation; if $f(\zeta)$ has an even symmetry, the proportionality

$$F(r) = 2F_0(r) \tag{4}$$

obtains in keeping with the relations

$$J_0(z) = \tfrac{1}{2}[H_0^1(z) + H_0^2(z)], \qquad H_0^2(e^{i\pi}z) = -H_0^1(z).$$

Our motivation in dealing with the H_0^1-transform comes from the fact that this integral appears in many wave problems. Standard representations of the Green's function in unbounded or layered domains are based on this integral transform. To illustrate this aspect let us consider for a moment some alternative representations of the free space Green's function for the Helmholtz equation, corresponding to a source at the origin and the time dependence e^{-ikct}. These representations include the explicit version

$$G(\mathbf{r}) = \frac{e^{ik(r^2+z^2)^{1/2}}}{4\pi(r^2+z^2)^{1/2}} \tag{5}$$

in terms of the cylindrical coordinates r, z and the respective integrals

$$G(\mathbf{r}) = \frac{i}{8\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{e^{i\xi x + i\eta y + i(k^2-\xi^2-\eta^2)^{1/2}|z|}}{(k^2-\xi^2-\eta^2)^{1/2}}\, d\xi\, d\eta \tag{6}$$

$$= \frac{i}{8\pi}\int_{-\infty}^{+\infty} \frac{\zeta H_0^1(\zeta r)\, e^{i(k^2-\zeta^2)^{1/2}|z|}}{(k^2-\zeta^2)^{1/2}}\, d\zeta \tag{7}$$

$$= \frac{i}{8\pi}\int_{-\infty}^{+\infty} H_0^1[(k^2-\zeta^2)^{1/2} r]\, e^{i\zeta|z|}\, d\zeta \tag{8}$$

$$= \frac{i}{4\pi}\int_{0}^{\infty} \frac{\zeta J_0(\zeta r)\, e^{i(k^2-\zeta^2)^{1/2}}}{(k^2-\zeta^2)^{1/2}}\, d\zeta \tag{9}$$

which differ in the underlying type of modal synthesis (see Felsen and Marcuvitz [16] and Aki and Richards [17]). The pair (6) and (7) feature plane and cylinder wave functions with direction of propagation normal to the z-axis; (8) involves propagating waveforms along the z-direction and a radial wave function that has a propagating or attenuating character depending on the magnitude of the axial wave number; and, evidently, (9) is the Fourier–Bessel counterpart of the H_0^1-transform (7) with the proportionality signalled in (4).

Although the representation (6) has merits for analytical purposes since the coordinates x, y, z are separated in a simple fashion, it is not well suited for numerical purposes since a double Fourier integral appears therein.

Now consider the last representation (9). This integral is of the Fourier–Bessel transform type. In analytical studies such integrals are usually more difficult to han-

dle than expressions like (7) because changes of integration path are less easily performed. From the numerical standpoint direct calculations based on expressions like (9) may lead to erroneous results because the transform kernel contains converging (towards the source) and diverging cylindrical waves. The converging components are not easily discarded in such calculations.

In contrast expression (7) is more readily manipulated, in particular by deforming the integration contour in the complex plane and employing saddle point asymptotic methods. It is also more amenable to direct numerical calculations of radiating fields because converging cylindrical wave components may be avoided.

The conclusion reached on this simple example actually applies to more complex situations of wave propagation in layered domains. When the radiation conditions must be enforced, H_0^1-transform representations are most suitable. Such representations are particularly useful when the function under transformation is complex and has no parity properties (as, for example, a function having poles which do not contribute to the far field and must be discarded).

Hence, a good algorithm for the calculation of H_0^1-transforms is clearly needed. We were unable to find a reference specifically concerned with the estimation of this transform but note that simple asymptotic evaluations have been used in the recent literature (see, for example, Di Napoli and Deavenport [19]).

Because the transform kernel contains the H_0^1 Hankel function, it is oscillatory and a straight forward integration based on standard rules is not very appropriate. Furthermore a conventional integration method requires a large number of calculations (a multiple of NL if there are N data points to define the function under transformation and L estimates of the transform to be determined).

In the present paper we develop an alternative method for the calculation of the H_0^1-transform. It uses a dual procedure technique similar to that of Ref. [14]. The gain achieved with this method is somewhat comparable to that obtained by estimating a Fourier transform with the FFT instead of using a conventional integrating rule.

The mathematical background for the two algorithms involved in the procedure is given in Section 2. Discrete formulations of the basic expressions are developed in Section 3. An error analysis is conducted in Section 4 and some guidelines for the choice of the main parameters of the discrete transform are given. Section 5 presents results of test calculations.

2. Mathematical Background

The H_0^1-transform defined in the introduction by expression (2) features the zeroth-order Hankel function H_0^1. This function has a branch singularity at $\zeta = 0$. A branch cut is conventionally defined along the negative real axis in the complex ζ-plane.

Since $H_0^1(\zeta r)$ decays exponentially when $|\zeta| \to \infty$ and $\mathrm{Im}(\zeta) > 0$ it is natural (see Felsen and Marcuvitz [16, p. 480]) to consider that the integration path follows

the real axis slightly above the branch cut for $\mathrm{Re}(\zeta) \leqslant 0$ and that it coincides with the real axis for $\mathrm{Re}(\zeta) > 0$.

Now to construct a dual procedure for the H_0^1-transform we have to formulate an algorithm for the lower-order transform samples (algorithm L) and another algorithm for the higher-order samples (algorithm A). We shall also describe a technique for removing singularities from the function under transformation.

A. *Basic Expressions for Algorithm L*

The H_0^1-transform is defined in the introduction by expression (2). In general $f(\zeta)$ is a complex function and its transform $F(r)$ is also complex. The transform argument r is a positive real number belonging to the open interval $]0, +\infty[$. The transform kernel contains the zeroth-order Hankel function of the first kind. This function may be replaced by the integral expression (see Abramovitz and Stegun [18])

$$H_0^1(x) = (2/i\pi) \int_0^\infty \exp(ix \cosh t)\, dt. \tag{10}$$

Inserting this representation in the definition (2) yields

$$F(r) = (2/i\pi) \int_{-\infty}^{+\infty} f(\zeta)\, \zeta\, d\zeta \int_0^\infty \exp(i\zeta r \cosh t)\, dt. \tag{11}$$

Interchanging the order of integration leads to

$$F(r) = (2/i\pi) \int_0^\infty \phi(r \cosh t)\, dt \tag{12}$$

where

$$\phi(\eta) = \int_{-\infty}^{+\infty} f(\zeta)\, \zeta \exp(i\zeta\eta)\, d\zeta \tag{13}$$

designates the Fourier transform of $f(\zeta)\,\zeta$ and η is the transform argument. According to these last expressions the transform $F(r)$ may be obtained by summing Fourier components $\phi(\eta)$ corresponding to the variable $\eta = r \cosh t$ for t belonging to the semi-infinite interval $[0, +\infty[$.

Now, expressions (12) and (13) are useful if the infinite integral (12) converges. To examine this important issue it is convenient to change the integration variable. Thus by letting $\eta = r \cosh t$ in expression (12) one obtains

$$F(r) = \frac{2}{i\pi} \int_r^\infty \frac{\phi(\eta)}{(\eta^2 - r^2)^{1/2}}\, d\eta \tag{14}$$

We now consider the finite integral

$$G(r, \eta_M) = \frac{2}{i\pi} \int_r^{\eta_M} \frac{\phi(\eta)}{(\eta^2 - r^2)^{1/2}} \, d\eta \tag{15}$$

and assume that the Fourier transform $\phi(\eta)$ decays sufficiently rapidly as η increases and is of order $\eta^{-\alpha}$ with $\alpha > 1$ as η exceeds η_M and tends to infinity

$$\phi(\eta) < A\eta^{-\alpha} \qquad \eta > \eta_M. \tag{16}$$

Then, it is a simple matter to show that

$$|F(r) - G(r, \eta_M)| < B\eta_M^{-\alpha} \tag{17}$$

where B is a constant.

As a consequence $G(r, \eta_M)$ is an asymptotic approximation to $F(r)$ and it tends uniformly to $F(r)$ as η_M tends to infinity.

Condition (16) is satisfied by a large class of functions and it is not very restrictive.

Under this rather weak condition the finite integral (15) may be used to approximate the H_0^1-transform.

In practice it is more convenient to use t as the integration variable.

Then expression (8) becomes

$$G(r, \eta_M) = (2/i\pi) \int_0^{t_M} \phi(r \cosh t) \, dt \tag{18}$$

where $t_M = \cosh^{-1}(\eta_M / r)$.

The first variant L1 of algorithm L is based on expressions (13) and (18). A second variant L2 relies on the expressions (13) and (15). However, the integrand of (15) is singular. The singularity must be removed and this may be achieved by writing

$$G(r, \eta_M) = \frac{2}{i\pi} \int_r^{\eta_M} d\eta \, \frac{\phi(\eta) - \phi(r)}{(\eta^2 - r^2)^{1/2}} + \phi(r) \frac{2}{i\pi} \int_r^{\eta_M} \frac{d\eta}{(\eta^2 - r^2)^{1/2}}. \tag{19}$$

The last term is integrated at once while the first has a regular integrand if $\phi(\eta)$ is continuous and differentiable at $\eta = r$:

$$G(r, \eta_M) = \frac{2}{i\pi} \int_r^{\eta_M} d\eta \, \frac{\phi(\eta) - \phi(r)}{(\eta^2 - r^2)^{1/2}} + \frac{2}{i\pi} \phi(r) \cosh^{-1}\left(\frac{\eta_M}{r}\right). \tag{20}$$

Expressions (13) and (20) constitute the variant L2 of algorithm L. This variant has some merits because it uses directly available Fourier components and thus avoids interpolation. On the other hand it requires the evaluation of $(\eta^2 - r^2)^{-1/2}$ and multiplications by this function. Algorithms L1 and L2 are otherwise comparable.

B. *Basic Expressions for Algorithm A*

The determination of the higher-order components of the H_0^1-tranform may be based on the classical asymptotic expansion of the Hankel function for large argument. This technique is well known and it is used, for example, by Di Napoli and Deavenport [19]. However, the precision of this approximation is not well documented and possible improvement of the first-order approximation based on the higher-order terms of the asymptotic expansion has not been investigated (see Candel [14] for a study of the last point in the case of the Fourier–Bessel transform).

We shall also show that further improvement of accuracy may be obtained by removing the singularity of the asymptotic transform.

The following argument closely follows that of Ref. [14].

Now let $q_1, q_2, \ldots, q_j, \ldots$ designate the successive terms of the Hankel function asymptotic expansion and let Q_M represent the series obtained by adding the first M terms of this expansion. Then

$$Q_M(z) = \sum_{j=1}^{M} q_j(z) \tag{21}$$

and

$$H_0^1(z) = Q_M(z) + O(1/z^{M+1/2}). \tag{22}$$

Expressions for $q_j(z)$ may be found in Abramovitz and Stegun [18]. We shall use only the first two terms:

$$q_1(z) = (2/\pi z)^{1/2} \exp(iz - i\pi/4) \tag{23}$$

$$q_2(z) = -(i/8z)(2/\pi z)^{1/2} \exp(iz - i\pi/4). \tag{24}$$

Higher-order terms may be included with little effort. However the Mth term $q_M(z)$ behaves like $1/z^{M-1/2}$ near $z = 0$ and as M increases this term becomes more singular. As a consequence it is not useful to include more than two terms in the asymptotic approximation of the Hankel function. Now consider the integral approximation obtained by replacing $H_0^1(\zeta r)$ by $Q_1(\zeta r)$ in definition (2):

$$F^{(1)}(r) = \int_{-\infty}^{+\infty} f(\zeta) \left(\frac{2}{\pi r}\right)^{1/2} \zeta^{1/2} \exp(i\zeta r - i\pi/4)\, d\zeta. \tag{25}$$

Clearly, the first-order asymptotic approximation of the H_0^1-transform has the form of a Fourier integral and it may be evaluated with the FFT.

Next let us consider the second-order approximation of the H_0^1-transform obtained by replacing the Hankel function $H_0^1(\zeta r)$ by $Q_2(\zeta r)$. This new estimate may be obtained simply by adding a correction term to the first-order approximation

$$F^{(2)}(r) = F^{(1)}(r) + E_2(r) \tag{26}$$

where

$$E_2(r) = -\frac{i}{8r^{3/2}}\left(\frac{2}{\pi}\right)^{1/2} \int_{-\infty}^{+\infty} \zeta^{-1/2} f(\zeta) \exp(i\zeta r - i\pi/4)\, d\zeta. \tag{27}$$

This correction also involves a Fourier integral and it may be calculated with the FFT.

Practical applications (see Section 5) indicate that the first-order approximation $F^{(1)}(r)$ rapidly converges towards the exact transform and that the convergence rate and accuracy are increased with the second-order correction. In fact the approximate kernel $K_1(\zeta, r) = Q_1(\zeta r)\,\zeta$ rapidly tends to the exact transform kernel $K(\zeta, r) = \zeta H_0^1(\zeta r)$ when r is increased. The convergence of $K_2(\zeta, r) = \zeta Q_2(\zeta r)$ is even faster. These aspects are illustrated in Ref. [14] in the case of the Fourier–Bessel transform for the kernel $\zeta J_0(\zeta r)$ which corresponds to the real part of the present kernel. Similar results are also obtained for the imaginary part.

C. *A Technique for Removing a Singularity from the Function under Transformation*

In many circumstances the function under transformation has a singularity at some point of the integration range.

If this singularity is removed and treated analytically the resulting transform estimates may improve drastically.

To show how this procedure may be used, let us consider the case of a function which is discontinuous at the origin. More specifically we assume that $f(\zeta)$ has a single jump discontinuity at $\zeta = 0$.

This jump is denoted by

$$[f] = f(0_+) - f(0_-).$$

Now it is possible to write $f(\zeta)$ as a sum of a continuous and a discontinuous function:

$$f(\zeta) = f_c(\zeta) + f_D(\zeta).$$

The function $f_D(\zeta)$ must have the same jump discontinuity as $f(\zeta)$. It is convenient to choose

$$\begin{aligned} f_D(\zeta) &= [f] \exp(-\zeta/b) && \text{for } \zeta \geqslant 0 \\ &= 0 && \text{for } \zeta < 0 \end{aligned}$$

where b is a positive constant.

With this decomposition the H_0^1-transform is

$$F(r)=\int_{-\infty}^{+\infty} f_c(\zeta)\,\zeta H_0^1(\zeta r)\,d\zeta+[f]\int_0^{+\infty} e^{-\zeta/b}\zeta H_0^1(\zeta r)\,d\zeta. \tag{28}$$

$F(r)$ is the sum of the H_0^1-transform of the continuous function $f_c(\zeta)$ and the H_0^1-transform of a decaying exponential. This last term may be determined analytically as

$$\int_0^{\infty} e^{-\zeta/b}\zeta H_0^1(\zeta r)\,d\zeta=\frac{b^2}{\alpha^3}+i(2/\pi)\,b^2\left[1-\alpha^{-1}\ln\left(\frac{1+\alpha}{br}\right)\right]\alpha^{-2} \tag{29}$$

where $\alpha=(1+b^2r^2)^{1/2}$.

The same argument may be used to treat singularities located at other points of the integration range.

The technique just described is exploited by Mook [15] in a paper dealing with the calculation of the Hankel transform. As indicated by this author the treatment is not traditional because the singularity is subtracted out with a windowing function (in this case $\exp(-\zeta/b)$). The reason for this treatment and the choice of the parameter b are discussed in that reference.

3. Numerical Implementation

For practical application it is necessary to derive discrete expressions from the various relations given in the previous section.

Let $\Delta\zeta$ designate the sampling period and let $\hat{f}(n)$ represent the discrete sequence obtained by sampling the function $f(\zeta)$ at a constant rate:

$$\begin{aligned}\hat{f}(n)&=f(n\Delta\zeta) && n=0,\ldots,P/2\\ &\quad f[(n-P)\,\Delta\zeta] && n=P/2+1,\ldots,P-1.\end{aligned} \tag{30}$$

The first $(P/2+1)$ samples of this sequence correspond to positive values of ζ while the next $(P/2-1)$ samples represent negative values of that argument.

Now we shall seek estimates of the H_0^1-transform $F(r)$ for a set of discrete values of r:

$$r_l=l\Delta r, \qquad l=1,\ldots,L.$$

It is convenient (but not essential for algorithm L) to relate the sampling periods $\Delta\zeta$ and Δr by $\Delta\zeta\,\Delta r=2\pi/N$ where N is an integer dividing P exactly. The standard rule for dual procedures is to set N equal to the number of samples P.

Now let

$$\tilde{F}(l)=F(l\Delta r)/(\Delta\zeta)^2, \qquad l=1,\ldots,L \tag{31}$$

designate a sequence of sampled and scaled H_0^1-transform values and let $\hat{F}(l)$ stand for the discrete estimates calculated numerically.

The input and output sequences are now defined and we may describe the various computational schemes.

We first consider algorithms L1 and L2 for the lower-order components of the H_0^1-transform and then algorithms A1 and A2 for the higher-order components, and finally conclude with the dual procedure.

A. *Algorithm L1*

The continuous Fourier transform (13) may be evaluated as a discrete transform with the FFT algorithm. For this we let

$$\begin{aligned} \hat{h}(n) &= \hat{f}(n)\, n, && n = 0,\ldots, P/2 \\ &= \hat{f}(n)(n-P), && n = P/2+1,\ldots, P-1. \end{aligned} \tag{32}$$

Then, the discrete Fourier transform of $f(\zeta)\,\zeta$ is defined by

$$\hat{\phi}(k) = \sum_{n=0}^{P-1} \hat{h}(n)\, e^{i2\pi kn/P}, \qquad k = 0,\ldots, P-1. \tag{33}$$

The components of this sequence constitute, under certain general conditions, a set of estimates of the continuous Fourier transform and it is possible to write

$$\phi(k\Delta\eta) \simeq \hat{\phi}(k)(\Delta\zeta)^2 \tag{34}$$

where the sampling period $\Delta\eta$ satisfies the standard rule $\Delta\zeta\,\Delta\eta = 2\pi/P$.

Our next task is to replace the continuous integral (18) by a discrete expression. Consider a fixed value of l. The corresponding upper limit of integration is given by

$$t_{\mathrm{MAX}} = \cosh^{-1}(\eta_{\mathrm{MAX}}/r_l).$$

Now the maximum value of η corresponding to the discrete transform (26) is $\eta_{\mathrm{MAX}} = (P/2)\,\Delta\eta$ so that

$$t_{\mathrm{MAX}} = \cosh^{-1}\left[\frac{(P/2)\,\Delta\eta}{l\,\Delta r}\right] = \cosh^{-1}\left(\frac{N}{2l}\right) \tag{35}$$

This upper limit exists only if $l \leqslant N/2$ and, as a consequence, $L < N/2$.

The continuous integral (18) may be evaluated with standard integration rules.

To be consistent with the FFT approximation used above it is natural to use the simplest schemes like the rectangle, midpoint, or Simpson rules. To be specific let us consider the second method. The integration range is split in S equal segments $\Delta t = t_{\mathrm{MAX}}/S$ and the midpoint of each segment is selected

$$t_j = (j+\tfrac{1}{2})\,\Delta t, \qquad j = 0,\ldots, S-1.$$

Then a simple approximation of $G(r_l, \eta_M)$ is obtained from

$$G(l\Delta r, \eta_M) = (2/i\pi)\,\Delta t \sum_{j=0}^{S-1} \phi(l\Delta r \cosh t_j). \tag{36}$$

Now the Fourier transforms appearing in this sum must be evaluated in terms of the available discrete samples $\hat{\phi}(k)$. This requires an interpolation which may be performed in various ways.

It is again consistent to use the simplest schemes like

(a) nearest-neighbour interpolation,

(b) linear interpolation,

(c) cubic Hermite interpolation.

For simplicity we only describe the first scheme. Now consider the set of real numbers

$$d(j, l) = l(\Delta r/\Delta\eta)\cosh t_j = l(P/N)\cosh t_j \tag{37}$$

and let $k(j, l)$ designate the set of integers obtained from

$$k(j, l) = \mathrm{Int}[d(j, l) + \tfrac{1}{2}] \tag{38}$$

where $\mathrm{Int}(\cdot)$ designates the integer part of its argument.

Clearly, $k(j, l)\,\Delta\eta$ is the Fourier transform argument closest to $\Delta r \cosh t_j$. Then, expression (36) may be approximated by

$$G_2(l\Delta r, \eta_M) = (2/i\pi)\,\Delta t \sum_{j=0}^{S-1} \phi[k(j, l)\,\Delta\eta]. \tag{39}$$

Finally we replace the sampled values of the continuous transform of the last expression by discrete estimates. This leads to

$$G_3(l\Delta r, \eta_M) = (2/i\pi)\,\Delta t \sum_{j=0}^{S-1} \hat{\phi}[k(j, l)](\Delta\zeta)^2. \tag{40}$$

By scaling this expression by $(\Delta\zeta)^2$ we obtain a sampled and scaled estimate of the H_0^1-transform:

$$\hat{F}(l) = G_3(l\Delta r, \eta_M)/(\Delta\zeta)^2 = (2/i\pi)\,\Delta t \sum_{j=0}^{S-1} \hat{\phi}[k(j, l)]. \tag{41}$$

The method just described is probably not the most efficient. It has many features in common with the Fourier–Bessel transform algorithm proposed in Ref. [10]. In the present case, however, the integration range is variable and the size of the elementary segments Δt and the location of the points t_j change with l.

Hence, it is necessary to determine Δt, t_j, $\cosh t_j$, $d(j, l)$, and $k(j, l)$ for each value of l. It is also necessary to form a sum of S selected Fourier components for each value of l. Now recall that algorithm L serves only to calculate a finite number of $\tilde{F}(l)$ corresponding to low values of l.

The other estimates are obtained with the much more efficient algorithm A. Thus the efficiency of algorithm L is not the main issue.

are summarized in Table I.

B. *Algorithm* L2

The following scheme is based on expressions (13) and (20). The integration range $[r, \eta_{MAX}]$ now corresponds to the range of Fourier transform indices $[l(P/N), P/2]$. If this range is divided in $S = P/2 - lP/N$ intervals then each

TABLE I

Expression Leading to the H_0^1-Transform (Algorithm L1)

1. Let $f(\zeta)$ designate the function under transformation
2. Choose the sampling periods $\Delta\zeta$ and Δr to be such that $\Delta\zeta\, \Delta r = 2\pi/N$
3. Choose the FFT size $P = 2^\gamma$ (γ is an integer) with $P \geqslant N$
4. Specify the number L of samples to be determined, $L \leqslant N/2$
5. Select the value of the number of summation segments S; S is typically of the same order as N
6. Define the sequence

$$\hat{h}(n) = f(n\,\Delta\zeta)\, n, \qquad n = 0,\ldots,\ P/2$$
$$\hat{h}(n) = f[(n-P)\,\Delta\zeta](n-P), \qquad n = P/2+1,\ldots,\ P-1$$

7. Compute the discrete Fourier transform of $\hat{h}(n)$

$$\hat{\phi}(k) = \sum_{n=0}^{P-1} \hat{h}(n) \exp(i2\pi kn/P), \qquad k = 0,\ldots,\ P-1$$

8. Set the value of $l = 1$
9. Calculate $t_{MAX} = \cosh^{-1}(N/2l)$ and $\Delta t = t_{MAX}/S$
10. Define the set of integration points

$$t_j = (j + \tfrac{1}{2})\,\Delta t, \qquad \text{for} \quad j = 0,\ldots,\ S-1$$

11. Obtain the set of indices

$$k(j, l) = \text{int}[l(P/N)\cosh(t_j) + \tfrac{1}{2}]$$

12. Perform

$$\hat{F}(l) = (2\Delta t/i\pi) \sum_{j=0}^{S-1} \hat{\phi}[k(j, l)]$$

13. If $l < L - 1$ let $l = l + 1$ and return to step 9

integration segment extends over a single sampling period $\Delta\eta$ and expression (13) is easily approximated with the rectangle rule

$$\hat{F}(l) = (2/i\pi)\,\hat{\phi}(lP/N)\cosh^{-1}(N/2l) + (2/i\pi)\sum_{j=1}^{S-1}\{\hat{\phi}[k(j,l)] - \hat{\phi}(lP/N)\}/\alpha(j,l)^{1/2} \tag{42}$$

where

$$k(j,l) = lP/N + j,$$
$$\alpha(j,l) = k(j,l)^2 - k(0,l)^2. \tag{43}$$

The main advantage of this new formulation is that it does not require an interpolation. The computational complexity is, however, of the same order as that of algorithm L1.

The expressions leading to the H_0^1-transform (algorithm L2) are summarized in Table II.

C. *Algorithms* A1 *and* A2

Algorithm A1 is based on expression (25). We first define the sequence

$$\begin{aligned}\hat{h}(n) &= \hat{f}(n)\,n^{1/2}\exp(-i\pi/4), && n = 0,\ldots, N/2\\ &= \hat{f}(n)(N-n)^{1/2}\exp(i\pi/4), && n = N/2+1,\ldots, N-1.\end{aligned} \tag{44}$$

and evaluate the Fourier integral appearing in (25) with the discrete transform

$$\hat{\psi}(k) = \sum_{n=0}^{N-1}\hat{h}(n)\exp(i2\pi kn/N). \tag{45}$$

The sampled and scaled asymptotic estimates of the H_0^1-transform are then given by

$$\hat{F}^{(1)}(l) = \frac{N^{1/2}}{\pi}\frac{1}{l^{1/2}}\hat{\psi}(l), \qquad l = 1,\ldots, N/2 \tag{46}$$

For algorithm A2 the transform estimate is obtained by adding a correction $\hat{E}_2(l)$ to $\hat{F}^{(1)}(l)$ according to (26) and (27).

This correction is obtained by defining the new sequence

$$\begin{aligned}\hat{e}(n) &= \hat{f}(n)\,n^{-1/2}\exp(-i\pi/4), && n = 1,\ldots, N/2\\ &= \hat{f}(n)(N-n)^{-1/2}\exp(i\pi/4), && n = N/2+1,\ldots, N-1\\ \hat{e}(0) &= [\hat{e}(1) + \hat{e}(N-1)]/2\end{aligned} \tag{47}$$

TABLE II

Expressions Leading to the H_0^1-Transform (Algorithm L2)

1. Let $f(\zeta)$ designate the function under transformation
2. Choose the FFT size $P = 2^\gamma$ (γ is an integer)
3. Choose the sampling periods $\Delta\zeta$ and Δr to be such that $\Delta\zeta\,\Delta r = 2\pi/N$
4. Specify the number L of samples to be determined, $L \leqslant N/2$
5. Define the sequence

$$\hat{h}(n) = f(n\,\Delta\zeta)\,n, \qquad n = 0,\ldots,\ P/2$$
$$\hat{h}(n) = f[(n-P)\,\Delta\zeta](n-P), \qquad n = P/2+1,\ldots,\ P-1$$

6. Compute the discrete Fourier transform of $\hat{h}(n)$

$$\hat{\phi}(k) = \sum_{n=0}^{P-1} \hat{h}(n)\exp(i2\pi kn/P), \qquad k = 0,\ldots,\ P-1$$

7. Set $l = 1$
8. Determine $S = P/2 - l(P/N)$
9. Calculate

$$k(j, l) = l(P/N) + j, \qquad j = 0,\ldots,\ S-1$$

10. Calculate

$$\alpha(j, l) = [k(j, l)]^2 - (lP/N)^2$$

11. Calculate

$$t_{\mathrm{MAX}} = \cosh^{-1}(N/2l)$$

12. Perform

$$\hat{F}(l) = \frac{2}{i\pi}\left\{\hat{\phi}(lP/N)\,t_{\mathrm{MAX}} + \sum_{j=1}^{S-1} \frac{\hat{\phi}[k(j, l)] - \hat{\phi}(lP/N)}{[\alpha(j, l)]^{1/2}}\right\}$$

13. If $l < L-1$ let $l = l+1$ and return to step 8

and taking the discrete Fourier transform of this sequence

$$\hat{\Theta}(k) = \sum_{n=0}^{N-1} \hat{e}(n)\exp(i2\pi kn/N). \tag{48}$$

The correction term is then estimated by

$$\hat{E}_2(l) = -\frac{N^{3/2}}{16\pi^2}\frac{1}{l^{3/2}}\hat{\Theta}(l), \qquad l = 1,\ldots,\ N/2 \tag{49}$$

and the second-order asymptotic H_0^1-transform is obtained from

$$\hat{F}^{(2)}(l) = \hat{F}^{(1)}(l) + \hat{E}_2(l). \tag{50}$$

TABLE III

Expressions Leading to the Asymptotic H_0^1-Transform (Algorithm A1)

1. Let $f(\zeta)$ designate the function under transformation
2. Choose the sampling period $\Delta\zeta$
3. Select the FFT size $N=2^\gamma$ (γ is an integer); if algorithm A1 is used in a dual procedure the value of N should be the same as that of algorithm L
4. Define the sequence

$$\hat{h}(n)=f(n\,\Delta\zeta)\,n^{1/2}\exp(-i\pi/4), \qquad n=0,\ldots,N/2$$
$$\hat{h}(n)=f[(n-N)\,\Delta\zeta](N-n)^{1/2}\exp(i\pi/4), \qquad n=N/2+1,\ldots,N-1$$

5. Compute the discrete Fourier transform

$$\hat{\psi}(k)=\sum_{n=0}^{N-1}\hat{h}(n)\exp(i2\pi kn/N), \qquad k=0,\ldots,N-1$$

Determine the asymptotic estimates of the H_0^1-transform

$$\hat{F}^{(1)}(l)=\frac{1}{\pi}\left(\frac{N}{l}\right)^{1/2}\hat{\psi}(l), \qquad l=1,\ldots,N/2$$

Expressions leading to the first-order asymptotic H_0^1-transform (algorithm A1) are assembled in Table III for easy reference.

D. *Dual Procedures*

It is now possible to construct various dual procedures by combining algorithm L1 or L2 with algorithm A1 or A2. These algorithms may be used in conjunction with the singularity removal technique of Section 2. In the first step a complete set of asymptotic transform estimates is obtained from algorithm A1 or A2. Next, algorithm L is started and the transform estimates are evaluated in a sequential fashion beginning with the $l=1$ sample and eventually ending with the $l=N/2$ sample.

At each step the transform estimates obtained from algorithms L and A are compared. The difference between these estimates is formed

$$\varepsilon(l)=\|\hat{F}^p(l)-\hat{F}(l)\|$$

where $p=1$, 2 and the norm $\|z\|$ represents the modulus of z or the sum of the real and imaginary parts of z.

Now this difference is bounded by the sum of the differences between the exact transform and its estimates

$$\varepsilon(l)\leqslant\|\tilde{F}(l)-\hat{F}^p(l)\|+\|\hat{F}^p(l)-\hat{F}(l)\|. \tag{51}$$

The first term corresponds to the error associated with algorithms A1 or A2. This term tends to zero as l increases. The second term represents the error associated with algorithm L. It is bounded and usually decreases with ζ.

When $\varepsilon(l)$ falls below a certain acceptable value, algorithm L may be stopped and the remaining transform estimates are those determined at the outset with algorithm A.

Another switching criterion is based on the value of the correction term $E_2(l)$ (if it has been computed). $E_2(l)$ is representative of the error associated with algorithm A1. Then if $E_2(l)$ becomes less than a certain small value this algorithm has converged and it is possible to stop algorithm L.

4. An Error Estimate for Algorithm L1

An error estimate for algorithm L1 may be obtained by following the reasoning of Ref. [10].

A first error ε_1 is introduced when the infinite integral (5) is replaced by the finite integral (11). If $\phi(\eta) \leqslant A\eta^{-\alpha}$ with $\alpha > 1$ as $\eta > \eta_M$ this error is bounded by

$$\varepsilon_1 = |F(r) - G(r, \eta_M)| \leqslant B\eta_M^{-\alpha}. \tag{52}$$

The second error arises from the application of a discrete integration rule to expression (18). For the rectangle rule the error bound may be deduced from Ref. [20]:

$$\varepsilon_2 = |G_1(r, \eta_M) - G(r, \eta_M)| \leqslant \frac{1}{12\pi}(\Delta t)^2 \left| \frac{d^2\phi(r\cosh t)}{dt^2} \right|_{\text{MAX}}. \tag{53}$$

The third error appears in the interpolation process. If nearest-neighbour interpolation is used $\phi(r\cosh t)$ is replaced by $\phi(\eta_j)$ where $\eta_j = k(j, l)\,\Delta\eta$. The corresponding error is bounded by

$$\begin{aligned} \varepsilon_3 &= |G_2(r, \eta_M) - G_3(r, \eta_M)| \\ &\leqslant \frac{2}{\pi}\Delta t |\phi'(\eta)|_{\text{MAX}} \sum_{j=0}^{S-1} |r\cosh t_j - \eta_j|. \end{aligned} \tag{54}$$

Now η_j is the nearest neighbour of $r\cosh t_j$ and consequently $|r\cosh t - \eta_j| \leqslant \Delta\eta/2$ and ε_3 is bounded by

$$\varepsilon_3 \leqslant \frac{2}{\pi} t_{\text{MAX}}\, \Delta\eta |\phi'(\eta)|_{\text{MAX}}. \tag{55}$$

The last error arises when one replaces the exact Fourier transform components $\phi(\eta_j)$ by discrete estimates $(\Delta\zeta)^2\,\hat{\phi}[k(j,l)]$. This error is bounded by

$$\varepsilon_4 = |G_3(r,\eta_M) - G_2(r,\eta_M)| \leqslant \frac{2}{\pi}\Delta t \sum_{j=0}^{S-1} |\phi(\eta_j) - (\Delta\zeta)^2\,\hat{\phi}[k(j,l)]|. \tag{56}$$

The average value of the difference between the continuous and discrete transforms appears in this bound. Thus the local errors induced by the discrete evaluation of the Fourier transform are averaged and the result affects the H_0^1-transform estimates.

The total error associated with algorithm L1 is less than the sum of the four previous bounds. The error estimate obtained is certainly coarse but it provides guidelines for the choice of parameters P, N, and S.

Consider the first error source ε_1. This error diminishes as η_{MAX} increases. Now $\eta_{MAX} = (P/2)\,\Delta\eta$ and because of the compatibility relation $\Delta\eta\,\Delta\zeta = 2\pi/P$, $\eta_{MAX} = \pi/\Delta\zeta$. Thus ε_1 may be reduced by decreasing the sampling period $\Delta\zeta$. Now let us assume that $\Delta\zeta$ has been selected so that η_{MAX} is fixed. Then if P is increased, the Fourier transform sampling period $\Delta\eta$ diminishes according to $\Delta\eta = 2\pi/(P\Delta\zeta)$ and as a consequence the interpolation error ε_3 decreases.

Reduction of ε_4 is achieved by increasing the Fourier transform size P and also by removing the singularities of the function being transformed. Thus large values of P should be used if accuracy is important. The computation time only increases like $P\log_2 P$ so that fairly large values of P may be selected and typically $P \geqslant 1024$.

An interesting feature of algorithm L1 is that the choice of N and L is independent of the value of P under the restrictions that $N \leqslant P$ and $L \leqslant N/2$. For example, one may choose $P = 4096$ to obtain precise estimates and set $N = 1024$ and $L = 128$. In this case the sampling rate of the H_0^1-transform estimates is $\Delta r = 4\Delta\eta$ and 128 values are determined. Hence, algorithm L1 is more flexible that the classical FFT.

Finally consider the error associated with the integration rule. This error is proportional to $(\Delta t)^2$ and $(\Delta t)^2 = S^{-2}[\cosh^{-1}(N/2l)]^2$. Then ε_2 is inversely proportional to S^2 and it decreases as l increases. A sufficiently large value of S must be used to reduce ε_2. On the other hand this value should not be too large because the number of operations performed to obtain a single transform estimate is a multiple of S. Typically, S should be greater than or equal to 256.

5. Examples of Calculations

We have submitted our algorithms to an extensive series of test calculations. It is in fact essential to demonstrate the validity of the proposed numerical methods in more than a single case. Indeed an examination of the literature dealing with the Fourier–Bessel transform shows that certain techniques which provide precise results in certain cases may also generate inaccurate estimates in other situations.

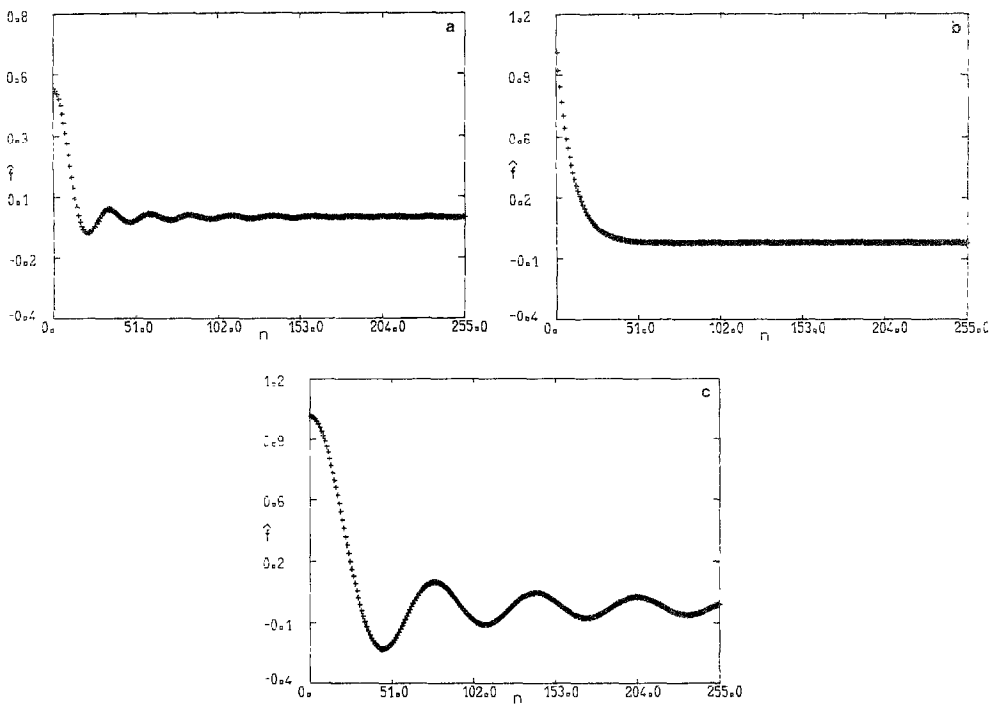

FIG. 1. Discrete samples of the three functions used to test the H_0^1-transform algorithms. (a) "Sombrero" or "jinc" function $\hat{f}(n) = J_1(n/M)/(n/M)$, $M = 4$. (b) Exponential function $\hat{f}(n) = \exp(-n/M)$, $M = 10$. (c) Sinc function $\hat{f}(n) = \text{sinc}(n/M)$, $M = 10$.

It is not possible to present all the tests performed but we have included results for three typical functions:

$$f(\zeta) = J_1(\zeta/a)/(\zeta/a), \qquad \zeta \geq 0 \tag{57a}$$

$$f(\zeta) = \exp(-\zeta/a), \qquad \zeta \geq 0 \tag{57b}$$

$$f(\zeta) = \sin(\zeta/a)/(\zeta/a) = \text{sinc}(\zeta/a), \qquad \zeta \geq 0. \tag{57c}$$

$f(\zeta) = 0$ for $\zeta < 0$ in all three cases. It is convenient to express the constant a in terms of the sampling period $\Delta\zeta$: $a = M\,\Delta\zeta$. Then the sequences obtained by sampling the previous functions at a constant rate are respectively

$$\hat{f}(n) = J_1(n/M)/(n/M) \tag{58a}$$

$$\hat{f}(n) = \exp(-n/M) \tag{58b}$$

$$\hat{f}(n) = \text{sinc}(n/M). \tag{58c}$$

Samples of these sequences are displayed in Fig. 1. The real parts of the H_0^1-transform of these functions may be deduced from Abramovitz and Stegun [18] or

the Bateman manuscript [21] and their sampled and scaled forms have been given in Refs. [10, 11].

Expressions for the imaginary parts of the H_0^1-transforms are available in the Bateman manuscript [21]. The sampled and scaled values are respectively given by

$$\mathrm{Im}[\hat{F}(l)] = -(M^2/\pi)\ln(1-\xi^{-2}), \qquad \xi > 1 \tag{59a}$$

$$\mathrm{Im}[\hat{F}(l)] = (2M^2/\pi)[1-\alpha^{-1}\ln(1+\alpha)\,\xi^{-1}]\,\alpha^{-2} \tag{59b}$$

$$\mathrm{Im}[\hat{F}(l)] = (2M^2/\pi)(1-\xi^2)^{-1/2}\ln[\xi^{-1}-(\xi^{-2}-1)], \qquad \xi < 1$$

$$= (2M^2/\pi)(\xi^2-1)^{-1/2}\arcsin(\xi^{-1}), \qquad \xi > 1 \tag{59c}$$

where

$$\alpha = (1+\xi^2)^{1/2}, \qquad \xi = 2\pi lM/N.$$

We have been unable to obtain a simple expression for the first test case in the range $0 < \xi < 1$.

A. *Results of Algorithm* L1

Results obtained from algorithm L1 for the three test functions are displayed in Fig. 2. The calculations correspond to $P = N = 1024$, $S = 512$. The transform estimates are represented by discrete symbols while the exact transforms are displayed as solid lines. The numerical estimates nearly coincide with the exact transforms in all cases. Weak oscillations may be observed in Fig. 2a near the transform discontinuity. This is related to Gibb's phenomenon. A slight offset is made evident on Fig. 2b. The estimates in Fig. 2c are extremely accurate while those in Fig. 2d also show a slight offset. The strongly singular transform of Figs. 2e and f is well evaluated numerically. The low-order estimates oscillate around the exact values while the higher-order samples are quite accurate. The general features of this transform are well retrieved.

The precision of algorithm L1 may be enhanced in various ways. For example consider the results obtained for the test function (*c*) by doubling the value of *P*. The results obtained for $P = N = 2048$, $S = 512$, are displayed in Fig. 3. Improvement of the imaginary part shown in Fig. 3b is quite evident.

Another method, presented in Section 2, consists of removing the singularities of the function under transformation. This technique is illustrated in Fig. 4 which displays the imaginary part of the H_0^1-transform for the test case (*b*). The offset in Fig. 2d has completely disappeared in Fig. 4.

B. *Results of Algorithms* A1 *and* A2

Results obtained with algorithm A1 are presented in Ref. [14] for the Fourier–Bessel transform (the real part of the H_0^1-transform if $f(\zeta)$ is real). The estimates tend asymptotically towards the exact transform but convergence is achieved for intermediate values of ζ.

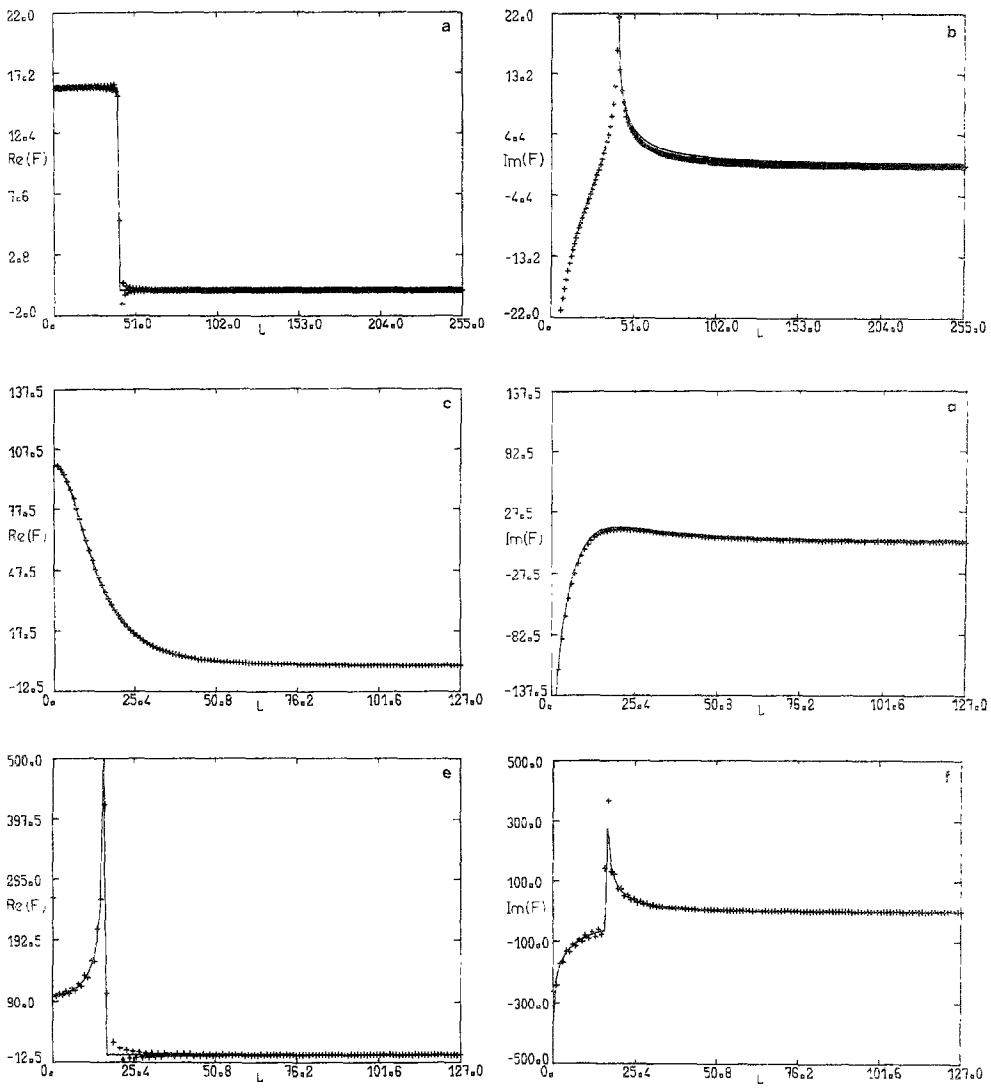

FIG. 2. H_0^1-transforms of the three test functions plotted in Fig. 1. The exact transforms are represented by solid lines. Estimates generated by algorithm L1 appear as discrete symbols. $P = N = 1024$, $S = 512$. (a, b) Transform of "jinc" function. (c, d) Transform of exponential function. (e, f) Transform of sinc function. (a), (c), and (e) display the real parts of the transform while (b), (d), and (f) show the imaginary parts.

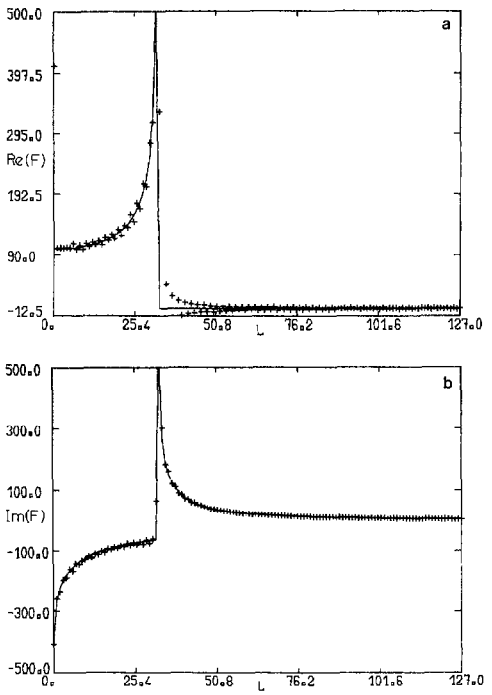

FIG. 3. H_0^1-transform of the sinc function plotted in Fig. 1c. The exact transform is represented by solid lines. Estimates generated by algorithm L1 appear as discrete symbols. The values of P and N have been doubled. $P = N = 2048$, $S = 512$. (a) Real part. (b) Imaginary part.

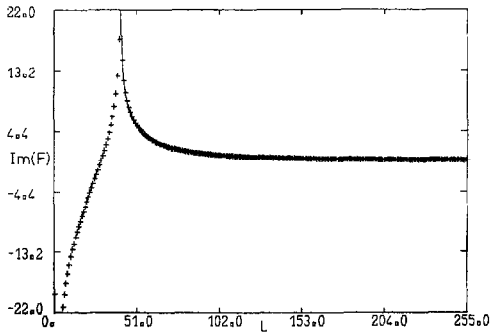

FIG. 4. Imaginary part of the H_0^1-transform of the "jinc" function plotted in Fig. 1a. The exact transform is represented by a solid line. Estimates generated by algorithm L1 appear as discrete symbols. The singularity at the origin is removed with the technique described in Section 2C. $P = N = 1024$, $S = 512$.

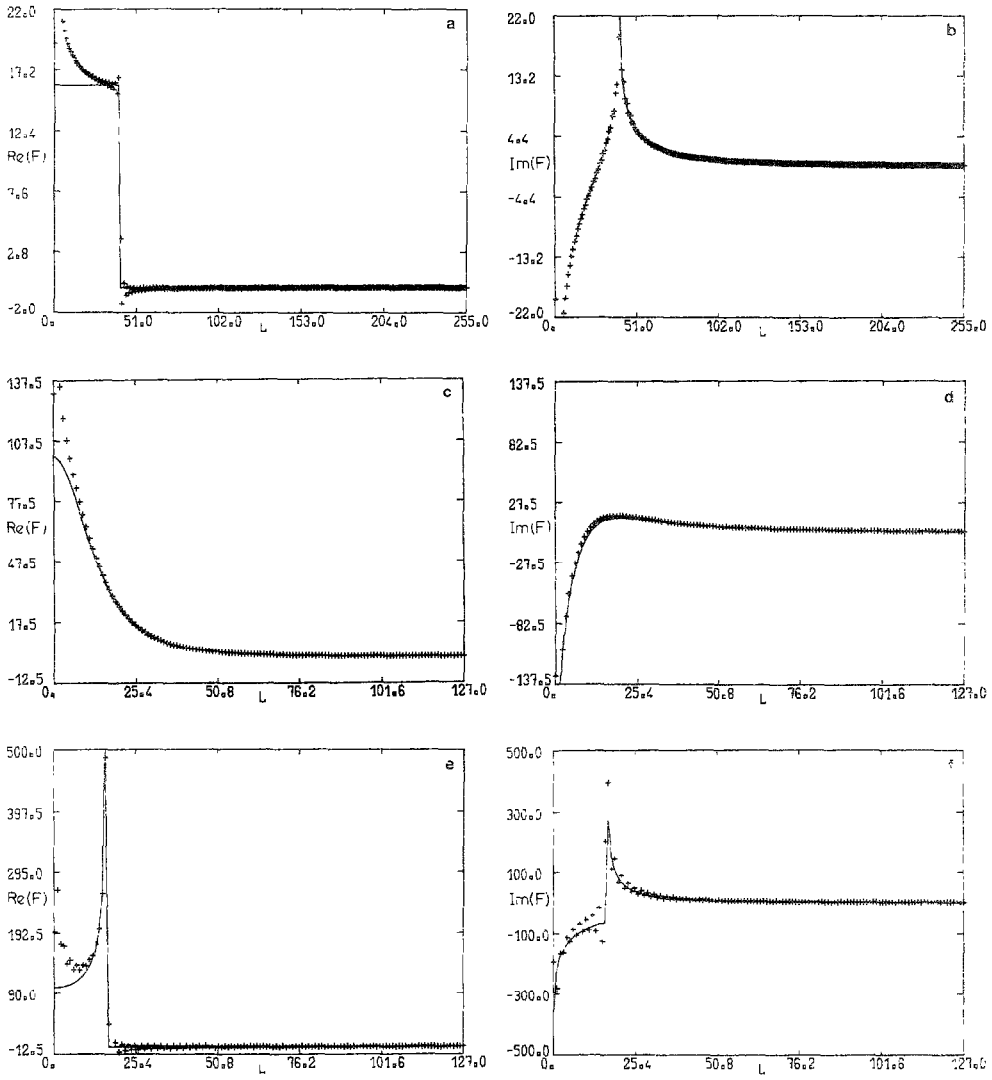

FIG. 5. H_0^1-transforms of the three test functions plotted in Fig. 1. The exact transforms are represented by solid lines. Estimates generated by algorithm A1 in conjunction with the singularity removal technique described in Section 2C appear as discrete symbols. $N = 1024$. (a, b) Transform of "jinc" function. (c, d) Transform of exponential function. (e, f) Transform of sinc function. (a), (c), and (e) display the real parts of the transform while (b), (d), and (f) show the imaginary parts.

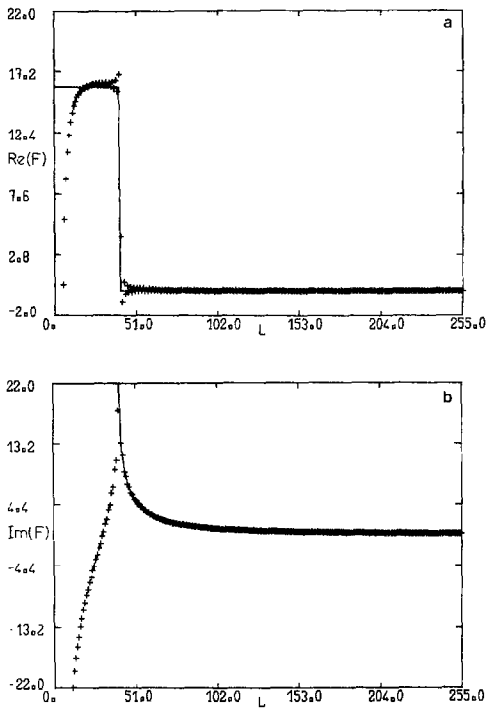

FIG. 6. H_0^1-transforms of the "jinc" function plotted in Fig. 1a. The exact transform is represented by solid lines. Estimates generated by algorithm A2 appear as discrete symbols. $N = 1024$. (a) Real part. (b) Imaginary part.

It is possible to improve this algorithm by applying the singularity removal technique of Section 2. Results obtained in this way are displayed in Fig. 5. The asymptotic estimates now rapidly converge towards the exact transforms.

Algorithm A2 also generates improved estimates in all cases. Figure 6 displays only the results obtained for the first test function.

C. *Results of a Dual Procedure*

Estimates obtained with a dual procedure based on algorithms L1 and A1 in conjunction with the singularity removal technique are plotted in Fig. 7. This procedure generates accurate estimates. Switching from L1 to A1 occurs for $l \lesssim 50$ and calculation is quite fast in all cases.

In conclusion we note that the various numerical methods proposed in this paper provide satisfactory estimates of the H_0^1-transform. Dual procedures constructed by combining two types of algorithms yield accurate results and require short computation times.

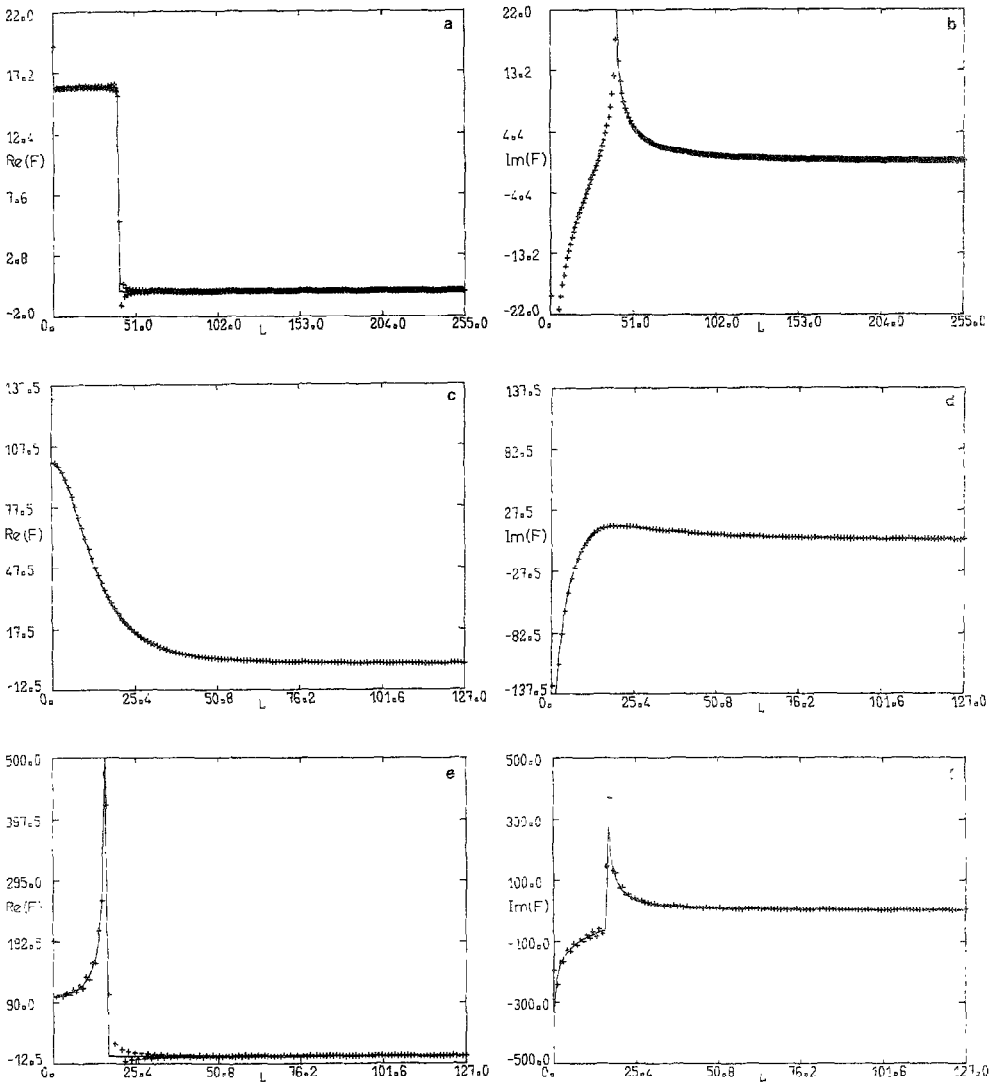

FIG. 7. H_0^1-transforms of the test functions plotted in Fig. 1. The exact transforms are represented by solid lines. Estimates generated by a dual procedure combining algorithms L1 and A1 and the singularity removal technique described in Section 2C appear as discrete symbols. $P = N = 1024$. $S = 512$. (a,b) Transform of jinc "function." (c, d) Transform of exponential function. (e, f) Transform of sinc function. (a), (c), and (e) display the real parts of the transform while (b), (d), and (f) show the imaginary parts.

6. Discussion

This paper presents various methods for calculating the H_0^1-transform. Two algorithms, L1 and L2, provide accurate estimates and may be used indifferently to determine the lower-order components of the transform. Two other algorithms, A1 and A2, based on an asymptotic expansion of the transform kernel provide estimates of the higher-order samples. Algorithm A2 is more accurate than A1 but requires more calculations. Numerical experiments indicate that the combination of L1, A1, and the singularity removal technique gives the same accuracy as the combination of L1 and A2.

Another important point is the choice between the H_0^1-transform and the Fourier–Bessel transform. For even functions some of the Fourier–Bessel transform algorithms developed in the recent literature require the smallest number of calculations.

When the function under transformation has no parity properties the H_0^1-transform is most suitable.

The dual procedure developed in this paper is fast when compared to a conventional integration of the transform. Another advantage of the method is that it resembles the fast Fourier transform. Hence it may be used with equal ease and requires similar precautions in sampling and windowing.

Acknowledgments

This work was performed in part while E. Boussarie and J. M. Loesch were final year engineering students in a training period at SINTRA-Alcatel, Département DSM. Part of this study was supported by a DRET contract. Permission to publish this work, granted by DRET, is gratefully acknowledged. Questions raised by the reviewers of the paper have been appreciated. We also acknowledge helpful discussions with Professor Harold Levine of Stanford University. Some test calculations were performed on an IBM 4341 computer made available to us by the IBM Company.

References

1. A. E. Siegman, *Opt. Lett.* **1**, 13 (1977).
2. S. C. Sheng, E. L. Gintzon Laboratory Report No. 3106, Stanford University, Stanford, CA, 1980. (unpublished)
3. J. D. Talman, *J. Comput. Phys.* **29**, 35 (1978).
4. J. Brunol and P. Chavel, *Proc. IEEE*, 1089 (1977).
5. E. Cavanagh and B. Cook, *IEEE Trans. Acoust. Speech Signal Process.* **27**, 361 (1979).
6. J. Nachamkin and C. J. Maggiore, *J. Comput. Phys.* **37**, 41 (1980).
7. J. E. Miraglia, R. D. Piacentini, and R. R. Rivarola, *Comput. Phys. Commun.* **19**, 299 (1980).
8. H. K. Johansen and K. Sorensen, *Geophys. Prospect.* **27**, 876 (1979).
9. A. V. Oppenheim, G. V. Frisk, and D. R. Martinez, *J. Acoust. Soc. Amer.* **68**, 523 (1980).
10. S. M. Candel, *Comput. Phys. Commun.* **23**, 343 (1981).
11. S. M. Candel, *J. Comput. Phys.* **44**, 243 (1981).
12. S. M. Candel, *Rech. Aerosp.*, 315 (1981).

13. S. M. CANDEL, *in Proceedings of IEEE* 1982 *International Conference on Acoustics Speech and Signal Processing, Paris, France* (IEEE, New York, 1982), p. 2076.
14. S. M. CANDEL, *IEEE Trans. Acoust. Speech Signal Process.* **29**, 963 (1981).
15. D. R. MOOK, *IEEE Trans. Acoust. Speech Signal Process.* **31**, 979 (1983).
16. L. B. FELSEN AND N. MARCUVITZ, *Radiation and Scattering of Waves* (Prentice–Hall, Englewood Cliffs, N. J., 1973).
17. W. AKI AND P. G. RICHARDS, *Quantitative Seismology* (Freeman, San Francisco, 1980), Vol. 1.
18. M. ABRAMOVITZ AND I. A. STEGUN, *Handbook of Mathematical Functions* (Dover, New York, 1969).
19. F. R. DI NAPOLI AND R. L. DEAVENPORT, *J. Acoust. Soc. Amer.* **67**, 92 (1980).
20. G. DAHLQUIST AND A. BJÖRCK, *Numerical Methods* (Prentice–Hall, Englewood Cliffs, N. J., 1974).
21. A. ERDELYI, W. MAGNUS, F. OBERHETTINGER, AND F. G. TRICOMI, *Tables of Integral Transforms, the Bateman Manuscript Project* (McGraw–Hill, New York, 1954).