Rose-Hulman Undergraduate Mathematics Journal

Volume 5
Issue 2

Article 8

Developing And Comparing Numerical Methods For Computing The Inverse Fourier Transform

Edgar J. Lobaton
Seattle University, edgar_loba@hotmail.com

Developing And Comparing Numerical Methods For Computing The Inverse Fourier Transform

Edgar J. Lobaton

Faculty Advisor: Dr. Donna Sylvester

Abstract

Computing the Fourier transform and its inverse is important in many applications of mathematics, such as frequency analysis, signal modulation, and filtering. Two methods will be derived for numerically computing the inverse Fourier transforms, and they will be compared to the standard inverse discrete Fourier transform (IDFT) method. The first computes the inverse Fourier transform through direct use of the Laguerre expansion of a function. The second employs the Riesz projections, also known as Hilbert projections, to numerically compute the inverse Fourier transform. For some smooth functions with slow decay in the frequency domain, the Laguerre and Hilbert methods will work better than the standard IDFT. Applications of the Hilbert transform method are related to the numerical solutions of nonlinear inverse scattering problems and may have implications for the associated reconstruction algorithms.

1. Introduction

The Fourier Transform has many uses in different areas of mathematics. It also has many applications in engineering and physical sciences. In signal processing, signals (functions dependent on time) can be transformed from the time domain to the frequency domain. Through this transformation the frequency components (spectrum) of a signal can be analyzed and passed through a filter to select a specific frequency range. The Fourier Transform is also used in physical sciences such as geophysics where scattering problems are encountered. In geophysics, for instance, acoustic signals (earthquakes) travel through a medium (earth) and many properties of the medium can be determined based on observations of the scattered signal. In particular, the frequency response properties (transmission properties) of the medium can be determined through a frequency analysis. Fourier Transforms are also used in other scattering applications such as transmission line and wave propagation problems in electrical engineering.

This paper will have a specific focus on Fourier Transforms with applications to scattering problems. For this type of applications, a function $f(x)$ is defined as having positive support (the function is equal to zero for $x < 0$) or negative support (the function is equal to zero for $x > 0$). Many different types of physical experiments can be modeled as scattering problems or inverse scattering problems. Scattering problems involve sending known signals into a medium and computing what will "bounce back". Physically, inverse scattering problems are problems where information about a medium can be recovered from known values measured at an exterior boundary. Some examples of inverse scattering problems are using sound waves to detect oil or earthquake fault lines, using radar to measure the thickness of ice and using impedance tomography to detect tumors. In some geophysical applications, the medium is under ground level, and the values measured at the exterior boundary are the reflections from wave signals sent underground. It is in this setting that having functions with negative support makes sense. This paper will therefore focus on the analysis of such functions.

The Fourier transform, $f^{\wedge}(\omega)$, of a function $f(x)$ is defined as

$$f^{\wedge}(\omega) = \int_{-\infty}^{\infty} f(x) \cdot e^{-ix\omega} dx \,. \tag{1}$$

This transform can be thought of as a mapping from the x-domain to the frequency (ω) domain. Some examples are shown in table 1. Table 2 shows some basic properties that will be used. The inverse Fourier transform of $F(\omega)$ is

$$F^{\vee}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cdot e^{ix\omega} d\omega = f(x) \,. \tag{2}$$

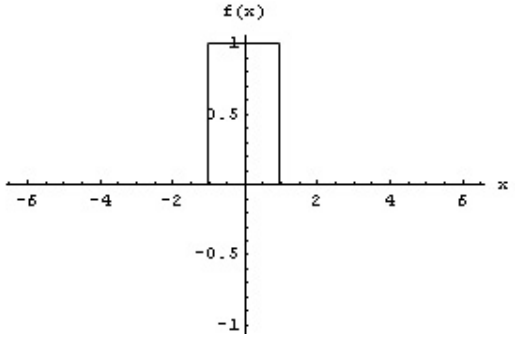
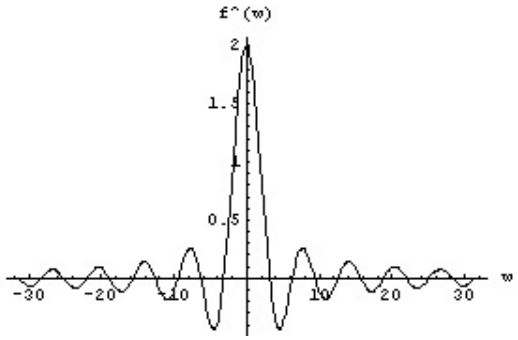
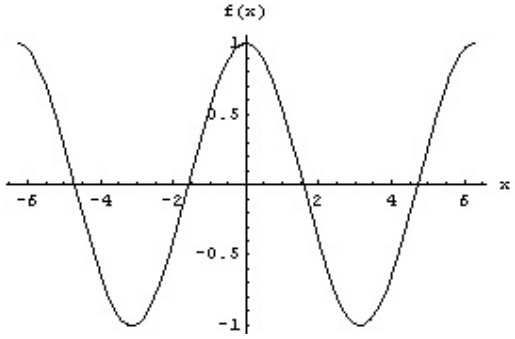
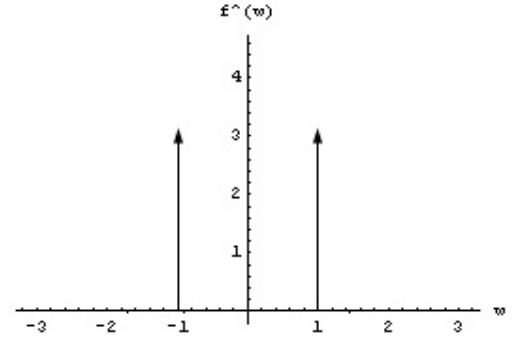
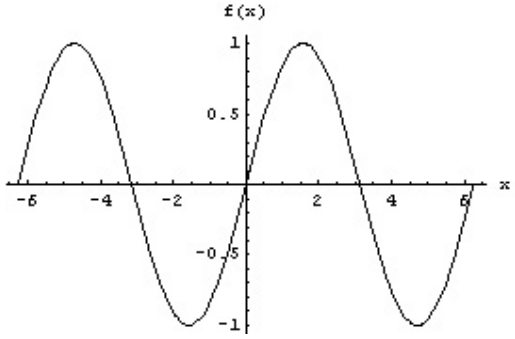
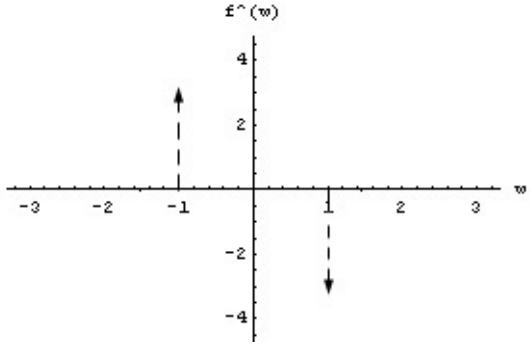
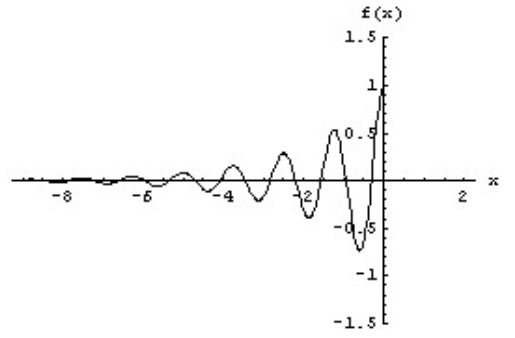
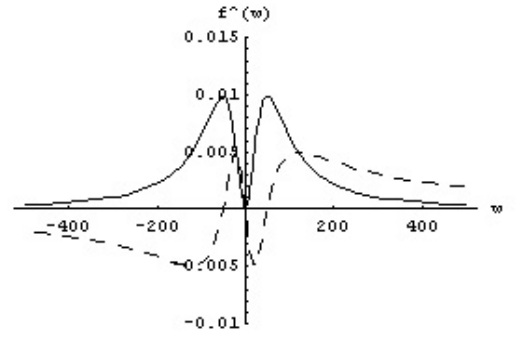
[1] $f(x)=\begin{cases}1, & x<\lvert 1\rvert \\ 0, & \textit{elsewhere}\end{cases}$ 	$f\hat{}(\omega)=\dfrac{2\sin(\omega)}{\omega}$ 
[2] $f(x)=\cos(x)$ 	$f\hat{}(\omega)=\pi\left[\delta(\omega-1)+\delta(\omega+1)\right]$ 
[3] $f(x)=\sin(x)$ 	$f\hat{}(\omega)=i\pi\left[\delta(\omega+1)-\delta(\omega-1)\right]$ 
[4] $f(x)=\begin{cases}e^{x/2}\cos(5x), & x\le 0 \\ 0, & \text{elsewhere}\end{cases}$ 	$f\hat{}(\omega)=\dfrac{2(1-2i\omega)}{100+(1-2i\omega)^2}$ 

Table 1 Sample functions (left) and their Fourier transform (right). (δ(ω) is the Dirac-delta function)

[1] $f(x)$	$f\hat{}(\omega)$
[2] $f(x-c)$	$f\hat{}(\omega)\cdot e^{-i\omega c}$
[3] $xf(x)$	$(-i)\cdot\dfrac{\partial f\hat{}(\omega)}{\partial\omega}$

Table 2 Some basic properties of functions (left) and their Fourier transforms (right).

The Fourier transform is (up to a negative sign and a scalar factor depending on the definition used) its own inverse. Thus, a numerical method that computes one, computes the other. However, with the exception of the Gaussian function (e^{-x^2}), functions and their Fourier transforms have different smoothness and decay properties. A function whose support is concentrated in a narrow interval will have a Fourier Transform with broad support. Some examples are shown in the first and fourth entries of table 1. Functions that are periodic can be represented as sums of discrete components in the frequency domain (i.e. Fourier series). The transform of the cosine or sine functions (second and third entries of table 1) are some examples. For this reason, a numerical method that approximates the Fourier Transform well for one class of functions may not be suitable to approximate the inverse Fourier Transform.

For a function supported on a bounded interval, the discrete Fourier Transform (DFT) (together with its fast implementation, the fast Fourier transform – FFT) provides a good approximation for the Fourier transform. For functions with broad support in the frequency domain (e.g., the whole real line) and slow decay (i.e. $1/\omega$), issues such as truncation error and aliasing can plague the DFT and the IDFT [4]. Two alternative methods for computing the inverse Fourier transform will be investigated for this class of functions.

The Discrete Fourier Transform, which is well known and widely used, will be one of the methods compared. The other two methods are derived from Laguerre expansions of the functions in the x-domain. The first method introduced will be the Laguerre method. This method computes the coefficients of the Laguerre expansion of a function in the x-domain and uses them to reconstruct the function from its Fourier transform. The second is the Hilbert method. This method makes use of the Riesz projections, also known as Hilbert projections, which separate the Fourier transform of a function into two orthogonal parts. The first is the Fourier transform of a function in the x-domain with support on the positive real line, and the second is the Fourier transform of a function supported on the negative real line. By alternating small shifts in the x-domain with these projections, we can represent a function in the frequency domain as a sum of functions, each of which is approximately the Fourier transform of a rectangle function (i.e. entry 1 in table 1). These methods will be described with more detail in the following section.

2. Background

2.1. Inverse discrete Fourier transform method (IDFT):

This method comes from the direct discretization of equation (2) through the use of the trapezoid rule. Suppose $f(\omega)$ is the Fourier transform of $F(x)$. To apply the IDFT, we first truncate $f(\omega)$ to the interval $[-W_0/2, W_0/2]$, and approximate the integral that defines the inverse Fourier transform as

$$F(x) \approx F^{IDFT}(x) = \frac{N}{2\pi W_0} \sum_{n=-N/2+1}^{N/2} f\hat{}(nW_0/N)\cdot e^{ix\cdot nW_0/N}, \qquad (3)$$

where N is the total number of points sampled in the frequency domain. Notice the approximation of $F^{IDFT}(x)$ in (3) is always periodic with period $T_0 = \dfrac{2\pi N}{W_0}$. This is a form of the celebrated reciprocity relation for the DFT:

$$T_0 \cdot W_0 = 2\pi \cdot N. \qquad (4)$$

This relation describes one limitation of the IDFT method. Since the IDFT interpolates a periodic approximation to the function $F(x)$, it will only yield useful information about $F(x)$ on the interval $(-T_0/2, T_0/2)$. This is particularly restrictive if we must take W_0 large to avoid truncation error. More details on this method can be found in [4].

2.2. Laguerre method:

The set of functions

$$\rho_n(\omega) = \left(\frac{i-\omega}{i+\omega}\right)^n \frac{i}{i+\omega},\ n = 0,\pm1,\pm2,... \tag{5}$$

form an orthogonal basis for $L^2(-\infty,\infty) = \{\ f\ \mid \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty\}$. The inverse Fourier transforms of these $\rho_n(\omega)$ functions are the Laguerre functions $L_n(x)$. For positive n, the $L_n(x)$ are zero for positive x. For negative x, some of these functions are

$L_0(x) = e^x$

$L_1(x) = (-2x-1)e^x$

$L_2(x) = (2x^2+4x+1)e^x$

Since the $\rho_n(\omega)$ form a basis for $L^2(-\infty,\infty)$,

$$f\hat{}(\omega) = \sum_{n=-\infty}^{\infty} f_n \cdot \rho_n(\omega), \tag{6}$$

for $f\hat{}(\omega) \in L^2(\mathbf{R})$. The function $f(x)$ can then be expressed as

$$f(x) = \sum_{n=-\infty}^{\infty} f_n \cdot \rho_n\check{}(x) = \sum_{n=-\infty}^{\infty} f_n \cdot L_n(x) \approx \sum_{n=-N+1}^{N} f_n \cdot L_n(x), \tag{7}$$

where N is an integer number. The coefficients f_n are defined as

$$f_n = \frac{1}{\pi}\int_{-\infty}^{\infty} f\hat{}(\omega)\overline{\rho_n}(\omega)d\omega, \tag{8}$$

where $\overline{\rho_n}(\omega)$ is the complex conjugate of $\rho_n(\omega)$. (See [5] for more details)

The following change of variable $\omega = \tan(\theta/2)$, which implies $e^{i\theta} = \frac{i-\omega}{i+\omega}$, can be made in equation (8) to obtain

$$f_n = \int_{-\infty}^{\infty} \frac{-i}{2\pi}(i+\tan(\theta/2))\, f\hat{}(\tan(\theta/2))e^{-in\theta}d\theta.$$ (See [2] for more details)

Letting $g(\theta) = \frac{-i}{2\pi}(i+\tan(\theta/2))\, f\hat{}(\tan(\theta/2))$,

$$f_n = \int_{-\infty}^{\infty} g(\theta)e^{-in\theta}d\theta = g\hat{}(n). \tag{9}$$

are the Fourier series coefficients of g, so the DFT or FFT can be used to compute $f_n = g\hat{}(n)$. Once the coefficients are found, the function can be reconstructed as the sum in (7).

The three term recursion relation

$$(n+1)L_{n+1}(x) + (2n+1)L_n(x) + nL_{n-1}(x) + 2xL_n(x) = 0 \tag{10}$$

can be used to compute the Laguerre functions. This relation can be verified by taking the Fourier transform of the left side of equation (10), (see table 2 for some transform properties used) which gives

$$(n+1)\rho_{n+1}(\omega) + (2n+1)\rho_n(\omega) + n\rho_{n-1}(\omega) + 2i\frac{\partial}{\partial\omega}\rho_n(\omega) = 0.$$

The left side of the previous equation is equal to zero after some algebraic manipulation. Since this equality is true, the inverse transform of the equality is also true, which verifies the validity of equation (10).

This gives an efficient way to compute the sum in equation (7). Based on the fact that the Laguerre functions have negative support for $n = 0, 1$; then it can be concluded from equation (10) that $L_n(x)$ has negative support for $n \geq 0$.

These results can be expanded to include $n < 0$ after noticing that

$$\overline{\rho_n}(\omega) = \overline{\left(\frac{i-\omega}{i+\omega}\right)^n \frac{i}{i+\omega}} = \left(\frac{i+\omega}{i-\omega}\right)^n \frac{i}{i-\omega}\left(\frac{i+\omega}{i+\omega}\right) = \left(\frac{i-\omega}{i+\omega}\right)^{-(n+1)} \frac{i}{i+\omega} = \rho_{-(n+1)}(\omega),$$

Also, for any complex function $h(\omega) \in L^2(\mathbf{R})$

$$(\overline{h(\omega)})^{\vee} = \frac{1}{2\pi}\int \overline{h(\omega)} \cdot e^{i\omega x} d\omega = \overline{\frac{1}{2\pi}\int h(\omega) \cdot e^{-i\omega x} d\omega} = -\overline{\frac{1}{2\pi}\int h(-\omega) \cdot e^{i\omega x} d\omega} = -\overline{h(-\omega)^{\vee}},$$

and

$$h(-\omega)^{\vee} = \frac{1}{2\pi}\int h(-\omega) \cdot e^{i\omega x} d\omega = -\frac{1}{2\pi}\int h(\omega) \cdot e^{-i\omega x} d\omega = -\frac{1}{2\pi}\int h(\omega) \cdot e^{i\omega(-x)} d\omega = -h^{\vee}(-x).$$

Combining these, it can be shown that

$$L_{-(n+1)}(x) = (\rho_{-(n+1)})^{\vee}(x) = (\overline{\rho_n})^{\vee}(x) = -\overline{\rho_n(-\omega)^{\vee}} = \overline{\rho_n}^{\vee}(-x) = \overline{L_n(-x)} = L_n(-x). \qquad (11)$$

The last equality is true since $L_n(x)$ is real.

In conclusion, $L_n(x)$ for $n < 0$ has positive support and is related to the $L_n(x)$ with $n \geq 0$ as shown in equation (11).

The Laguerre method uses equation (9) to find the coefficients for the Laguerre expansion of a function. As described in equation (9), these coefficients are obtained from the Fourier transform of $g(\theta)$. By using these coefficients, the function $f(x)$ can be reconstructed through the use of equation (7).

2.3. Hilbert method:

This method of reconstruction uses the Laguerre expansion discussed previously and the Riesz projection, also known as the Hilbert projection.

The Hilbert projections project the square integrable functions (L^2) onto the Hardy spaces H_2^+ and H_2^-. The Hardy space, H_2^+, contains functions which extend to be analytic in the upper complex half-plane. They are the Fourier transforms of functions with positive support in the x-domain. H_2^- consists of functions which extend to be analytic in the lower complex half plane; the Fourier transforms of functions with negative support. For the current analysis, $P^+[\varphi(\omega)]$ and $P^-[\varphi(\omega)]$ are used to represent orthogonal projections in L^2 onto the Hardy spaces H_2^+ and H_2^- respectively [1]. These spaces are defined as

$$H_2^+ = \left\{ g^{\wedge}(\omega) \in L^2(\mathbf{R}) : (g^{\wedge}(\omega))^{\vee} = g(x) = 0 \text{ for } x < 0 \right\}$$

and

$$H_2^- = \left\{ g^{\wedge}(\omega) \in L^2(\mathbf{R}) : (g^{\wedge}(\omega))^{\vee} = g(x) = 0 \text{ for } x > 0 \right\}$$

The projection mentioned before will be used to separate $\varphi(\omega) = f^{\wedge}(\omega)$, into components with disjoint positive and negative support in the x-domain respectively. This means that

$$\varphi(\omega) = P^+[\varphi(\omega)] + P^-[\varphi(\omega)]. \qquad (12)$$

That is to say,

$$f(x) = \left\{ P^+[\varphi(\omega)] \right\}^{\vee} + \left\{ P^-[\varphi(\omega)] \right\}^{\vee}. \qquad (13)$$

This is due to the orthogonal splitting of $L^2(R)$.

For the particular case of a function $f(x)$ of negative support,

$P^+[f^\wedge(\omega)] = 0$

$P^-[f^\wedge(\omega)] = f^\wedge(\omega).$

The way in which these projections will be implemented is through the use of the Laguerre expansion which is defined in the previous section. As was noted before, the Laguerre coefficients with $n \geq 0$ were defined so they correspond to Laguerre functions with negative support, and $n < 0$ corresponds to Laguerre functions with positive support. $P^-[f^\wedge(\omega)]$ can be obtained by using equation (6) with $n \geq 0$, which assures that only the Laguerre functions with negative support are used. In the same way, $P^+[f^\wedge(\omega)]$ can be computed by using equation (6) with $n < 0$.

For this particular method, a negatively supported function will be approximated as

$$f(x) \approx \sum_{m=1}^{\infty} c_m u_m(x), \tag{14}$$

where $u_m(x)$ is defined as

$$u_m(x) = \begin{cases} 1 & x \in (-m \cdot \Delta x, -(m-1) \cdot \Delta x] \\ 0 & \text{elsewhere} \end{cases} \tag{15}$$

In other words, the function is approximated with rectangular pulses of width Δx.

The basic idea behind this method is to shift the function $f(x)$ to the right by Δx, which has the following impact in its Fourier transform (see table 2):

$$f(x-\Delta x)^\wedge = f^\wedge(\omega) \cdot e^{-i \cdot \omega \cdot \Delta x}. \tag{16}$$

The function $F_1(\omega)$, which will be defined as

$F_1(\omega) = P^-[f^\wedge(\omega)] \cdot e^{-i \cdot \omega \cdot \Delta x} = f^\wedge(\omega) \cdot e^{-i \cdot \omega \cdot \Delta x}$

with some positive support for its inverse transform. By examining equations (14) and (16), it can be concluded that

$$P^+[F_1(\omega)] = c_1 \cdot u_0{}^\wedge(\omega). \tag{17}$$

Therefore, the value of c_1 can be obtained from equation (17).

In a similar way, $F_2(\omega)$ can be defined as

$F_2(\omega) = P^-[F_1(\omega)] \cdot e^{-i \cdot \omega \cdot \Delta x}$.

As in the previous case, it can also be concluded that

$P^+[F_2(\omega)] = c_2 \cdot u_0{}^\wedge(\omega)$.

In conclusion, a function with negative support can be approximate using equation (14), where the coefficients can be obtained using

$$F_m(\omega) = P^-[F_{m-1}(\omega)] \cdot e^{-i \cdot \omega \cdot \Delta x} \tag{18}$$

with

$$c_m = P^+[F_m(\omega)] / u_0{}^\wedge(\omega) \tag{19}$$

for $m > 0$ and $F_0(\omega) = f^\wedge(\omega)$. The projections are computed using the Laguerre expansion as described above.

3. Results

The methods will be examined by comparing the reconstruction of functions from their transforms in frequency domain. Two main study cases will be compared by looking at the L^2 error given by

$$err = \sqrt{\int |f(x) - f_r(x)|^2 \, dx}, \tag{20}$$

where $f_r(x)$ is the reconstructed function, $f(x)$ is the exact function, and the limits of the integration are given by the interval in which the reconstruction is computed.

There will be a choice of parameters when comparing the three methods for reconstructing a function. One of these parameters is the total number of samples taken in the frequency domain (N). This parameter is used in all the methods. The reconstruction of the function will have to be limited to a finite interval $[-T, 0]$, for some positive number T. Also, the distance between samples reconstructed in the x-domain (Δx) will have to be specified. For the cases considered, $\Delta x = T/(2N)$ has been chosen. This parameter affects the Hilbert method the most, since this method depends on approximating this function by rectangular pulses of width Δx. Finally, the IDFT method requires a frequency range (W_0). For this method, the transformed function in the frequency domain will be sampled in the interval $[-W_0/2, W_0/2]$.

3.1. Square pulse reconstruction:

The function to be reconstructed is a square pulse in x of total width 2^{k+1} (where k is an integer), with equation

$$f(x) = \begin{cases} 1, & |x| < 2^k \\ 0, & \text{elsewhere.} \end{cases}$$

The transform of this function is

$$f\hat{}(\omega) = 2\sin(2^k\omega)/\omega .$$

(See [5] for more details.)

The parameters used for reconstruction are $N = 2^8$ (total number of sample points), $T = 150$ (x range of reconstruction), $\Delta x = 0.2930$ (x step of reconstruction), and $W_0 = 8$ (frequency range for IDFT).

Figure 1 shows the results for $k = 5$. The dashed line curves represent the exact representation of the function and the lighter curve is the real component of the reconstructed functions. The Laguerre and Hilbert methods generate high frequency oscillations that do not match the exact representation of the function. These oscillations are very similar for both methods and we believe it to be a manifestation of the Gibbs phenomenon. It occurs because both methods split f(x), which is continuous at zero, into a sum of discontinuous functions, one with positive and one with negative support. Fourier representations of discontinuous functions typically exhibit this oscillatory phenomenon. The IDFT (top plot) does not split f in this way and therefore does not exhibit this shortcoming; it gives the best results for the reconstruction. However, in Figure 2 and Figure 3, it becomes apparent what the limitations of the IDFT are. In particular, Figure 2 ($k = 7$) shows a mismatch on the reconstruction due to the periodic repetition of the results for the IDFT on the interval $[-2\pi \cdot N/W_0, 0] = [-201.06, 0]$. See section 2.1 for more details.

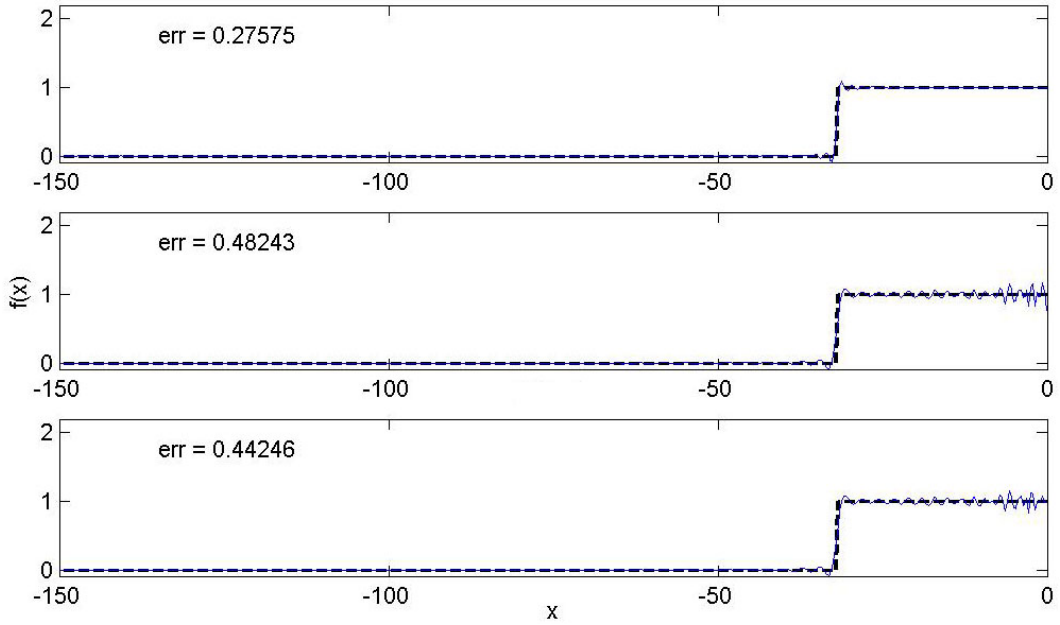


Figure 1 IDFT (top), Laguerre (middle) and Hilbert (bottom) reconstruction of square wave with $k = 5$.

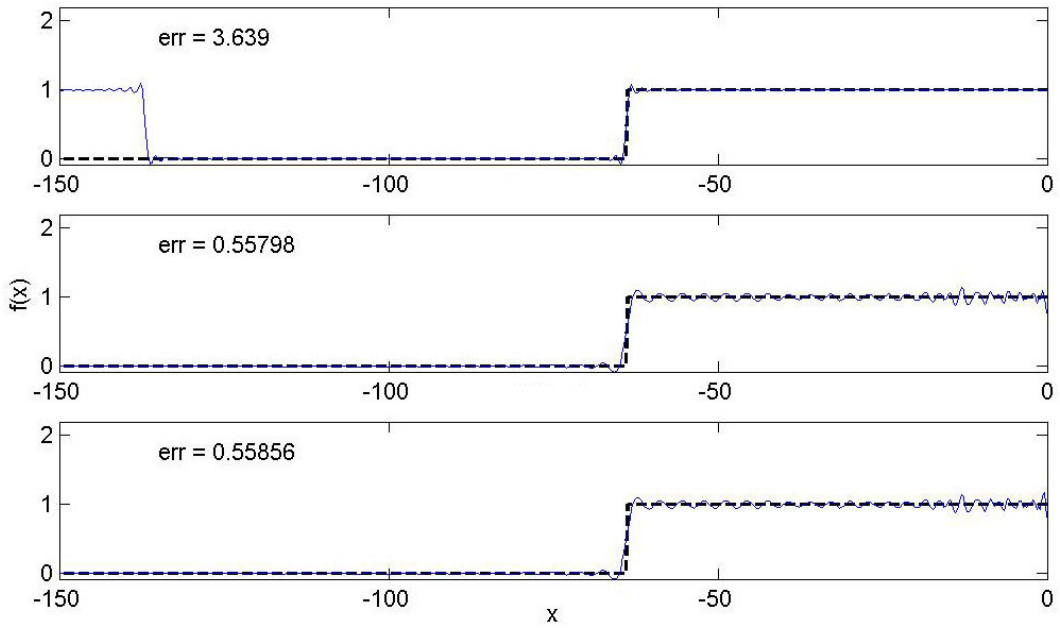


Figure 2 IDFT (top), Laguerre (middle) and Hilbert (bottom) reconstruction of square wave with $k = 6$.

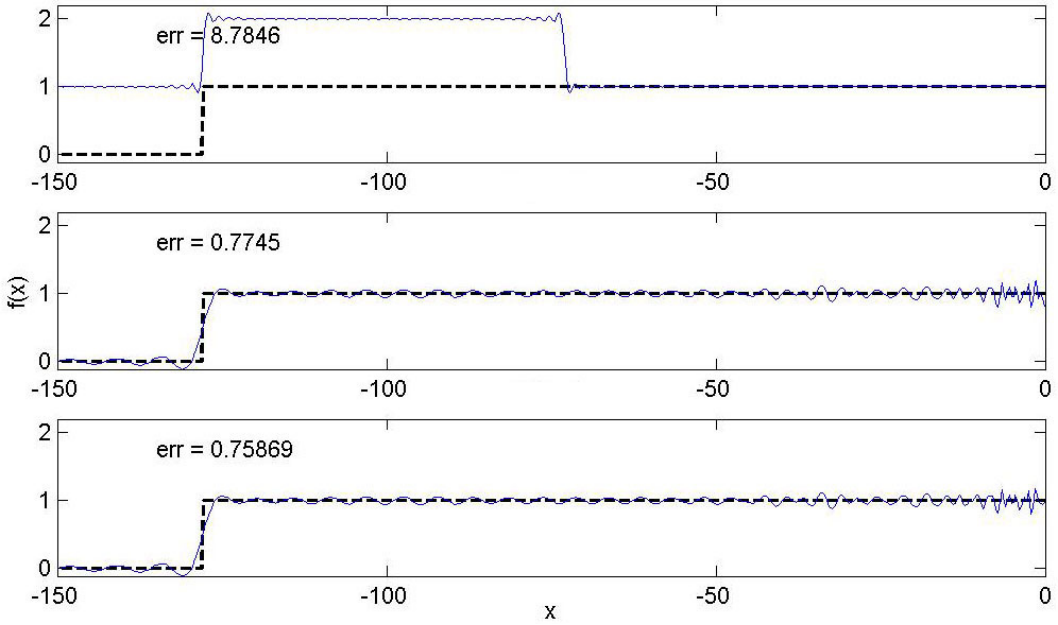


Figure 3 IDFT (top), Laguerre (middle) and Hilbert (bottom) reconstruction of square wave with $k = 7$.

3.2. Damped cosine reconstruction:

The function to be reconstructed is an exponentially decaying cosine as x goes to negative infinity defined as

$$f(x) = \begin{cases} e^{x/2}\cos(5x), & x \le 0 \\ 0, & \text{elsewhere} \end{cases}$$

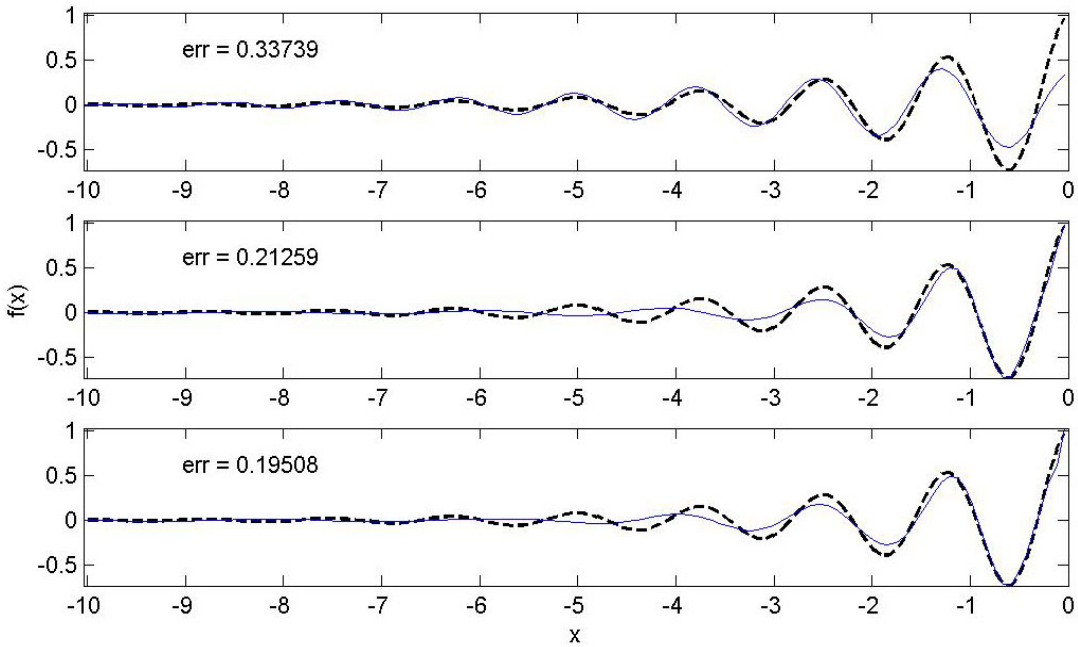


Figure 4 IDFT (top), Laguerre (middle) and Hilbert (bottom) reconstruction of exponentially decreasing cosine wave.

The Fourier transform of this function is:

$$f\hat{}(\omega) = \frac{2(1-2i\omega)}{100+(1-2i\omega)^2}.$$

(See [8] for more details)

The parameters used for reconstruction are $N = 2^6$ (total number of sample points), $T = 10$ (x range of reconstruction), $\Delta x = 0.0781$ (x step of reconstruction), and $W_0 = 11$ (frequency range for IDFT).

For this function and other functions that have finite or rapidly convergent expansions in the ρ_n's, the Laguerre and Hilbert approaches will succeed. Because of the reciprocity relation, which necessitates periodicity, the IDFT will never do well on functions which decay slowly for large ω.

For this particular set of results (see Figure 4), the Laguerre and Hilbert method give the best approximation. By increasing the number of sample points, N, the reconstructed functions for these two methods would get closer to the exact representation of the function. However, the IDFT method does not show any noticeable improvement due to truncation error. The Hilbert and, especially, the Laguerre method yield better results since they use a $O(1/\omega)$ basis in the frequency domain to approximate a $O(1/\omega)$ transform of the function to be recovered.

4. Conclusion

This paper shows different ways of computing the inverse Fourier transform, which can be extended to compute the Fourier transform, by using the Laguerre expansion of a function. The Laguerre and Hilbert methods work best for exponentially decreasing functions and have certain advantages over the well-known IDFT method which limits the higher frequency for reconstruction. The limitation on the later method is due to the relation between the frequency range W_0 sampled and the period T of the function recovered. Due to this limitation, as observed in figures 1 and 2, the IDFT does not always give the correct results for certain applications. However, the IDFT does give the best results for reconstructing a function from its Fourier transform if something is known about the range of the frequency spectrum.

The Laguerre and Hilbert methods compute the inverse Fourier transform by sampling over a larger frequency span. Due to this feature, these methods are more convenient for reconstructing functions with a broad frequency support. Both of these methods are very appealing for possible applications in scattering problems, in particular the Hilbert method since it performs the reconstruction in a layer-by-layer fashion.

5. References

[1] Dym, H. & McKean, H.P. (1997). Fourier Series and Integrals. San Diego, CA: Academic Press.

[2] Weideman, J.A.C. "Computing the Hilbert Transform on the Real Line", Mathematics of Computation, Vol. 64, No. 210, (1995), pp 745-762.

[3] Andrews, L.C. (1984). Special Functions for Engineers and Applied Mathematics. New York: Macmillan Publishing Company.

[4] Briggs, W. & Henson, V. E. (1995), The DFT: An Owner's Manual for the Discrete Fourier Transform, Philadelphia: Society for Industrial and Applied Mathematics (SIAM).

[5] Folland, G.B. (1992). Fourier Analysis and its Applications, California: Brooks/Cole Publishing Company.

[6] Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). Numerical Recipes in C. 2nd Edition. New York: Cambridge University Press.

[7] Lathi, B.P. (1998). Signal Processing & Linear Systems. California: Berkeley Cambridge Press.

[8] Tao, H. (2004). Signals & Systems – Reference Tables. [Online]
http://www.soe.ucsc.edu/classes/cmpe163/winter04/Lec23Ap.pdf

6. Appendix: Code for Reconstruction Methods

6.1. mLag.m

This function generates the Laguerre functions using the recursive relation given in equation (10).

```
function Ln = lag(N,x)

L0 = exp(x).*(x<=0);
Ln(1,:) = (-2*x - 1).*exp(x).*(x(1,:)<=0);
Ln(2,:) = ((-3-2*x).*Ln(1,:)-L0)/2.*(x<=0);

for n = 2:N
    Ln(n+1,:) = (-(2*n+1+2*x).*Ln(n,:)-n*Ln(n-1,:))/(n+1);
end

Ln = [L0; Ln];
```

6.2. IDFT.m

It performs the reconstruction of a function using the IDFT method.

```
function f = IDFT(x,w,fhat)

f = zeros(size(x));
dw = abs(w(2)-w(1));

for n = 1:length(w)
    f = f + (dw/(2*pi))*fhat(n)*exp(i*w(n)*x);
end
```

6.3. Lrec.m

It performs the reconstruction of a function using the Laguerre method.

```
function fp = funRec(x,w,fhat)

N = length(w)/2;
g = -(i/2*pi)*fhat.*(i+w);

fr = fft(fftshift(g))/N/pi;
fr = fr.*exp(-i*(0:2*N-1)*pi/N/2); %%% Taking into account
                                   %%% theta shifting.

%%% Computing Laguerre Functions
disp('Computing Laguerre Functions ...');
L = nLag(length(fr),x);
disp('After Computing Laguerre Functions');
dim = size(L);

fp = zeros(size(x));
for n = 1:length(fr)/2
    fp = fp + fr(n)*L(n,:);
end
```

6.4. Hrec.m

It performs the reconstruction of a function using the Hilbert method.

```
function fr = Hrec(x,w,fhat)

W = abs(x(1)-x(2));
fshift = fhat;
N_shift = length(x)+1;  %%% Computing No. of points for
                                  %%% shifting to get to end of x.
%%% Computing Heavyhat transform
s = exp(i*w*W);
Heavyhat = (1-1./s)./(i*w);

p1 = 0;
for n = 1:N_shift
    %%% A = P+(f^); B = P-(f^):
    [A,B] = proj(fshift,w);
    %%% Fitting Projection to Heavyhat:
     fr(n) = A/Heavyhat;
    %%% Applying shift: exp(-i*w*d) P-(f^):
    fshift = exp(-i*w*W).*B;
    %%% Displaying Percentage Complete
    p2 = floor(n/N_shift*20);
    if p1~=p2
        disp([num2str(5*p2),' %']);
    end
    p1 = p2;
end
fr = fr(end:-1:2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [Pplus,Pminus] = proj(fhat,w)
% [Pplus,Pminus] = proj(f)
%

N = length(fhat)/2;
g = -(i/2*pi)*fhat.*(i+w);

fr = fft(fftshift(g))/N/pi;
frp = [fr(1:N) zeros(1,N)]; %%% Taking the negative support
Pminus = N*pi*fftshift(ifft(frp))./(-(i/2*pi)*(i+w));

Pplus = fhat - Pminus;.
```

6.5. ILH.m

This function calls and plots the results for the different reconstruction methods.

```
%%% COMPUTING TRANSFORMS OF FUNCTION
w2 = mfreq(N/2);
```

```
w1 = (-w0/2:w0/N:w0/2-w0/N);
fhat1 = feval(ffhandle,w1);
fhat2 = feval(ffhandle,w2);

%%% RECONSTRUCTING FUNCTION
m = ceil(L/W); x = (-m*W-W/2:W:-W/2);
tic, f1 = IDFT(x,w1,fhat1); t1 = toc;
tic, f2 = Lrec(x,w2,fhat2); t2 = toc;
tic, f3 = Hrec(x,w2,fhat2); t3 = toc;

%%% COMPUTING EXACT OR APPROXIMATED FUNCTION FOR COMPARISON
if ~isempty(fstring)
    f = feval(fhandle,x);
elseif flag_IDFT
    f = IDFT(x,(-300:0.02:300),feval(ffhandle,(-300:0.02:300)));
else
    f = nan * zeros(size(x)); %%nan's don't plot
end

%%% COMPUTING ERRORS
err_IDFT = sqrt(W*sum(abs(f-f1).^2));
err_Lrec = sqrt(W*sum(abs(f-f2).^2));
err_Hrec = sqrt(W*sum(abs(f-f3).^2));

%%% PLOTTING RECONSTRUCTED FUNCTIONS
figure(1)
subplot(4,2,1), plot(w1,real(fhat1),w1,imag(fhat1));
title('Evenly Spaced Frequency');
subplot(4,2,2), plot(w2,real(fhat2),w2,imag(fhat2));
title('Evenly Spaced Theta');
y1 = max([max(real(f)) max(real(f1)) max(real(f2)) max(real(f3))]);
y2 = min([min(real(f)) min(real(f1)) min(real(f2)) min(real(f3))]);
subplot(4-0,1,2-0), plot(x,real(f),'k--','linewidth',1.5);
hold on; plot(x,real(f1),'b'); hold off;
axis([min(x) 0 y2 y1]);
text(0.9*min(x),0.6*y1,['err = ',num2str(err_IDFT)]);
subplot(4-0,1,3-0), plot(x,real(f),'k--','linewidth',1.5);
hold on; plot(x,real(f2),'b'); hold off; ylabel('f(x)');
axis([min(x) 0 y2 y1]);
text(0.9*min(x),0.6*y1,['err = ',num2str(err_Lrec)]);
subplot(4-0,1,4-0), plot(x,real(f),'k--','linewidth',1.5);
hold on; plot(x,real(f3),'b'); hold off;
axis([min(x) 0 y2 y1]); xlabel('x');
text(0.9*min(x),0.6*y1,['err = ',num2str(err_Hrec)]);

%%% DISPLAYING ERRORS
disp(' ');
disp('Method || Time Elapsed || Error');
disp('-----------------------------');
disp(['IDFT   || ',num2str(t1),'   ||  ',num2str(err_IDFT)]);
disp(['Lrec   || ',num2str(t2),'   ||  ',num2str(err_Lrec)]);
disp(['Hrec   || ',num2str(t3),'   ||  ',num2str(err_Hrec)]);
```

6.6. FRun

This function sets the parameters and functions for the ILH function.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% SETTING PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Total number of points in frequency domain (must be even)
N= 2^6;
%%% Computing length of reconstruction
L = 10;
%%% Delta Width for recovered data
W = L/(2*N);
%%% Range of Frequency for DFT method
w0 = 11;
%%% Flag telling to use IDFT approximation of exact solution
%%% when none is provided.
flag_IDFT = 1;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% SAMPLE FUNCTIONS USED FOR RECONSTRUCTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fstring = [];

fstring = 'exp(x/2).*cos(10*x/2)';
fhatstring = '2*(1-i*2*w)./(100+(1-i*2*w).^2)';

%k = 7;
%fstring = ['abs(x)<2^' int2str(k)];
%fhatstring = [int2str(2*2^k) '*sinc((' int2str(2^k) ')*w/pi)'];



if ~isempty(fstring)
    fhandle = inline(fstring);
end
ffhandle = inline(fhatstring);

ILH; figure(1);
```