Spectrum estimation

Topics: DFT, FFT, Spectrogram, Spectral resolution

INTRODUCTION

A *spectrogram* is a plot of the spectral distribution (energy versus frequency) over time. The *spectrograms* of several signals are shown in Figure 1. These notes cover how a spectrogram is computed, and how it should be interpreted. The discussion will lead us to two important DSP topics: the DFT and the FFT.

We start by considering a signal of N samples: $x[0], x[1], \ldots x[N-1]$. Since sinusoids are building blocks for signals, we want to understand how much energy is contained in our signal at various sinusoidal frequencies. There cannot be infinitely many independent frequencies in only N samples—if there are N samples, then there are only N degrees of freedom in the signal, and so the signal can only have energy at N distinct frequencies. A complex sinusoid of frequency F cycles/sample is $\exp(j2\pi \text{ Fn})$, and what we want to find out is how to represent our signal as a sum of N weighted sinusoids.

We may proceed as follows. Since frequencies in sampled signals span a range of 0 to 1 cycle/sample, we may suppose that the N frequencies we need are equally spaced in this interval

$$\left\{0, \frac{1}{N}, \frac{2}{N}, \cdots \frac{N-1}{N}\right\}$$

Note that F = 0 is DC. What we want is to be able to write something like this:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j2\pi(k/N)n} \tag{1}$$

Here $X[k]$ is a complex-valued weight for the sinusoid $e^{j2\pi(k/N)n}$, whose frequency is k/N cycles/sample. The equation above expresses the signal $x[n]$ as a sum of N weighted sinusoids with equally spaced frequencies[1].

We need to figure out how to compute the weights in eq. (1). Once we do that, we know how much each sinusoid contributes to the signal. We will prove the following:

---

[1] The factor of $1/N$ in front accounts for spacing of frequencies being 1/N cycles/sample, and its role will be more obvious later.

$$X[k] = \sum_{m=0}^{N-1} x[m] e^{-j2\pi(k/N)m} \qquad (2)$$

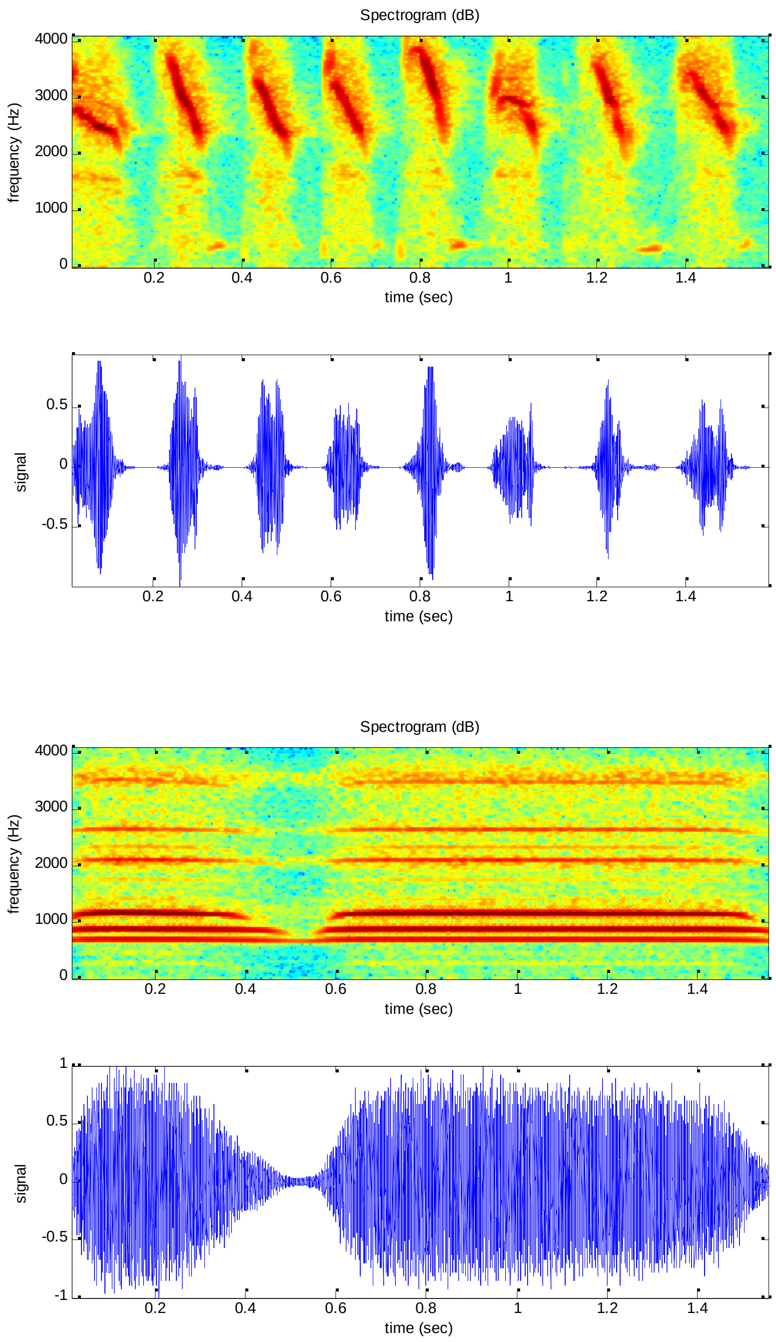


Figure 1. Spectrograms of: (a) series of 8 downward chirps; (b) two bursts of train whistle. Both were computed using the "showspectrogram" MATLAB function. Data sets are available in MATLAB—type "help audio".

Note how similar this equation looks to equation (1). Aside from a factor of $1/N$ in front and a negative in the exponential, the two equations are the same.

Now to prove eq. (2). It is pretty simple to show for any complex number $z \neq 1$, that

$$\sum_{n=0}^{N-1} z^n = \frac{1-z^N}{1-z}$$

As a result of this

$$\sum_{n=0}^{N-1} e^{j2\pi(p/N)n} = \begin{cases} N & p=0 \\ 0 & p \neq 0 \end{cases}$$

To prove eq. (2), we simply substitute the expression for $X[k]$ into the right side of eq. (1) and observe, after a little algebra, that we get back $x[n]$. Note that the factor of 1/N in eq. (1) is necessary for an exact inverse.

What we have proven establishes the discrete Fourier transform (DFT) pair:

DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi(k/N)n}$$

Inverse DFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi(k/N)n}$$

INTERPRETING THE DFT

The frequency points evaluated in the DFT are in units of cycles/sample. These correspond to the continuous-time frequencies by the simple formula

$$\frac{cycles}{second} = \frac{cycles}{sample} \times \frac{samples}{second}$$

For example, suppose N=100 samples of a real-valued continuous-time signal $x(t)$ are obtained at a sampling rate of 8192 Hz, and that the sampling rate is more than twice the maximum frequency in the signal. Then the 100 values of $X[k]$ obtained from the DFT are estimates[2] of the frequency spectrum of $x(t)$ at frequencies which are multiples of $8192/100 = 81.92$ Hz. Therefore, $X[0]$ is an estimate of the DC value of $x(t)$, $X[1]$ is

---

[2] The DFT values can only be *estimates* of the spectrum of $x(t)$ because they are based on a finite number of samples.

an estimate of the spectral energy of $x(t)$ at 81.92 Hz, $X[2]$ is an estimate of the spectral energy of $x(t)$ at $2*81.92=163.84$ Hz, etc. Similarly, $X[50]$ is an estimate of the energy at half the sampling rate, or 4096 Hz. But, because we assumed that the sampling rate is more than twice the maximum frequency in the signal, we must have $X[50]=0$

What does $X[51]$ represent? By assumption, there is no energy at or above 4096 Hz, so we might expect this to be zero as well. However, because we have a real-valued sampled signal $x[n]$, there exists a conjugate symmetry in the DFT values. Specifically, we can prove that $X[51]=\overline{X[49]}$, $X[52]=\overline{X[48]}$, .. $X[99]=\overline{X[1]}$, or more generally for any k that

$$X[N-k]=\overline{X[k]}=X[-k]$$

This is easy to prove by using the fact that $e^{-j2\pi(N-k)/N*n}=e^{j2\pi k/Nn}$. Therefore , half of the DFT values are complex conjugates of the other half when using a real-valued sampled signal $x[n]$. This makes sense, since we start with N real-valued samples in the signal, and should get the equivalent of N real numbers in the DFT values.

The relationship between $x(t)$ and the DFT points in this example is shown in Figure 2.

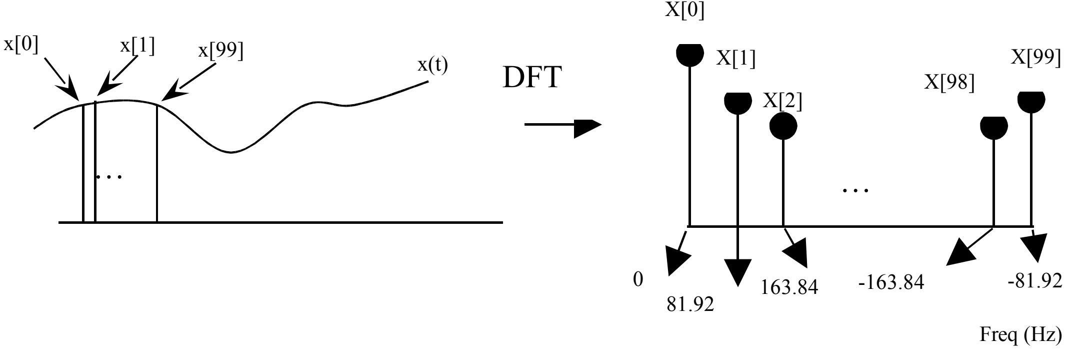
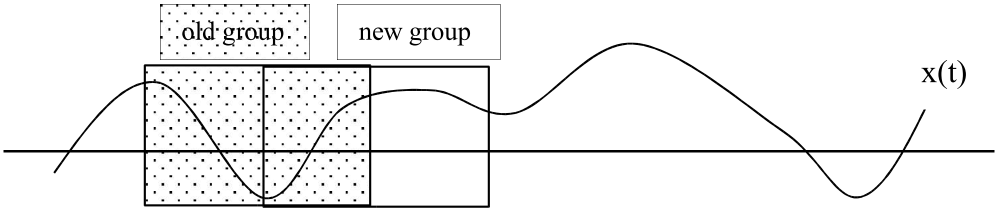


Figure 2: Relationship between samples of a continuous-time signal and estimates of its spectral energy. The sampling rate is 8192 Hz.

A spectrogram is computed by using the DFT on groups of N samples of the signal $x(t)$ at a time. The next group is samples is offset by N/2 to produce a smooth transition between groups[3], so that only half the samples in a group are new; see Figure 3.



---

[3] The offset does not need to be N/2; small offsets produce a smooth spectrogram, but require more computation, while large offsets result is abrupt changes between groups.

Figure 3: Two successive groups of N samples used in the spectrogram. Each group is offset by N/2.

Interpreting the DFT

The DFT of the following 100 point signal is shown in Figure 4:

$$x[n] = \sin(2\pi\,(1/\sqrt{10})n), \qquad n = 0,1,\cdots,99$$

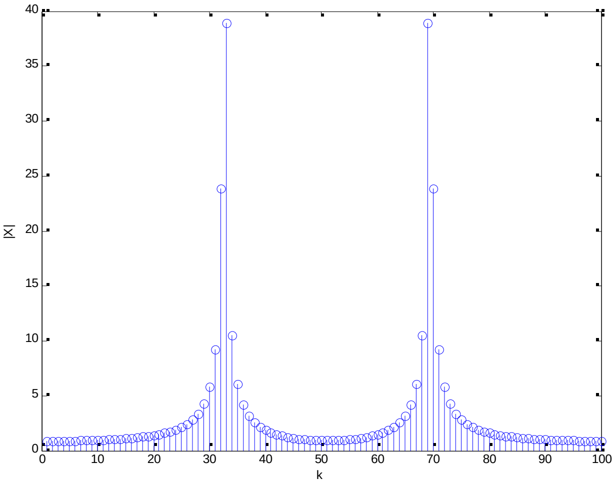


Figure 4: DFT of 100 samples of a sinusoid with frequency $1/\sqrt{10}$ *is shown*

Note that although the input is a single sinusoid which has only two complex frequencies, there are many non-zero values in the DFT shown in Fig 4. This is called *spectral leakage*: energy at one frequeny spills over into nearby frequencies. Spectral leakage occurs when the input frequency is not one of the frequencies k/N sampled by the DFT. In this case, the frequency is $1/\sqrt{10}$ cycles/second. We will examine the cause of spectral leakage shortly.

The *spectral resolution* of the DFT with N points is 1/N cycles/sample, which means that any two input frequencies that are separated by 1/N cycles/sample can be distinguished, while those that are separated by less cannot be distinguished from one another because of spectral leakage. As an aid to distinguishing closely-spaced frequencies, a technique known as *zero-padding* can be used. This technique simply evaluates the DFT on M > N points by using the formula:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi(k/M)n} \qquad k = 0,1,\cdots M-1 \tag{3}$$

As the name suggests, evaluating the DFT at more frequencies than samples is equivalent to “padding” the input with M-N zeros, and evaluating the M point DFT of the padded signal. Figure 5 shows the DFT of a 32-point signal with two sinusoids with and without zero-padding.

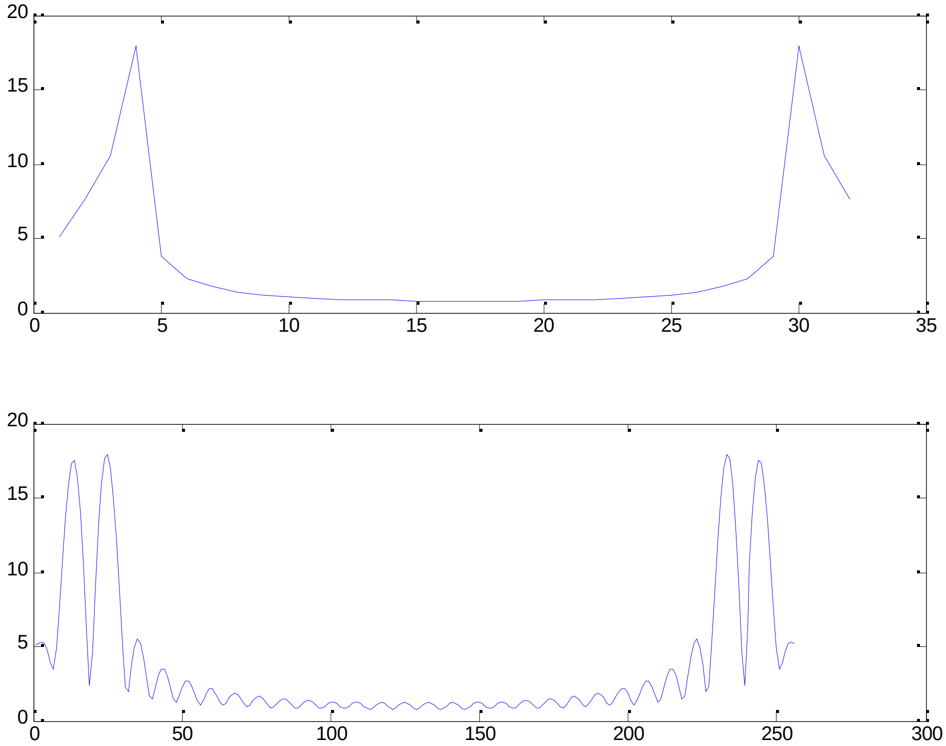


Figure 5: A 32-point signal with two sinusoids, at respective frequencies $\sqrt{3}/32$ *and* $\sqrt{2}/16$ *cycles/sample is processed using a 32-point DFT (top) and a 256-point DFT with zero-padding (bottom).*

It should be emphasized that zero-padding does NOT improve the *resolution* of a DFT, since no new data is introduced. Zero-padding only interpolates between existing DFT values to make for a smoother graph.

The reasons for spectral leakage and for the resolution limit become clearer if we consider the limiting case of zero-padding, by letting $M \to \infty$. Then, the DFT becomes the Discrete-time Fourier Transform (DTFT)

$$X(F) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi Fn} \qquad 0 \le F < 1$$

Note that although the frequency F is now a continuous variable, resolution is still limited by the number of data points N. Note also that the DFT is a sampled DTFT.

It can be shown if a signal has P true frequencies $F_1, F_2, \cdots, F_P$, with corresponding complex amplitudes $X_1, \cdots, X_P$, then the DTFT obtained with a window on N samples is

$$X(F) = \sum_{k=1}^{P} X_k W_N(F - F_k)$$

where the window function's DTFT is

$$W_N(F) = \sum_{n=0}^{N-1} w[n] e^{-j2\pi F n} \qquad 0 \le F < 1\,.$$

For the rectangular window, it can be shown that

$$W_N(F) = \sum_{n=0}^{N-1} e^{-j2\pi F n} = e^{-j\pi F(N-1)} \frac{\sin(\pi F N)}{\sin(\pi F)}$$

The right hand side is known as the Dirichlet function. Note that $W_N(0) = N$ by using L'Hopital's rule, and that $W_N(F)=0$ when $F=1/N, 2/N, \ldots N-1/N$.

Figure 6 illustrates the resolution of two sinusoids with closely-spaced frequencies when N=32 samples are obtained. The signal is

$$x[n] = \cos(2\pi F_1 n) + \cos(2\pi F_2 n), \qquad n = 0,1,\ldots 31$$

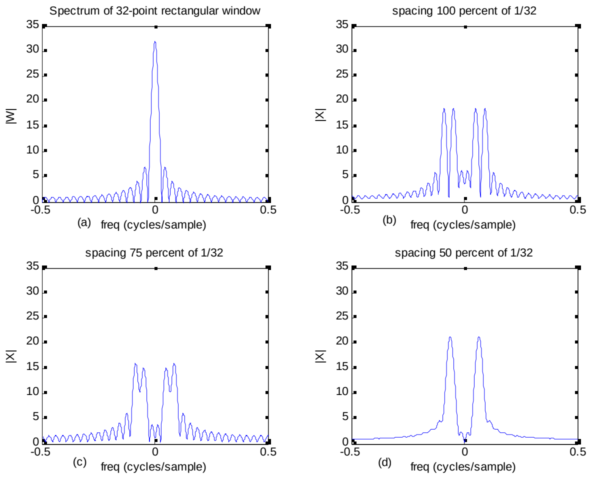


Figure 6:Illustration of frequency resolution with 32 samples of a sum of two equal-amplitude sinuosids. A 1024 point DFT was computed using zero-padding. (a) Spectrum of rectangular window (Dirichlet function) for N=32; (b) Two sinusoids with frequencies F1 and F2 spaced 1/32 apart; (c) with frequencies spaced 75% of 1/32 apart (note amplitudes do not appear the same); (d) with frequencies 50% of 1/32 apart (unresolvable as two sinusoids).

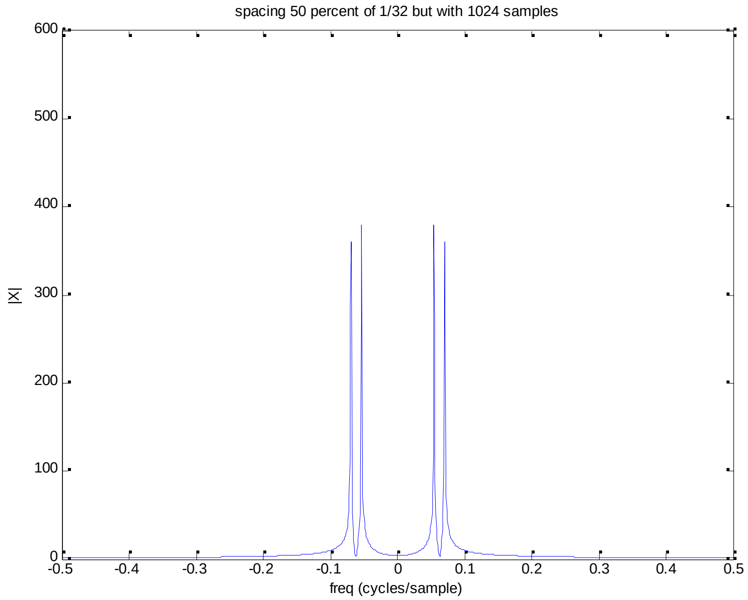


Figure 7: Same signal as in Figure 6(d), but with N=1024 samples: note the clear separation!

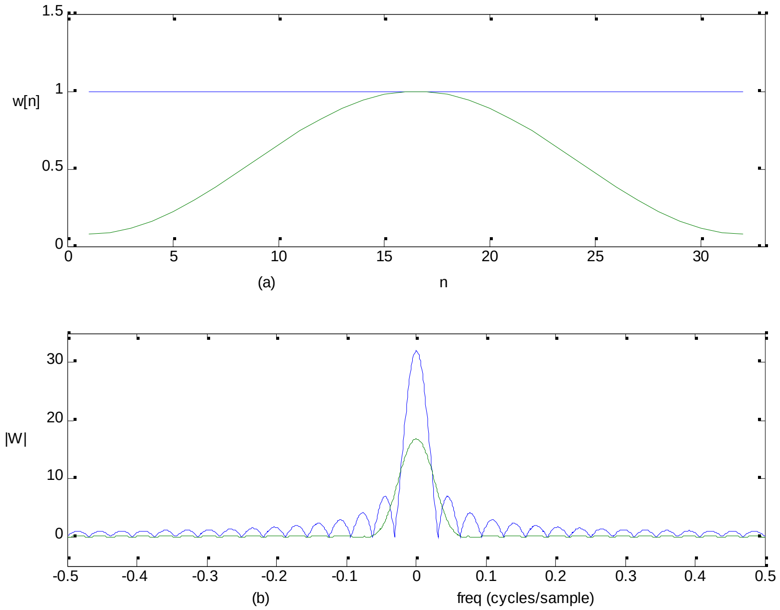


Figure 8: (a) The Hamming window for N=32 is shown, along with the rectangular "window" (dashed); (b) The spectra of the two windows functions on the same axes. Note that the Hamming window is shorter and broader (twice as wide main lobe), but has much less oscillation away from the main lobe.

Figure 7 shows for comparison the spectra of the two sinusoids in Figure 6 with N=1024 signal samples. As always, it is obvious that more data is better! Figure 8 shows the Hamming window in comparison to the rectangular window in both the time and spectral domains. Note that the main lobe of the Hamming window is twice as wide as the rectangular window, but aside from the main lobe it has much smaller amplitude.

The FFT

The Fast Fourier Transform (FFT), whose basic idea can be attributed to Gauss, is a fast way to evaluate the DFT. An examination of a N-point DFT shows that it requires N multiplies and N-1 adds for each frequency point, and since there are N frequency points, this is $N(2N-1)$ operations. This reasoning also shows that a N/2-point DFT requires less than half the number of operations. For example, a 100-point DFT requires 100*(199) operations, while a 50-point DFT requires 50*(99) operations, which is roughly 25% of the 100-point DFT. Hence, even evaluating two 50-point DFTs would still require *less* computation than a single 100-point DFT.

The idea behind the FFT is to break up the N-point DFT into two N/2 point DFTs, the first over the even numbered samples and the second over the odd numbers.

$$\begin{aligned}\sum_{n=0}^{N-1} x[n]e^{-j2\pi(k/N)n} &= \sum_{n=0}^{N/2-1} x[2n]e^{-j2\pi(k/N)2n} + \sum_{n=0}^{N/2-1} x[2n+1]e^{-j2\pi(k/N)(2n+1)} \\ &= \sum_{n=0}^{N/2-1} x[2n]e^{-j2\pi(k/N)2n} + e^{-j2\pi(k/N)} \sum_{n=0}^{N/2-1} x[2n+1]e^{-j2\pi(k/N)(2n)}\end{aligned}$$

Combining the two N/2 point DFTs takes only 1 multiply and 1 add per frequency, which puts the total number of operations at

$$2\times\frac{N}{2}(N-1)+2N = N^2+N$$

But this still less than the $2N^2-N$ operations required for the N point DFT.

The idea of breaking up the N-point DFT into two N/2-point DFTS can be continued: each of the N/2 point DFTs can be further broken up into two N/4-point DFTs. The scheme works best when N is a power of 2, in which the result can be reduced to finally to 2-point DFTs. Hence the FFT is most effective when applied to data which has a length that is a power of 2.

The DFT is considered an $O(N^2)$ algorithm while the FFT is considered an $O(N\log_2 N)$ algorithm because it relies on recursively dividing the N point DFT into half-size DFTs.

Additional topics

1. The results of a DFT (or FFT) can be reordered for display so that DC is in the middle, negative frequencies are on the left and positive on the right. This is done using the "fftshift" command in MATLAB.
2. The DFT can be used for creating an "arbitrary" filter by specifying a filter response $H[k]$ which multiplies the input signal DFT $X[k]$. However, since the convolution of a N-point input with a L point impulse response has N+L-1 points, the DFT of both the filter and the input must be evaluated on N+L-1 points using zero-padding prior to multiplication. The output of filtering is then obtained by using an inverse DFT as follows

   $$y[n] = DFT_{N+L-1}^{-1}\left\{DFT_{N+L-1}\left\{H[k]\right\} * DFT_{N+L-1}\left\{X[k]\right\}\right\} \qquad (3)$$

   Note that this formula seem to work even if $L = 1$. In that case, eq. (3) computes a *circular convolution*, in which the impulse response obtained by $h[n] = DFT_N^{-1}\left\{H[k]\right\}$ is circularly wrapped around due to the periodicity of the inverse DFT:

   $$\begin{aligned} h[n+N] &= \frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi(k/N)(n+N)} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi(k/N)(n)} \\ &= h[n] \end{aligned}$$

   Circular convolution creates artifacts due to this wrap around, which is why it should be used with care or avoided when possible.
3. The DFT can be used to convolve a long sequence with an impulse response

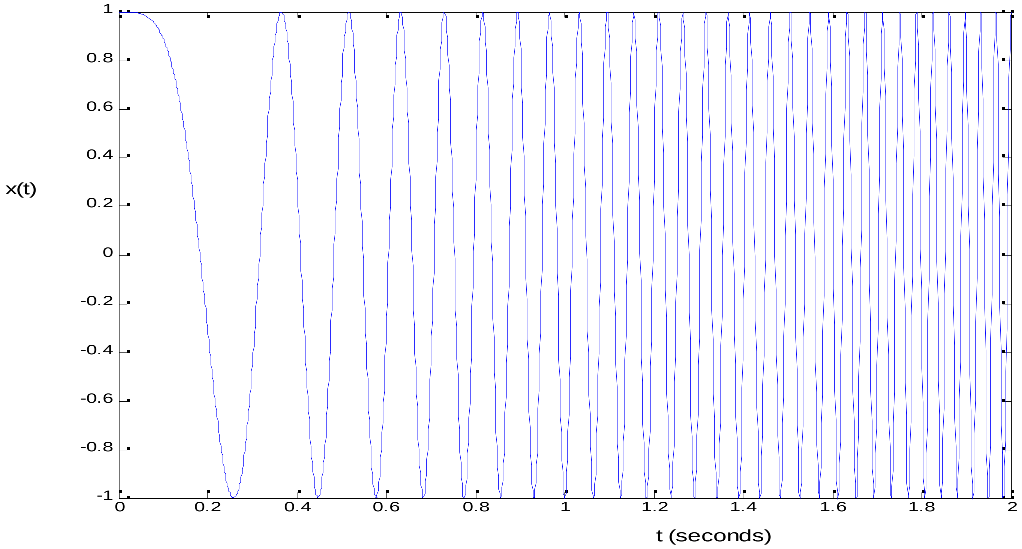
   dividing it into blocks of length L (the length of the impulse response) and computing using eq. (3) with N=L. The resulting 2L-1 point inverse DFT must be added to the 2L-1 point inverse DFT of the next block with an overlap of L-1 points.
4. Zero-padding in MATLAB is accomplished with a second argument to the "fft". Hence "fft(x)" is the normal DFT, and "fft(x,M)" computes a M-point DFT, padding with zeros if M>length(x).

```
function showspectrogram(y,fs)
   % usage
   %        showspectrogram(y,fs)
   %
   % shows the spectrogram and the time domain plot of the
   % signal y.  Default sampling rate is fs=8192.
   %
   % R Kakarala, PhD
   % U C Berkeley Extension
   % last rev: 24 feb 2003
   if nargin<2
       fs = 8192;
   end;
   % compute the spectrogram
   % assume a 256 point Hamming window, with 128 sample overlap
   % applied to the data.
   Ly = length(y);
   L = 256;
   Nfft = 256;
   Noverlap = 256;  % this, the max overlap, is adjusted below
   num_segs = 1025;
   while num_segs > 1024
       Noverlap = Noverlap-10;
       shift = abs(L-Noverlap);
       num_segs = 1 + fix( (Ly-L)/shift );
   end;

   window = 0.54-0.46*cos(2*pi*(0:L-1)/(L-1));  % Hamming window
   window = window(:); y = y(:);
   % make both column vectors to multiply element by element
   %  Next, divides the input into segments,
   % each of length L and takes the windowed fft of each one
   B = zeros( Nfft/2+1, num_segs ); %- Pre-allocate the matrix
   iseg = 0;
   while( iseg<num_segs )
       nstart = 1 + iseg*shift;
       ysegw = window .* y( nstart:nstart+L-1);
       YF = fft( ysegw, Nfft );
       iseg = iseg + 1;
       B(:,iseg) = YF(1:Nfft/2+1);
   end
   % use imagesc -- image scaled to show the histogram in dB
   F = (0:(Nfft/2))/Nfft * fs;
   T = ( L/2 + shift*(0:num_segs-1) ) / fs;
   subplot(2,1,1);
   imagesc(T,F,20*log10(abs(B)+eps)); axis xy; colormap(jet);
   ylabel('frequency (Hz)'); xlabel('time (sec)'); title
   ('Spectrogram (dB)');
   % plot time signal, trying to align time axes for both plots
   maxsampleused = nstart+L-2;
   subplot(2,1,2);
   plot((0:maxsampleused)/fs,y(1:maxsampleused+1));
   axis([min(T),max(T),min(y),max(y)]);
   xlabel('time (sec)'); ylabel('signal');
```

Homework

1. Sketch the spectrogram of the “chirp” signal shown below. A rough sketch is fine, but try to guess as many numerical values for frequency as possible.



2. Prove that

$$\sum_{n=0}^{N-1} e^{-j2\pi Fn} = e^{-j\pi F(N-1)} \frac{\sin(\pi FN)}{\sin(\pi F)}$$

The right hand side is known as the Dirichlet function. Roughly sketch its absolute value as a function of F, for $-2 \le F \le 2$. Try to indicate where the zeros are.

3. Sketch a signal whose spectrogram would look as below. A rough sketch is fine, and you can break the sketch up into parts if it helps to make your point.