Problem 1. [20 pts.] *Bandwidth Considerations for Discrete-Time Simulation of Continuous-Time Systems.*

In order to numerically simulate a continuous-time communication system or phase locked loop, we need to understand the relationship between continuous-time and discrete-time systems and the sampling theorem, which connects the two. The following equations are given to help you review.

Let $x(t) \leftrightarrow X(f)$ be a continuous time Fourier Transform (CTFT) pair. Suppose that an idealized sampled-data signal $x_s(t)$ is created by multiplying $x(t)$ by a train of Dirac impulses, spaced uniformly in time, i.e.,

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s). \tag{1}$$

The sampled-data continuous-time signal $x_s(t)$ is uniquely determined from knowledge of the samples $\{x(nT_s) : n \in \mathcal{Z}\}$ and the sampling interval T_s. The sequence made up of the samples $x_d[n] = x(nT_s)$, $n \in \mathcal{Z}$, is a discrete-time signal.

There are two equivalent ways to compute the continuous-time Fourier Transform of the sampled data signal $x_s(t)$.

- The first is based on the first equality in Eq. (1). It uses the *multiplication-in-time* $\leftrightarrow$ *convolution-in-frequency* property and yields

$$X_s(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(f + k/T_s). \tag{2}$$

  In otherwords, $X_s(f)$ is a scaled, superposition of replicas of the original spectrum, each shifted to a multiple of the sampling frequency $f_s = 1/T_s$.

- The second way to compute $X_s(f)$ uses the second equality in Eq. (1), linearity of the Fourier Transform, the Fourier Transform of the Dirac delta, and the *time-shifting* property. This yields

$$X_s(f) = \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi(fT_s)n}. \tag{3}$$

- Eqs. (2) and (3) must be equal. But Eq. (3) is actually the discrete-time Fourier Transform (DTFT)[2] of the sample sequence $x_d[n]$ evaluated at $\lambda = fT_s$.

The *Sampling Theorem* says that if the signal $x(t)$ is strictly band-limited, i.e., if $X(f) = 0$ for $|f| > B$, and if sampling is done at a rate more than twice the highest frequency

$$f_s > 2B$$

[2] A DTFT pair is defined by

$$x_d[n] = \int_{-1/2}^{1/2} X_d(\lambda) e^{j2\pi\lambda n} \leftrightarrow X_d(\lambda) = \sum_{n=-\infty}^{\infty} x_d[n] e^{-j2\pi\lambda n}.$$

then there is no aliasing and the CTFT is uniquely determined from the DTFT of the samples by selecting its fundamental period ...

$$X(f) = \begin{cases} T_s X_d(fT_s) & |f| \leq 1/(2T_s) \\ 0 & |f| \leq 1/(2T_s) \end{cases}.$$

Equivalently, the original time-domain waveform can be exactly reconstructed from its samples via the interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x_d[n] \frac{\sin(\pi(t - nT_s)/T_s)}{\pi(t - nT_s)/T_s}.$$

(a) In the real world it is very unusual to have signals that are exactly band-limited. In fact, it is impossible for a signal to be simultaneously time-limited and band-limited. Yet, all real world signals are approximately time-limited and band-limited. Take the example of Gaussian pulses ...

$$x(t) = e^{-\pi(t/\tau)^2} \leftrightarrow X(f) = \tau e^{-\pi(f\tau)^2}.$$

The above CTFT pair is neither time-limited nor band-limited. Use Matlab to plot for some sample values of the parameter τ and suggest reasonable practical limits on the time-domain and frequency-domain extents.

(b) From the plots of Part (a) it is apparent that the tails in frequency or time fall off very rapidly. In this part you will step through the process of creating a bound on aliasing error as a function of the sampling frequency relative to the bandwidth parameter $1/\tau$. The worst case aliasing occurs at the foldover frequency $f = 1/(2T_s) = f_s/2$. Evaluating Eq. (2) at $f = f_s/2$

$$X_s(f_s/2) = f_s \sum_{n=-\infty}^{\infty} X(f_s/2 + kf_s).$$

Since the unsampled value of the spectrum at the foldover frequency is $X(f_s/2)$ we can use the above to write the total error as

$$\mathcal{E}_{total} = f_s^{-1} X_s(f_s/2) - X(f_s/2) = 2 \sum_{n=1}^{\infty} X((k + 1/2)f_s))$$

and the relative error would then be

$$\mathcal{E} = \mathcal{E}_{total}/X(f_s/2) = \frac{2}{X(f_s/2)} \sum_{n=1}^{\infty} X((k + 1/2)f_s)).$$

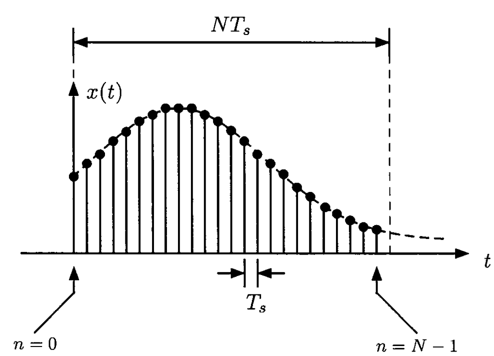
For the Gaussian pulse find a bound (maybe very loose) on the relative error and use Matlab to plot it as a function of the relative sampling rate $f_s\tau$.

In the case of Gaussian pulses we have an exact formula for the pulse in the time-domain and in the frequency domain. There are only a few other examples where exact formulas are available – see a typical table of Fourier Transforms. But real world signals never look like this, nor do we really have formulas for them in the time-domain. For such, a numerical solution for the CTFT is the only option. Let $x(t) \leftrightarrow X(f)$ be a CTFT pair. Assume that signals have finite energy, which then implies that tails die off for either $|t|$, $|f| \to \infty$. If, in addition, we assume that the time function is causal, then we might approximate its CTFT by truncation:

$$X(f) \approx \int_0^{T_0} x(t)e^{-j2\pi ft}\mathrm{d}t.$$

Discretize the integral by sampling at $t = nT_s$ and let $T_0 = NT_s$ as shown in the figure below. Then we may approximate the truncated integral with a Riemann sum ...

$$X(f) \approx \underbrace{T_s \sum_{n=0}^{N-1} x(nT_s)e^{-j2\pi fnT_s}}_{T_s\tilde{X}(fT_s)}.$$



- Note that $\tilde{X}(fT_s)$ is a DTFT and therefore periodic in f of period $1/T_s$.
- Since the original CTFT $X(f)$ was not periodic, the above approximation can only hold over a restricted range of f, namely, it can only hold for frequencies less than the foldover frequency ...

$$|f| \leq 1/(2T_s).$$

- Now $\tilde{X}(\cdot)$ is the DTFT of the length N sequence defined by $\tilde{x}[n] = x(nT_s)$, $n = 0, 1, \ldots, N-1$. If we sample it in frequency at $f = k/(NT_s)$ ...

$$\begin{aligned} X(k/NT_s) &\approx T\sum_{n=0}^{N-1} x(nT_s)e^{-j2\pi kn/N} \\ &\approx T_s\tilde{X}_k \end{aligned}$$

for $k = -(N/2)+1, \ldots, -1, 0, 1, \ldots, N/2$ where $\tilde{X}_k$ is the N-point DFT of the sequence $\tilde{x}[n]$. (Above assumes N is even.)

(c) If you had a time-domain signal $y(t)$, which was not causal, but supported on a finite interval, say $[-T_1, T_2]$ where T_1, $T_2 > 0$, then how could you use the previous idea and a DFT to approximate the Fourier Transform $Y(f)$?

(d) Now use what you've learned in the previous parts to experiment with numerical calculation of the CTFT of the Gaussian pulse using sampling, time-windowing, and DFT computation. Compare the numerical calculation with the exact formula above for the CTFT of the Gaussian pulse.

Problem 1

(a) $x(t) = e^{-\pi (t/\tau)^2} \longleftrightarrow X(f) = \tau e^{-\pi (f\tau)^2}$

Plots are given for several values of τ. Main take away is that tails fall of very quickly in time and frequency domains.

(b) To evaluate $\mathcal{E}$ we need

$$X(f_s/2) = \tau e^{-\pi (f_s \tau/2)^2} = \tau e^{-\frac{\pi}{4}(f_s \tau)^2}$$

$$X((k+\tfrac{1}{2})f_s) = X((2k+1)f_s/2)$$

$$= \tau e^{-\frac{\pi}{4}((2k+1)f_s\tau)^2}$$

Let $E \triangleq e^{-\frac{\pi}{4}(f_s\tau)^2} \Longrightarrow X(f_s/2) = \tau E$ and

$$X((k+\tfrac{1}{2})f_s) = \tau E^{(2k+1)^2}$$

$$\therefore \quad \mathcal{E} = \frac{2}{\tau E} \sum_{k=1}^{\infty} \tau E^{(2k+1)^2}$$

$$= 2 \sum_{k=1}^{\infty} E^{(2k+1)^2 - 1} = 2 \sum_{k=1}^{\infty} E^{4k(k+1)}$$

Note that the powers of E amount to a very sparse sub-sampling of the powers present in a geometric series ...

k	$4k(k+1)$
1	8
2	24
3	48
4	80
⋮	

Since $0 < E < 1$ we can bound the sum with a geometric series

$$\mathcal{E} = 2\sum_{k=1}^{\infty} E^{4k(k+1)} \leq 2\sum_{\ell=1}^{\infty} E^{\ell}$$

$$= 2\left(\frac{1}{1-E} - 1\right)$$

$$= 2\frac{E}{1-E}$$

Plotting this bound as a function of $f_s\tau$ we see how fast it goes to zero.

(c) $y(t)$ only nonzero on $[-T_1, T_2]$ or approx. so ... where $T_1, T_2 > 0$. Define

$$x(t) = y(t - T_1)$$

Then $x(t)$ is only nonzero on $[0, T_1 + T_2]$ for which we can apply the approx of the DFT for a causal signal. From the time-shifting property

$$X(f) = e^{-j2\pi f T_1} Y(f)$$

$$\Longrightarrow \quad Y(f) = e^{+j2\pi f T_1} X(f).$$

Define $T_0 = T_1 + T_2$. Discretize $t = nT_s$ and let N be st. $T_0 = NT_s$.

Then samples of the CTFT $X(f)$ are found from

$$X(k/NT_s) \approx T_s \tilde{X}_k$$

where $\tilde{X}_k$ is the N-point DFT of

$$\tilde{x}[n] = x(nT_s).$$

$$\tilde{X}_k = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi kn/N}$$

$$\updownarrow$$

$$\tilde{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_k e^{+j2\pi kn/N}$$

If $\tilde{x} = \left[\tilde{x}[0] \ \cdots \ \tilde{x}[N-1]\right]^T$ then

$$\tilde{X} = \mathrm{fft}(\tilde{x})$$

produces an $N \times 1$ vector containing

$$\tilde{X} = \left[\tilde{X}_0 \ \tilde{X}_1 \ \cdots \ \tilde{X}_{N-1}\right]^T$$

Note that Matlab indexes vectors starting from 1 rather than zero.

Contents

Bandwidth Considerations for DT Sim. of CT Systs. Problem 1.

ECE 440. Take Home Exam. Spring 2022.

File = THE_P1.m

`The basic Gaussian pulse`

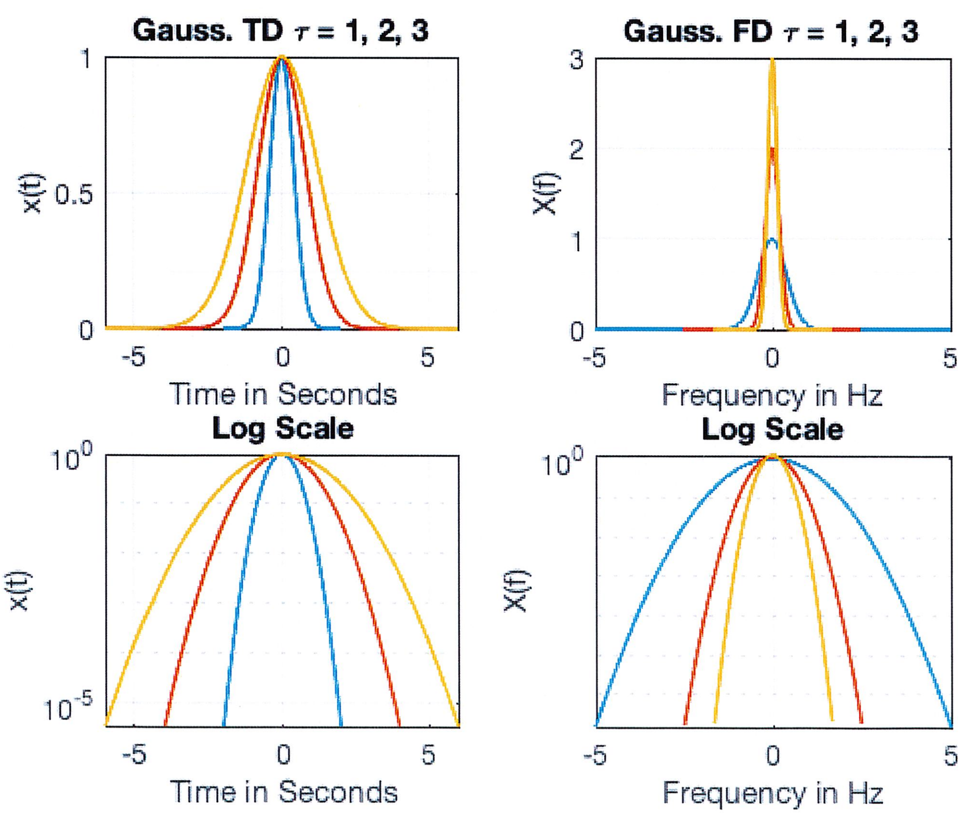
$x(t) = e^{-\pi(t/\tau)^2} \leftrightarrow X(f) = \tau e^{-\pi(f\tau)^2}$

```
close all
clear all
```

Part (a): Plots of Gaussian Pulses.

```
tau = [1, 2, 3];                                 %3 values. Sets the BW of the pulse

for i = 1:3

tmax = 2*tau(i);
tindex = -tmax:0.01:tmax;
h = exp(-pi*((tindex/tau(i)) .^ 2));    %Direct computation of time pulse

fmax = 5/tau(i);
findex = -fmax:0.01:fmax;
H = tau(i)*exp(-pi*((findex*tau(i)) .^ 2));    %Direct computation of freq. pulse

% Plots for comparing the time and frequency domain. Note we also plot the
% pulse in log domain in order to demonstrate the precision attainable with
% Matlab -- an IEEE double precision float.

figure(1)

subplot(2,2,1)
plot(tindex,h,'LineWidth',2)
grid
xlabel('Time in Seconds')
ylabel('x(t)')
%axis([-tmax tmax 0 1.1])
title(['Gauss. TD \tau = 1, 2, 3'])
set(gca,'Fontsize',14)

hold on

subplot(2,2,2)
plot(findex,H,'LineWidth',2)
grid
xlabel('Frequency in Hz')
ylabel('X(f)')
```

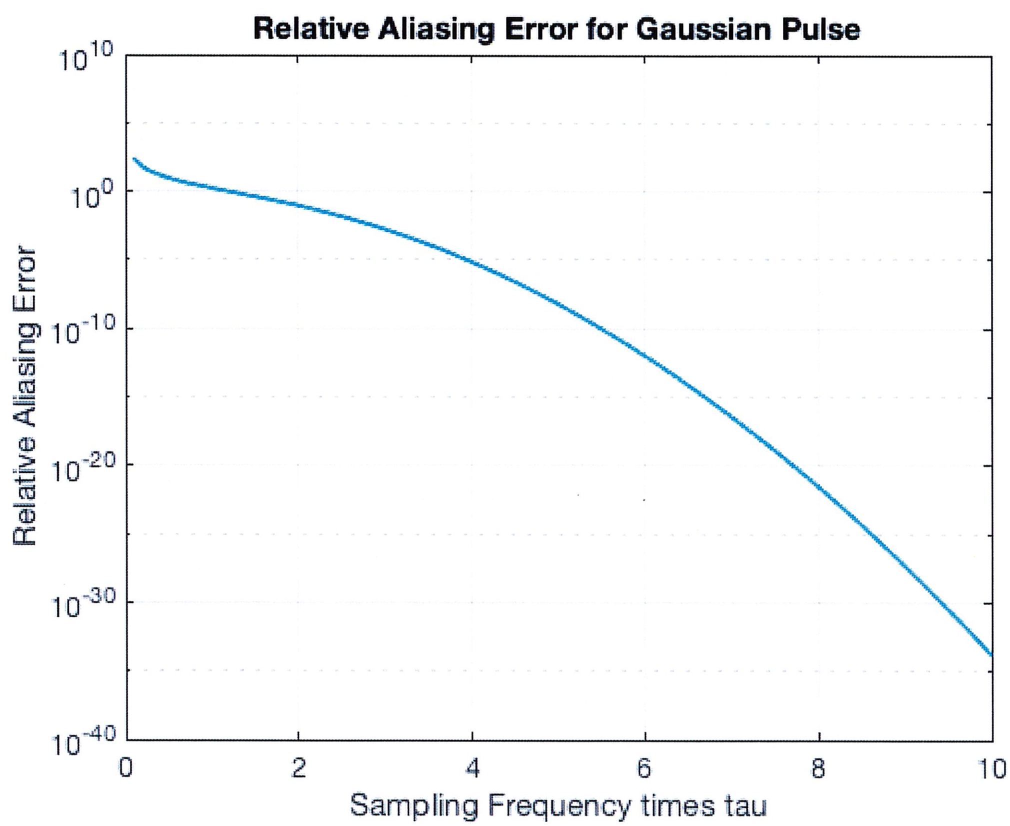
```
%axis([-fmax fmax 0 1.1*tau])
title(['Gauss. FD \tau = 1, 2, 3'])
set(gca,'Fontsize',14)

hold on

subplot(2,2,3)
semilogy(tindex,h,'LineWidth',2)
grid
xlabel('Time in Seconds')
ylabel('x(t)')
title('Log Scale')
set(gca,'Fontsize',14)

hold on

subplot(2,2,4)
semilogy(findex,H,'LineWidth',2)
grid
xlabel('Frequency in Hz')
ylabel('X(f)')
title('Log Scale')
set(gca,'Fontsize',14)

hold on

end
```



Part (b): Plotting the Bound.

```
%Code for computing the approximate bound on aliasing error for the
%Gaussian pulse
```

```
fsNorm = 0.1:0.1:10;
E = exp(-(pi/4)*(fsNorm .^ 2));
Erel = 2*E ./ (1-E);

figure(2)

semilogy(fsNorm,Erel,'LineWidth',2)
grid
xlabel('Sampling Frequency times tau')
ylabel('Relative Aliasing Error')
title('Relative Aliasing Error for Gaussian Pulse')
set(gca,'Fontsize',14)
```

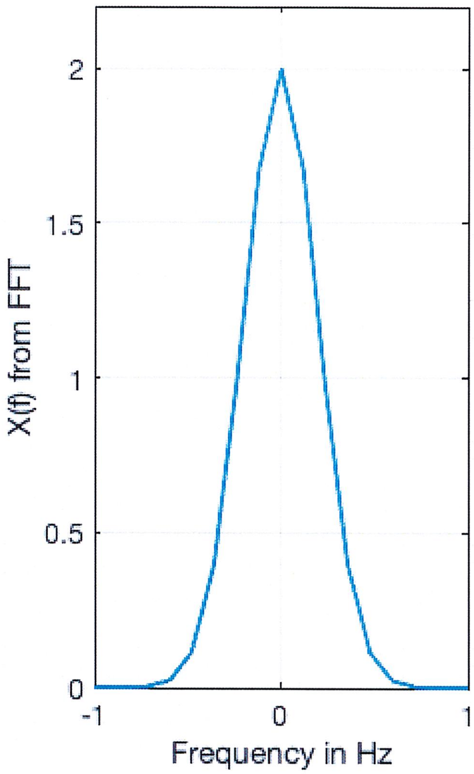


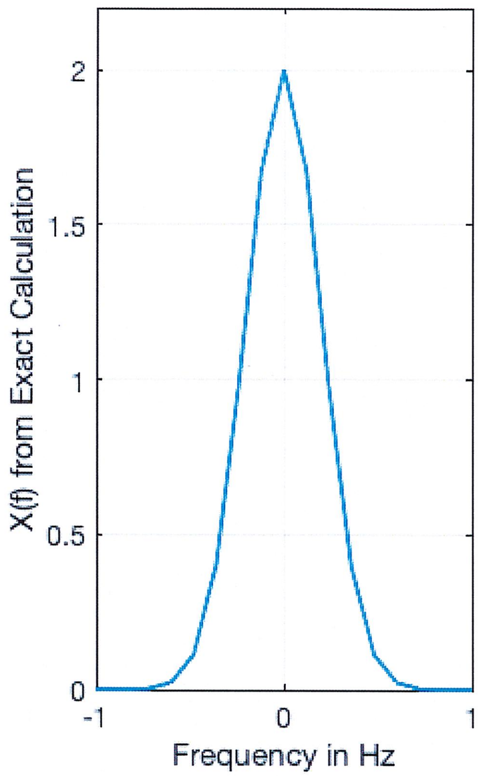
Part (d): Numerical calculation of CTFT and comparison

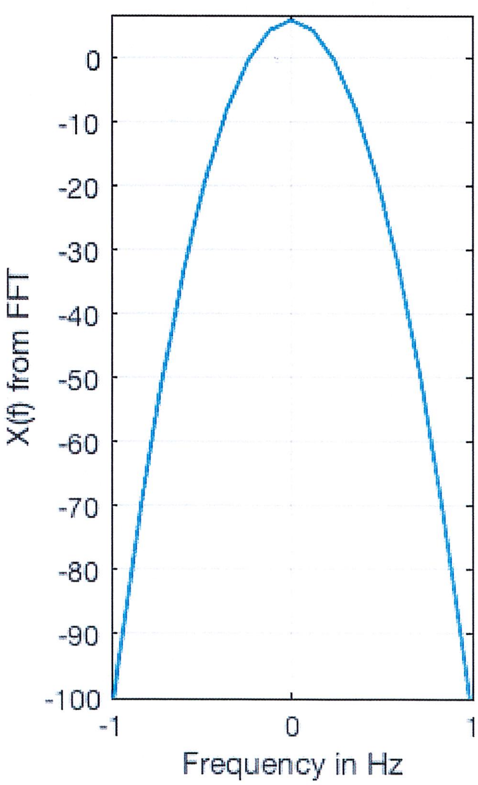
```
% The 3dB point on the Gaussian frequency pulse is the value of f where the
% magnitude response drops to 0.707 of is value at f = 0. The calculation
% below shows that f3dB is close to 1/tau.

tau = 2;
f3dB = sqrt(log(2)/2*pi)/tau;

% The bandwidth parameter tau is set above. Since the bandwidth is
% proportional to 1/tau, we will set the sampling rate in proportion to
% that. Start with a large oversampling factor, say 50.

fs = 50/tau;
Ts = 1/fs;

% Where to time window the time-domain pulse? Set an amplitude limit.
```

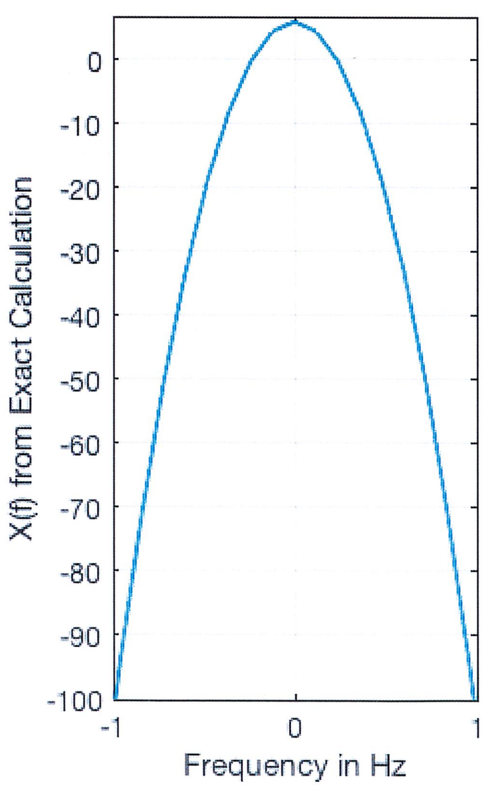
```
epsilon = 1e-6;
tmax = tau*sqrt(-log(epsilon)/pi);
tindex_p = 0:Ts:tmax;

tindex_n = -fliplr(tindex_p);
tindex_n(end) = [];
tindex = [tindex_n tindex_p];

h1 = exp(-pi*((tindex/tau) .^ 2));  %Direct computation of time pulse

%Approx. evaluation of CT Fourier transform using fft of windowed
%time samples.
H1_tmp = fft(h1);
H1 = Ts*fftshift(H1_tmp);           %Line fft samples in frequency.
H1_dB = 20*log10(abs(H1));

N = length(h1);

N_end = (N-1)/2;
N_start = -N_end;

nindex = N_start:N_end;                  %Discrete time index.

findex = nindex/(N*Ts);

%Exact evaluation of CT Fourier transform, evaluated at points in the new
%findex.

H_exact = tau*exp(-pi*((findex*tau) .^ 2));

H_exact_dB = 20*log10(abs(H_exact));

figure(3)

subplot(1,2,1)
plot(findex,abs(H1),'LineWidth',2)
axis([-2/tau 2/tau 0 1.1*tau])
grid
xlabel('Frequency in Hz')
ylabel('X(f) from FFT')
set(gca,'Fontsize',14)

subplot(1,2,2)
plot(findex,abs(H_exact),'LineWidth',2)
axis([-2/tau 2/tau 0 1.1*tau])
grid
xlabel('Frequency in Hz')
ylabel('X(f) from Exact Calculation')
set(gca,'Fontsize',14)

figure(4)

subplot(1,2,1)
plot(findex,H1_dB,'LineWidth',2)
axis([-2/tau 2/tau -100 20*log10(1.1*tau)])
grid
xlabel('Frequency in Hz')
ylabel('X(f) from FFT')
set(gca,'Fontsize',14)

subplot(1,2,2)
```

```
plot(findex,H_exact_dB,'LineWidth',2)
axis([-2/tau 2/tau -100 20*log10(1.1*tau)])
grid
xlabel('Frequency in Hz')
ylabel('X(f) from Exact Calculation')
set(gca,'Fontsize',14)
```


X(f) from FFT
Frequency in Hz


X(f) from Exact Calculation
Frequency in Hz


X(f) from FFT
Frequency in Hz


X(f) from Exact Calculation
Frequency in Hz