BGHiggins/UCDavis_March_2008

Primer on Fourier Transforms

Introduction

We start our discussion with the definition of the Fourier Transform of a function h[t]:

$$H[\omega] = \int_{-\infty}^{\infty} h[t]\, e^{i\,\omega\,t}\, dt \tag{1}$$

Then the inverse transform of H[ω] is given by

$$h[t] = \frac{1}{2\,\pi} \int_{-\infty}^{\infty} H[\omega]\, e^{-i\,\omega\,t}\, d\omega \tag{2}$$

This definition for the Fourier Transform pair is not universal, however, and various definitions in the literature can often lead to confusion. A more general definition for the Fourier Transform of f[t] is

$$H[\omega] = \sqrt{\frac{|b|}{(2\,\pi)^{1-a}}} \int_{-\infty}^{\infty} h[t]\, e^{i\,b\,\omega\,t}\, dt \tag{3}$$

where the parameters a, b can take on various values. The values of the parameters in the expression given by Eq. (1) are $b = 1$, and $a = 1$. In *Mathematica* these options can be set in the function **FourierTransform**. The default values are $a = 0,\ b = 1$. Different disciplines tend to have their favorite definitions. For example, in signal processing and electrical engineering the parameters of choice are: $\{a, b\} \to \{0, -2\,\pi\}$ or $\{-1, -2\,\pi\}$.

Consider the waveform β Cos[α t]. Here α represent the angular frequency which is equal to the frequency f (s^{-1}) by the usual formula:

$$f = \frac{\alpha}{2\,\pi} \tag{4}$$

The period T is related to the frequency f by

$$T = \frac{1}{f} = \frac{2\,\pi}{\alpha} \tag{5}$$

The Fourier Transform of the cosine wave form β Cos[α t] using the default values for the parameters {a,b} gives

```
FourierTransform[β Cos[α t], t, ω]
```

$$\sqrt{\frac{\pi}{2}}\ \beta\, \text{DiracDelta}[-\alpha + \omega] + \sqrt{\frac{\pi}{2}}\ \beta\, \text{DiracDelta}[\alpha + \omega]$$

Thus in the frequency domain, the Fourier Transform of a cosine function results in *two* real-valued Dirac delta functions, centered at $\omega = \pm\alpha$ with magnitude $\beta\sqrt{\pi / 2}$. Note that the magnitude of the Dirac delta function is proportional to the original amplitude of the wave form. The actual value depends on the parameters {a,b} us in the transform, see Eq. (3) above. Consider next the Fourier Transform of the waveform function

$$h[t] = \text{Cos}[2\pi f_0 t] - \frac{1}{3}\text{Cos}[6\pi f_0 t] + \frac{1}{5}\text{Cos}[10\pi f_0 t] \qquad (6)$$

In this case the waveform is a linear superposition of 3 cosine function with different frequencies and amplitudes. The resulting transform is a sum of 3 Dirac Delta functions centered at the angular frequencies

$$\omega = \pm 2\pi f_0,\ \ \omega = \pm 6\pi f_0,\ \ \omega = \pm 10\pi f_0 \qquad (7)$$

with magnitudes proportional to the original amplitudes of the cosine functions in the wave form. Here is the *Mathematica* rendition of the Fourier Transform

```
h[t_] := Cos[2 π f0 t] - 1/3 Cos[6 π f0 t] + 1/5 Cos[10 π f0 t];
FourierTransform[h[t], t, ω]
```

$$\frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 10\pi f_0] - \frac{1}{3}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 6\pi f_0] + \sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 2\pi f_0] + \sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 2\pi f_0] - \frac{1}{3}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 6\pi f_0] + \frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 10\pi f_0]$$

Now let us replace the Cos[$6\pi f_0 t$] in $h(t)$ with Sin[$6\pi f_0 t$]

```
h2[t_] := Cos[2 π f0 t] - 1/3 Sin[6 π f0 t] + 1/5 Cos[10 π f0 t];
FourierTransform[h2[t], t, ω]
```

$$\frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 10\pi f_0] - \frac{1}{3}\, i\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 6\pi f_0] + \sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega - 2\pi f_0] + \sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 2\pi f_0] + \frac{1}{3}\, i\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 6\pi f_0] + \frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\omega + 10\pi f_0]$$

Note that the magnitude in the frequency domain for that mode is an imaginary number. Thus in general the Fourier Transform is a complex quantity which can be represented as

$$H(f) = R(f) + i\, I(f) = |H(f)|\, e^{i\,\theta(f)}$$

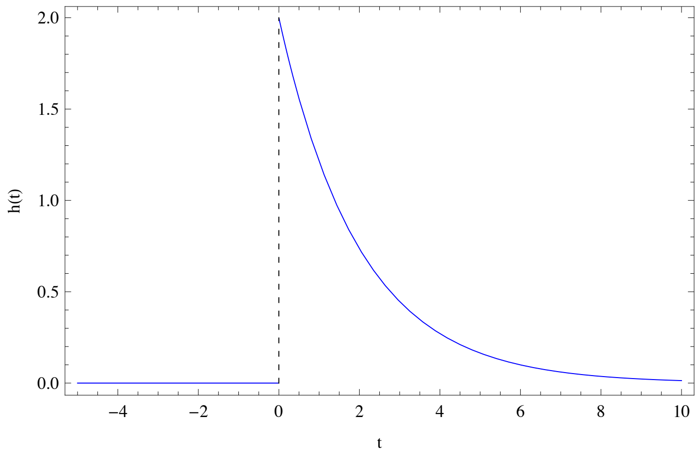
In this expression $R(f)$ is the real part of the Fourier Transform, $I(f)$ is the imaginary part, $|H(f)|$ is the amplitude or Fourier spectrum of $h(t)$ and is given by $\sqrt{R^2 + I^2}$. The quantity $\theta(f)$ is the phase angle of the Fourier transform and is given by $\text{Tan}^{-1}[I(f)/R(f)]$

As our final example we consider the function

```
h3[t_] := UnitStep[t] β e^(-α t)
```

Here is a plot of this function for a select set of parameters $\{\beta \to 2,\ \alpha \to 0.5\}$

```
Plot[Evaluate[h3[t] /.{β → 2.`, α → 0.5`}], {t, -5, 10},
 PlotRange → All, PlotStyle → RGBColor[0, 0, 1], Frame → True, Axes → False,
 FrameLabel → {"t", "h(t)"}, Epilog → {Dashing[Small], Line[{{0, 0}, {0, 2}}]}]
```



The Fourier Transform of this function is

```
H3[ω_] = FourierTransform[h3[t], t, ω]
```

$$\frac{\beta}{\sqrt{2\pi}\,(\alpha - i\,\omega)}$$

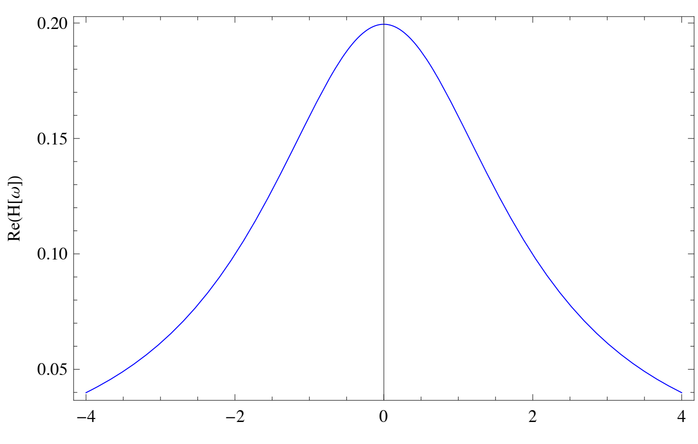
This clearly show that the Fourier Transform of `h3[t]` is a complex variable. We can interrogate its properties as follows. Here is the real part of `h[ω]`

```
realH = ComplexExpand[Re[H3[ω]], TargetFunctions → {Re, Im}] // PowerExpand
```

$$\frac{\alpha\,\beta}{\sqrt{2\pi}\,(\alpha^2 + \omega^2)}$$

An here is a plot of `Re (H[ω])` as a function of ω

```
Plot[Evaluate[realH /.{α → 2, β → 1}], {ω, -4, 4},
 PlotStyle → RGBColor[0, 0, 1], Frame → True, FrameLabel → {"ω", "Re(H[ω])"}]
```



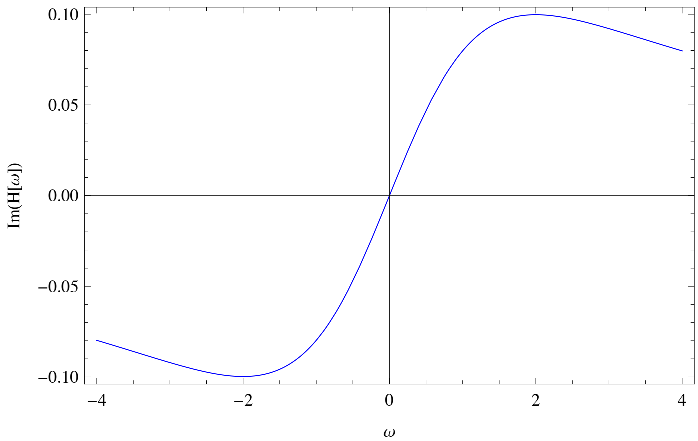
Next, let us consider the imaginary component of `H[ω]`

```
imagH = ComplexExpand[Im[H3[ω]], TargetFunctions → {Re, Im}] // PowerExpand
```

$$\frac{\beta\,\omega}{\sqrt{2\,\pi}\,\left(\alpha^2+\omega^2\right)}$$

Shown below is a plot of `Im (H[ω])`

```
Plot[Evaluate[imagH /.{α → 2, β → 1}], {ω, -4, 4},
 PlotStyle → RGBColor[0, 0, 1], Frame → True, FrameLabel → {"ω", "Im(H[ω])"}]
```



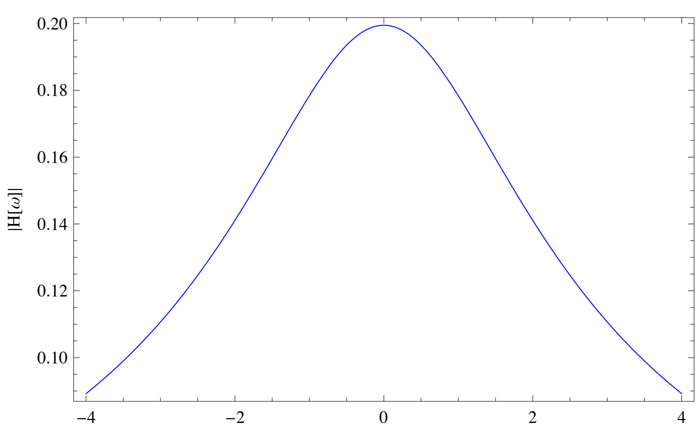
Final here is the amplitude |H(f)|

```
absH = ComplexExpand[Abs[H3[ω]], TargetFunctions → {Re, Im}] // PowerExpand
```

$$\frac{\beta}{\sqrt{2\,\pi}\,\sqrt{\alpha^2+\omega^2}}$$

and its plot

```
Plot[Evaluate[absH /.{α → 2, β → 1}], {ω, -4, 4}, PlotStyle → RGBColor[0, 0, 1],
 Frame → True, FrameLabel → {"ω", "|H[ω]|"}, Axes → False]
```



In summary, the Fourier Transform of a real function results in *a complex variable*. Further, the frequency domain of the transform function includes negative frequencies.

Discrete Fourier Transform

In this section we consider the discrete version of the Fourier Transform. We start our discussion by considering the following

$$f[t] = \text{Cos}[\alpha\, t] \tag{8}$$

As before α represent the angular frequency which is equal to the frequency f $\left(s^{-1}\right)$ by the usual formula:

$$f = \frac{\alpha}{2\,\pi} \tag{9}$$

and the period T is related to the frequency f by

$$T = \frac{1}{f} = \frac{2\,\pi}{\alpha} \tag{10}$$

We showed in the previous section that the Fourier Transform of this function gives

```
FourierTransform[Cos[α t], t, ω]
```

$$\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[-\alpha+\omega] + \sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[\alpha+\omega]$$

Thus in the frequency domain we have spikes at $\omega = \pm\alpha$. Thus in the frequency domain ω represents the *angular frequency*. On the other hand if we use different FourierTransform parameters, the interpretation is different. With $\{a,\ b\} \to \{0,\ -2\,\pi\}$, the Fourier Transform results in spikes at $\omega = \pm\ \alpha\,/\,2\,\pi$. In this case the Dirac delta spikes define *the frequency* of the function! Here is the result using *Mathematica*

```
FourierTransform[Cos[α t], t, ω, FourierParameters → {0, -2 π}]
```

$$\pi\,\text{DiracDelta}[\alpha - 2\,\pi\,\omega] + \pi\,\text{DiracDelta}[\alpha + 2\,\pi\,\omega]$$

Thus the Fourier Transform can be in expressed terms of *frequency* or *angular frequency*. What interpretation we used is based on the FourierTransform parameters we select.

In the next several examples we illustrate properties of the discrete Fourier Transform.

Sampling a function

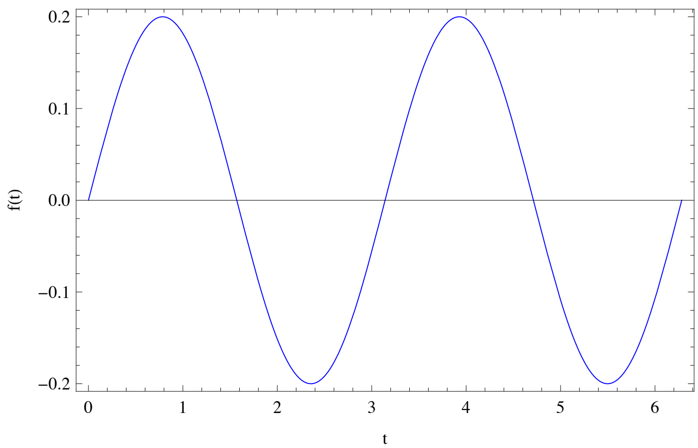
In this section we are going to sample the function $\frac{1}{5}\,\text{Sin}(2\,t)$, and then in a later section compute its discrete Fourier transform. We consider the following *Mathematica* function

```
f[t_] = 1 / 5 Sin[2 t]
```

$$\frac{1}{5}\,\text{Sin}[2\,t]$$

Here is a plot of this function for x in the range $0 \le t \le 4$

```
Plot[f[t], {t, 0, 2 π}, PlotStyle → RGBColor[0, 0, 1],
 Frame → True, FrameLabel → {"t", "f(t)"}]
```



We are going to sample this function at intervals Δt_s to create a set of N equally spaced points. Since the range of t for our function is $t = T = 2\pi$, then the time step for our sampling is

$$\Delta t_s = \frac{T}{N-1} \tag{11}$$

Thus if t_j represents the value of t after j time steps we have

$$t_j = j\,\Delta t_s,\quad j = 0,\ 1,\ 2,\ \ldots,\ N-1 \tag{12}$$

where

$$t_{N-1} = (N-1)\,\Delta t_s = T \tag{13}$$

Using these definitions, it follows that the value of our function evaluated at these time steps is

$$f_j = f[t_j],\quad j = 0,\ 1,\ \ldots,\ N-1 \tag{14}$$

Now if us apply this discretization to our function. First we define the number of points Npts and the range T_{max}

```
Npts = 32;
Tmax = 2 π;
```

From these to quantities we can compute the spacing between the sample points

```
Δt = (Tmax / (Npts - 1) // N)
```

```
0.202683
```

Using these values, we obtain a discrete version of our function given by the following list of data with length 32:

```
fdata = Table[f[t] , {t, 0, Tmax, Δt}]
```

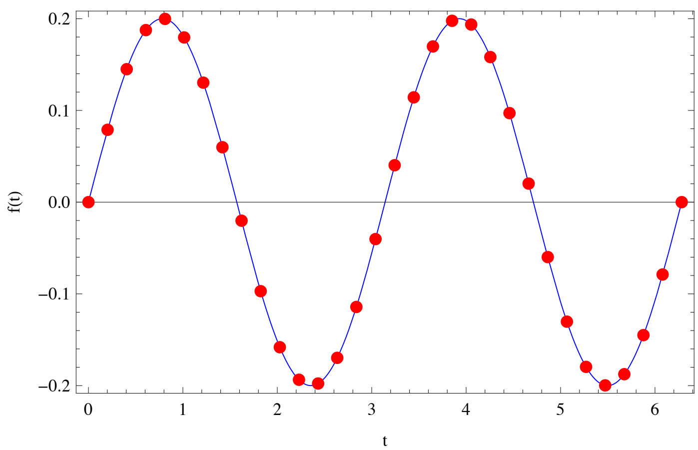
```
{0., 0.0788712, 0.144959, 0.18755, 0.199743, 0.179561, 0.130274,
 0.0598726, -0.0202337, -0.0970604, -0.158155, -0.193615, -0.197694,
 -0.169729, -0.114254, -0.0402597, 0.0402597, 0.114254, 0.169729,
 0.197694, 0.193615, 0.158155, 0.0970604, 0.0202337, -0.0598726, -0.130274,
 -0.179561, -0.199743, -0.18755, -0.144959, -0.0788712, -9.79717 × 10^-17}
```

Let us create a point object for our sample points to display on the graph

```
dataPts = Join[{Red, PointSize[0.02]}, Map[Point[{#, f[#]}] &, Range[0, Tmax, Δt]]];
```

Here is a plot that shows how we sampled our function

```
Plot[f[t], {t, 0, Tmax}, PlotStyle → RGBColor[0, 0, 1],
 Epilog → dataPts, Frame → True, FrameLabel → {"t", "f(t)"}]
```



In summary we have N=32 sample points and N-1 intervals of length Δt=0.202683. Note that $N\cdot\Delta t \neq 2\pi$. This is going to be important later.

Discrete Fourier Transform Pair

We begin our discussion by noting the following orthogonality condition

$$\sum_{k=0}^{N-1} e^{i\,2\,\pi\,r\,k/N}\, e^{-i\,2\,\pi\,n\,k/N} = \begin{pmatrix} N & \text{if } r = n \\ 0 & \text{if } r \neq n \end{pmatrix} \tag{15}$$

We define the discrete Fourier transform of $f(t)$ as

$$F\left(\frac{n}{N\,\Delta t}\right) = \sum_{k=0}^{N-1} f\,(k\,\Delta t)\; e^{-i\,2\,\pi\,n\,k/N}, \quad n = 0,\ 1,\ \ldots,\ N-1 \tag{16}$$

where $t_k = k\,\Delta t$ is the k^{th} sample point, and $f_n = n/(N\,\Delta t)$ is the n^{th} frequency. The inverse transform is then

$$f\,(t_k) = \frac{1}{N} \sum_{n=0}^{N-1} F\,(f_n)\; e^{i\,2\,\pi\,n\,k/N}, \quad k = 0,\ 1,\ \ldots,\ N-1 \tag{17}$$

We can use the orthogonality property to prove that (16) and (17) are Fourier Transform pairs by direct substitution of (17) into (16):

$$\begin{aligned} F\,(f_n) &= \sum_{k=0}^{N-1} \left(\frac{1}{N} \sum_{r=0}^{N-1} F\,(f_r)\; e^{i\,2\,\pi\,r\,k/N} \right) e^{-i\,2\,\pi\,n\,k/N} \\ &= \frac{1}{N} \sum_{r=0}^{N-1} F\,(f_r) \sum_{k=0}^{N-1} e^{i\,2\,\pi\,r\,k/N}\, e^{-2\,\pi\,n\,k/N} \end{aligned} \tag{18}$$

Applying the orthogonality condition gives

$$F\ (f_n) = \frac{1}{N}\ F\ (f_n)\ N = F\ (f_n) \qquad (19)$$

In *Mathematica* we work with lists, and thus it is convenient to transform the summation terms (17) and (18) to start from $n = 1,\ k = 1$, rather than $n = 0,\ k = 0$. It will be convenient as well to rescale our transform pair so that the definitions are consistent with those used by *Mathematica* with the default FourierParameters. Our definitions for the Discrete Fourier Transform (DFT) pair become

$$F\ (f_n) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} f\ (t_k)\ e^{i\,2\,\pi\,(n-1)\ (k-1)/N},\ n = 1,\ 2,\ \ldots,\ N \qquad (20)$$

$$f\ (t_k) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F\ (f_n)\ e^{-i\,2\,\pi\,(n-1)\ (k-1)/N},\ k = 1,\ 2,\ \ldots,\ N \qquad (21)$$

Thus the Discrete Fourier Transform (DFT) pair relates N points in the frequency domain with N points in the time domain.

Example 1: Discrete Fourier Transform

In a previous section we sampled the function $\frac{1}{5}$ Sin[2 t] over the interval $0 \le t \le 2\pi$, with N=32 sample points and $\Delta t = 0.202683$. The frequency of our Sine function is $f = 2/(2\pi) = 0.31831$ We will use Eq. (21) to compute the DFT of our data stored in the variable **fdata**

$$\mathbf{Fdata[n\_] := \frac{1}{\sqrt{Npts}} \sum_{k=1}^{Npts} fdata[[k]]\ e^{i\,2\,\pi\,(n-1)\ (k-1)/Npts}}$$

This gives the Fourier components with amplitudes Fdata[n] and frequencies $f_n = (n-1)/(\text{Npts}\ \Delta t)$, with $n = 1, 2, \ldots,$ Npts.

Here are the amplitudes of our data in the frequency domain. Note that the amplitudes involve complex values

```
FdataList = Table[Fdata[n], {n, 1, Npts}]
```

```
{-1.44889 × 10^-16, 0.00216654 + 0.0219972 i, 0.10782 + 0.542046 i,
 -0.0134289 - 0.0442693 i, -0.00963811 - 0.0232685 i, -0.00852602 - 0.0159511 i,
 -0.0080248 - 0.01201 i, -0.00775178 - 0.00944557 i, -0.00758609 - 0.00758609 i,
 -0.00747842 - 0.00613738 i, -0.00740534 - 0.00494809 i, -0.00735444 - 0.00393103 i,
 -0.00731874 - 0.00303152 i, -0.00729405 - 0.00221263 i, -0.00727787 - 0.00144766 i,
 -0.00726868 - 0.000715903 i, -0.00726571, -0.00726868 + 0.000715903 i,
 -0.00727787 + 0.00144766 i, -0.00729405 + 0.00221263 i,
 -0.00731874 + 0.00303152 i, -0.00735444 + 0.00393103 i, -0.00740534 + 0.00494809 i,
 -0.00747842 + 0.00613738 i, -0.00758609 + 0.00758609 i, -0.00775178 + 0.00944557 i,
 -0.0080248 + 0.01201 i, -0.00852602 + 0.0159511 i, -0.00963811 + 0.0232685 i,
 -0.0134289 + 0.0442693 i, 0.10782 - 0.542046 i, 0.00216654 - 0.0219972 i}
```

We can compare these values with those returned by the *Mathematica* function Fourier:

```
Chop[FdataList - Fourier[fdata]]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

where we have used the function Chop to replace terms that are smaller in absolute magnitude than 10^{-10} with the integer 0. Thus the Fourier components using the two methods are identical within an error of 10^{-10} .

Although frequency distributions are generally complex functions, the relative strength of various frequency components can be assessed from the power spectrum. From Parseval's theorem the power spectrum in the time domain is related to the power spectrum in the frequency domain

$$\sum_{k=0}^{N-1} f(t_k)^2 = \sum_{n=0}^{N-1} |F(f_n)|^2 \tag{22}$$

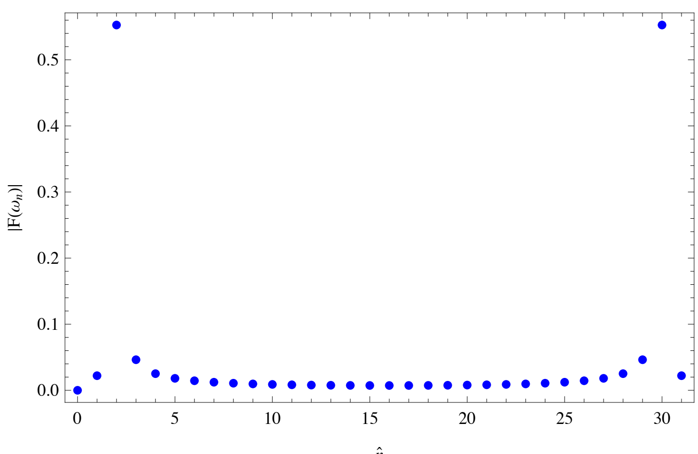
We use the *Mathematica* function Fourier to obtain a discrete transform

```
FData = Abs[Fourier[fdata]]
```

```
{9.81308 × 10^-17, 0.0221036, 0.552665, 0.0462613, 0.0251856, 0.0180867, 0.0144443,
 0.0122192, 0.0107283, 0.00967441, 0.00890632, 0.00833912, 0.00792174,
 0.00762227, 0.00742045, 0.00730385, 0.00726571, 0.00730385, 0.00742045,
 0.00762227, 0.00792174, 0.00833912, 0.00890632, 0.00967441, 0.0107283,
 0.0122192, 0.0144443, 0.0180867, 0.0251856, 0.0462613, 0.552665, 0.0221036}
```

If we inspect the above data, we observe that the maximum amplitudes occur at position n =3 and n =31, where n denotes the position in the data list In the following plot we show the raw data of the spectrum plotted against the value of the index $\hat{n} = n - 1$ in Eq. (22). For plotting purposes we create a list of values {0, Npts – 1}. Recall from the definition of the transform, values of $n >$ Npts/2 represent negative frequencies

```
xcoor = Table[k, {k, 0, Npts - 1}];
FPlot = ListPlot[Transpose[{xcoor, FData}],
  Joined → False, AxesOrigin → {0, 0}, PlotRange → All, Frame → True,
  PlotStyle → {Blue, PointSize[Medium]}, FrameLabel → {"n̂", "|F(ωn)|"}]
```



Let us connect the index $\hat{n}$ with actual frequencies. The discrete frequencies f_n for the horizontal-coordinate axis (abscissa) are related to the index n by

$$f_n = \frac{n}{\text{Npts}\,\Delta t}, \quad n = 0, 1, 2, \ldots, \text{Npts} - 1 \tag{23}$$

Here are the corresponding frequencies

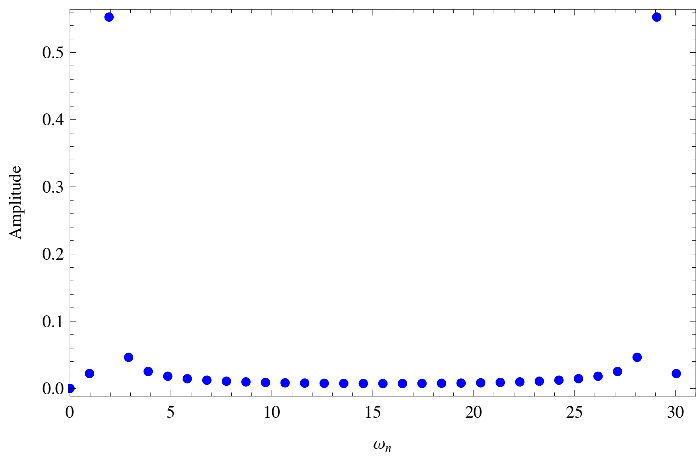
```
Npts = 32;
Tmax = 2 π;
Δt = (Tmax / (Npts - 1) // N);
freq = Table[n / (Npts Δt), {n, 0, Npts - 1}]
```

```
{0, 0.154181, 0.308363, 0.462544, 0.616725, 0.770907, 0.925088, 1.07927,
 1.23345, 1.38763, 1.54181, 1.69599, 1.85018, 2.00436, 2.15854, 2.31272,
 2.4669, 2.62108, 2.77526, 2.92945, 3.08363, 3.23781, 3.39199, 3.54617,
 3.70035, 3.85453, 4.00872, 4.1629, 4.31708, 4.47126, 4.62544, 4.77962}
```

Thus the maximum amplitude occurs at a frequency $f_2 = 2/(32\ 0.202683)$=0.30836. The actual frequency of the input single was $f = 0.31831$.

Here is the same plot except now we use the angular frequency ω_n=$2\pi\ f_n$ as the abscissa

```
ListPlot[Transpose[{2 π freq, FData}], Joined → False, AxesOrigin → {0, 0},
 PlotRange → {{0, 31}, All}, PlotStyle → {RGBColor[0, 0, 1], PointSize[0.015]},
 Frame → True, FrameLabel → {"ωn", "Amplitude"}]
```



Note that the peak amplitude occurs at an angular frequency $\omega_2 \approx 2$. Values of $\omega_n > \omega_{16}$ represent negative angular frequencies. That is, for the 32 data points, the amplitude plot is folded about Npts/2=16.

Recall that the continuous Fourier transform of a function f(t) is symmetric about $\omega = 0$ frequency. Here is the continuous amplitude spectrum for our function $\frac{1}{5}$ Sin[2 t]:

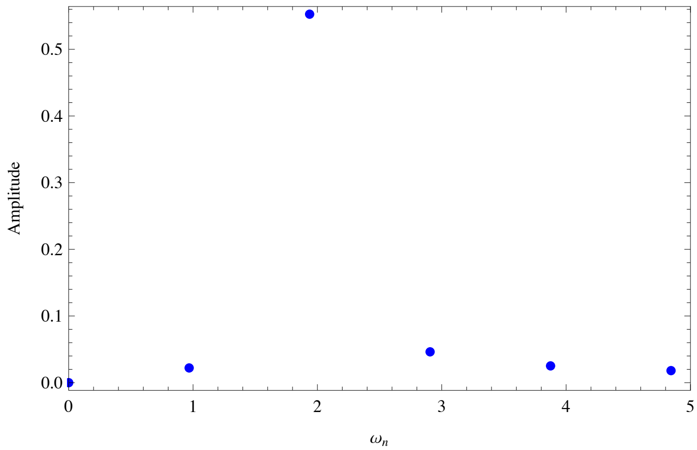
```
H4[ω_] = FourierTransform[1/5 Sin[2 t], t, ω]
absH = ComplexExpand[Abs[H4[ω]], TargetFunctions → {Re, Im}] // PowerExpand
```

$$\frac{1}{5}\,i\,\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[-2+\omega] - \frac{1}{5}\,i\,\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[2+\omega]$$

$$\frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[-2+\omega] - \frac{1}{5}\sqrt{\frac{\pi}{2}}\ \text{DiracDelta}[2+\omega]$$

Since the discrete form for the power spectrum *is symmetric about the mid-point of the data list*, we need only consider half of the transformed data to determine the spectrum. In this example the maximum occurs at $\omega_2 \approx 2$. This can be readily seen by simply plotting the data for the first 5 points

```
ListPlot[Transpose[{2 π freq, FData}], Joined → False, AxesOrigin → {0, 0},
 PlotRange → {{0, 5}, All}, PlotStyle → {RGBColor[0, 0, 1], PointSize[0.015]},
 Frame → True, FrameLabel → {"ωn", "Amplitude"}]
```



A careful inspection of this plot shows that the discrete transform has not reproduced the Fourier Transform of our function, which is a Dirac delta function centered at ω=2. The power spectrum has small but finite values at ω_0, ω_1, ω_3, ω_5. Also, the value of $\omega_2 \neq 2$!

We can improve matters by being careful how we sample the function.

Example 2:

In this example we will drop the last sample point

```
fdata2 = Drop[Table[f[t] , {t, 0, Tmax, Δt}], -1];
```

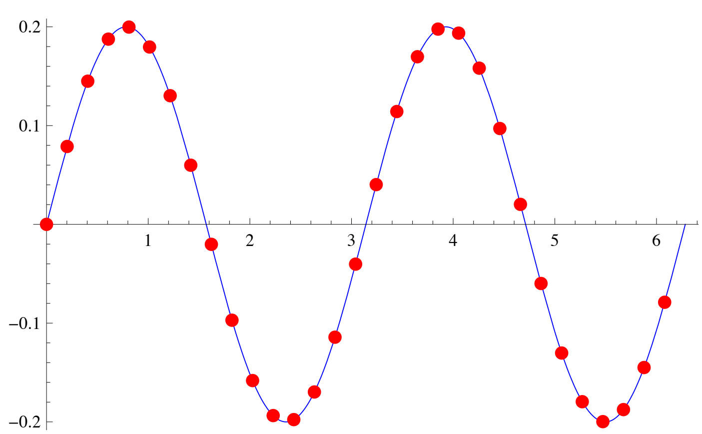
We now have 31 data points

```
Length[fdata2]
```

31

This plot shows how the function was sampled

```
Plot[f[t], {t, 0, Tmax}, PlotStyle → RGBColor[0, 0, 1], Epilog → Drop[dataPts, -1]]
```



As before we compute the spectrum

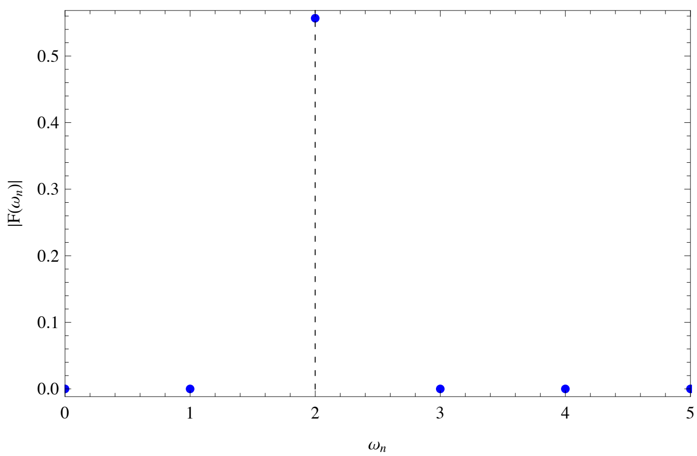
```
FData2 = Abs[Fourier[fdata2]];
```

The discrete frequencies are

```
freq2 = Table[n / ((Npts - 1) Δt), {n, 0, Npts - 2}];
```

An then plot the results for the first 5 points

```
FPlot2 = ListPlot[Transpose[{2 π freq2, FData2}],
  Joined → False, AxesOrigin → {0, 0}, PlotRange → {{0, 5}, All},
  PlotStyle → {RGBColor[0, 0, 1], PointSize[Medium]}, Frame → True,
  FrameLabel → {"ωn", "|F(ωn)|"}, Epilog → {Dashing[Small], Line[{{2, 0}, {2, 0.6}}]}]
```



The discrete transform now exactly reproduces the Fourier Transform. We have a Dirac delta-like function at $\omega_3 = 2$. Examples 1 and 2 show how the sampling strategy can affect the quality of the spectrum. This is called leakage. It occurs when the truncation interval for sampling (N Δt) is not a multiple of the period $T = 2\pi/f$. In our previous example the truncation interval was $N\,\Delta t = (32 \times 0.202683) = 6.485856$, while in this example the truncation interval is $N\,\Delta t = (31 \times 0.202683) = 6.283173 = 2\,T$

Example 3

We will use the same waveform as earlier, viz. $f(t) = \frac{1}{5}\,\mathrm{Sin}(2\,t)$. In this example we choose the truncation interval (N Δt) to be a multiple of the period $T = 1/f = 2\pi/\omega$

```
ω = 2; Npts = 31;
Tmax = 4 (2 π / 2);
Δt = (Tmax / (Npts) // N)
Δt Npts / Tmax
```

```
0.405367
```

```
1.
```

From this data we can calculate the frequency of our signal

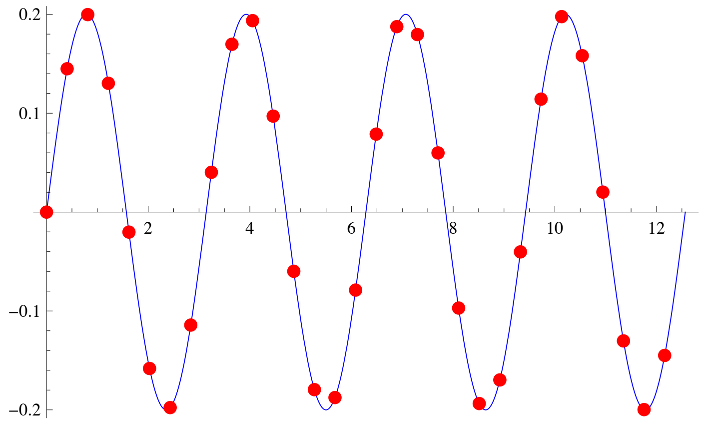
```
freq = N[ω / (2 π)]
```

```
0.31831
```

As before we compute the Npts sample points and display them on the signal

```
dataPts3 = Join[{RGBColor[1, 0, 0], PointSize[0.02]},
   Map[Point[{#, f[#]}] &, Range[0, Tmax - Δt, Δt]]];
```

```
Plot[f[t], {t, 0, Tmax}, PlotStyle → RGBColor[0, 0, 1], Epilog → dataPts3]
```



Next we sample our function

```
fdata3 = Table[f[t] , {t, 0, Tmax - Δt, Δt}];
Length[fdata3]
```

31

Then compute the spectrum

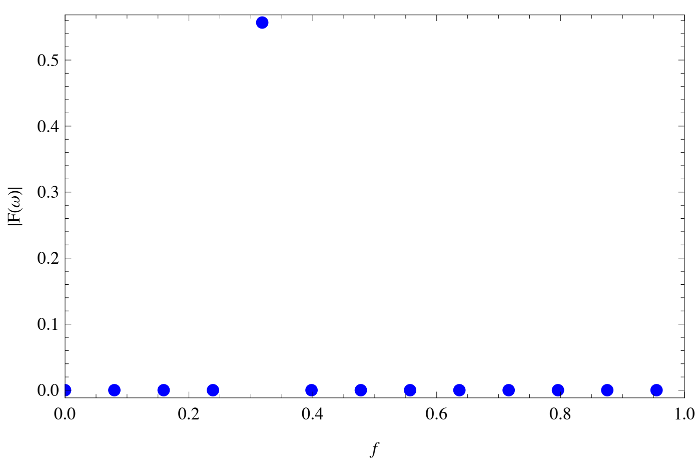
```
FData3 = Abs[Fourier[fdata3]];
```

We calculate the discrete frequencies for the horizontal-coordinate axis from

$$f_k = \frac{k}{Npts \, \Delta t}, \quad k = 0, 1, 2, \ldots, Npts - 1 \tag{24}$$

```
freq = Table[k / (Npts Δt), {k, 0, Npts - 1}];
```

```
FPlot2 = ListPlot[Transpose[{freq, FData3}], Joined → False, AxesOrigin → {0, 0},
  PlotRange → {{0, 1}, All}, PlotStyle → {RGBColor[0, 0, 1], PointSize[0.02`]},
  Frame → True, FrameLabel → {"f", "|F(ω)|"}]
```

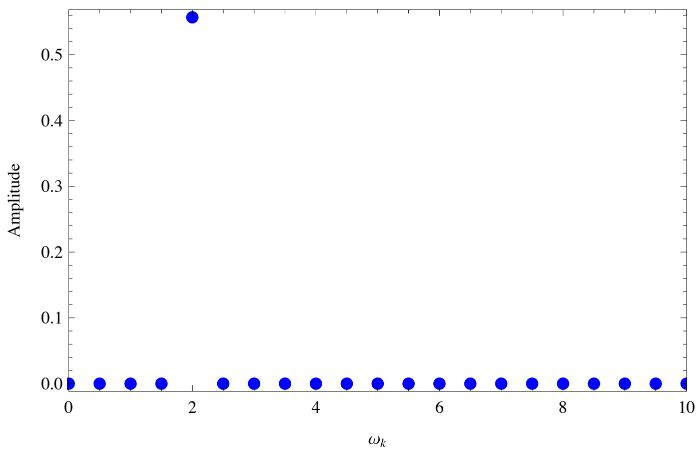


```
freq[[5]]
```

```
0.31831
```

Thus we see that we have a maximum signal at $f_4 \approx 0.31831$, which is in full agreement with actual frequency of f=0.31831.... We can also display the plot in terms of the angular frequency ω

```
FPlot2 = ListPlot[Transpose[{2 π freq, FData3}], Joined → False, AxesOrigin → {0, 0},
  PlotRange → {{0, 10}, All}, PlotStyle → {RGBColor[0, 0, 1], PointSize[0.02`]},
  Frame → True, FrameLabel → {"ωk", "Amplitude"}]
```



As expected the maximum occurs at $\omega_4 = 2$.

Example 4

We will repeat the above calculation in Example 3 but now we use a Hanning Filter to suppress leakage when the truncation interval is not a multiple of the period T.

```
ω = 2; Npts = 32;
Tmax = 4 (2 π / 2);
Δt = (Tmax / (Npts - 1) // N)
```

```
0.405367
```

In this case Npts Δt = 12.971744 $\neq n\pi$

The Hanning function $\mathbb{H}$(t) is given by

$$\mathbb{H}(t) = \frac{1}{2}(1 - \text{Cos}[2\pi t / T]) \tag{25}$$

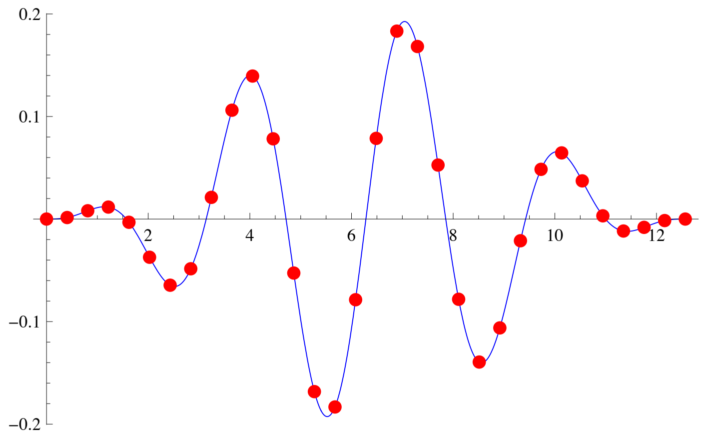
Here is the *Mathematica* implementation

```
ℍ[t_] := 1/2 (1 - Cos[2 π t / Tmax])
fdata4 = Table[ℍ[t] f[t] , {t, 0, Tmax, Δt}];
FData4 = Abs[Fourier[fdata4]];
```

```
dataPts4 = Join[{RGBColor[1, 0, 0], PointSize[0.02]},
   Map[Point[{#, ℍ[#] f[#]}] &, Range[0, Tmax, Δt]]];
```

Here is a plot of `ℍ[t] f[t]`

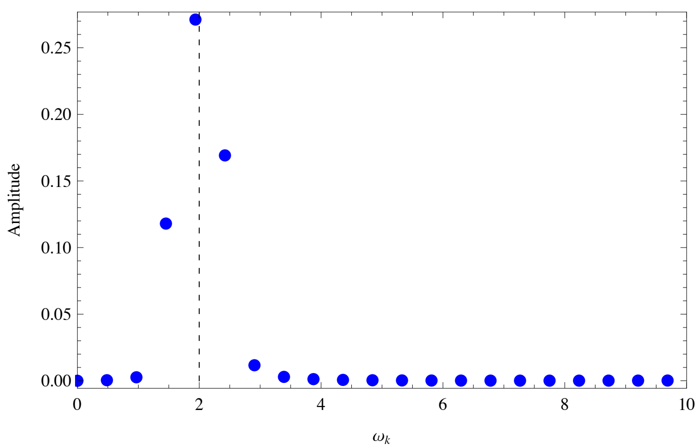
```
Plot[ℍ[t] f[t], {t, 0, Tmax}, PlotStyle → RGBColor[0, 0, 1], Epilog → dataPts4]
```



Next we take the Fourier Transform of the data

```
FData4 = Abs[Fourier[fdata4]];
freq = Table[k / (Npts Δt), {k, 0, Npts - 1}];
```

```
FPlot2 = ListPlot[Transpose[{2 π freq, FData4}],
  Joined → False, AxesOrigin → {0, 0}, PlotRange → {{0, 10}, All},
  PlotStyle → {RGBColor[0, 0, 1], PointSize[0.02`]}, FrameLabel → {"ωk", "Amplitude"},
  Frame → True, Epilog → {Dashing[Small], Line[{{2, 0}, {2, 0.3}}]}]
```



If we compare this plot with the one obtained in Example 3 we note that the leakage is reduced when $\omega_k > 4$. However, the non-zero components of the frequency are broaden or smeared about ω_4. Thus we have a compromise. The peak amplitude occurs at $\omega_4 = 1.9375$

Example 5 : Fourier Filtering

Consider a signal f(x)

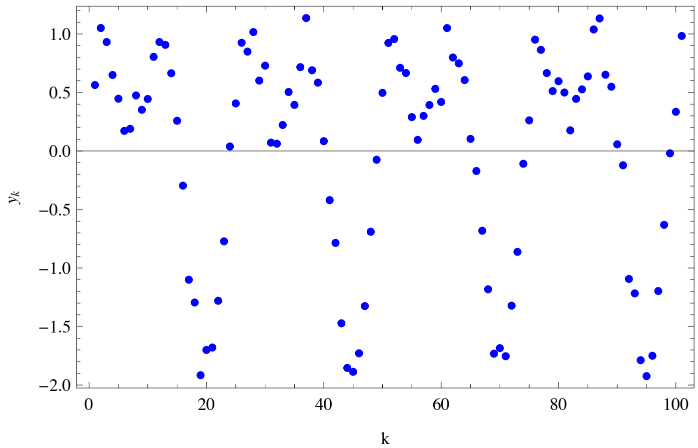
$$f(x) = \text{Sin}(4x) + 0.8\,\text{Cos}(8x) \tag{26}$$

that has been contaminated with random noise

```
noisyData = Table[Sin[4 x] + 0.8` Cos[8 x] + 0.5` (RandomReal[] - 0.5`), {x, 0, 2 π, 2 π/100}];
```

Here is a plot of the signal with noise

```
ListPlot[noisyData, PlotStyle → {Blue, PointSize[0.013`]},
 Frame → True, FrameLabel → {"k", "y_k"}]
```



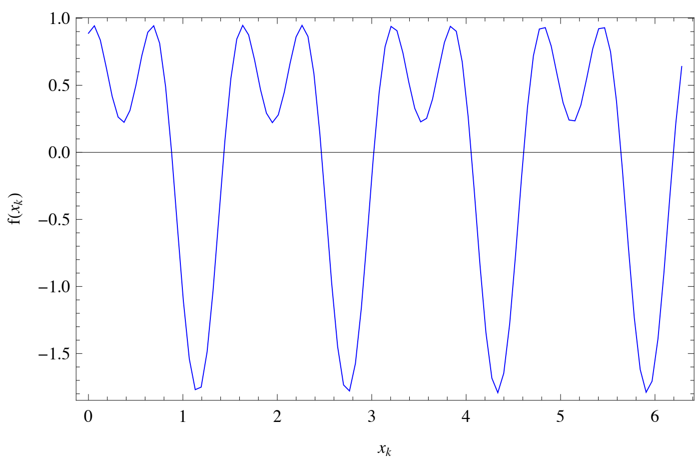
We can filter the data by taking the Fourier Transform of the signal and then removing components in the frequency domain that have amplitudes less than some cut-off value. In this case we take as the cut-off amplitude as 0.5

```
TransformData = Chop[Fourier[noisyData], 0.5];
```

Then we take the inverse Fourier Transform and remove any residual terms that are smaller than 10^{-10}

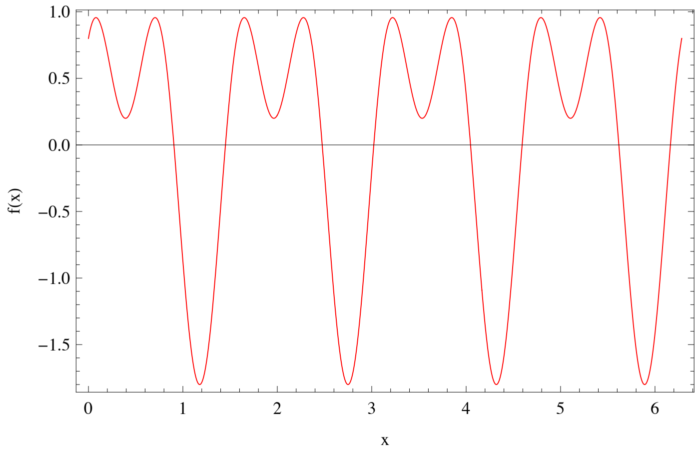
```
inverseTransform = Chop[InverseFourier[TransformData], 0.001];
```

```
plt1 = ListPlot[Transpose[{Range[0, 2 π, 2 π/100], inverseTransform}],
  PlotStyle → Blue, Joined → True, Frame → True, FrameLabel → {"x_k", "f(x_k)"}]
```



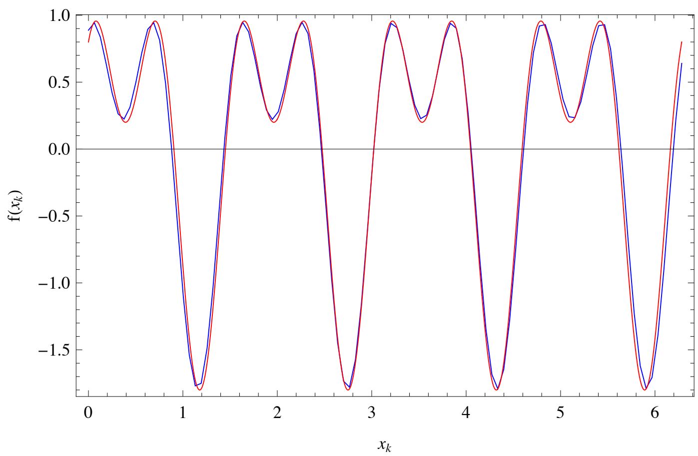
Here is the original data without the noise

```
plt2 = Plot[Sin[4 x] + 0.8` Cos[8 x], {x, 0, 2 π},
  PlotStyle → Red, Frame → True, FrameLabel → {"x", "f(x)"}]
```



If we combine the plots we can see that the Fourier Transformed data almost reproduces the originally data

```
Show[plt1, plt2]
```



We can also perform a spectral analysis of the signal. The data is sampled according to the following specifications

```
ω = 2; Npts = 128;
xmax = 2 π;
dx = (xmax / (Npts - 1) // N)
```

```
0.0494739
```

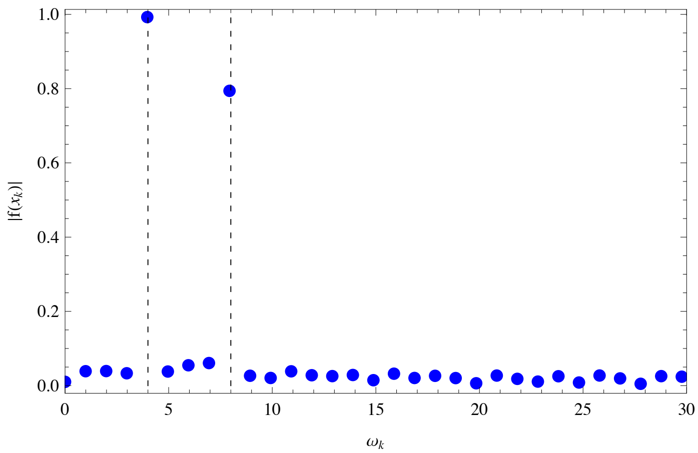
```
noisyData1 =
  Table[Sin[4 x] + 0.8` Cos[8 x] + 0.5` (RandomReal[] - 0.5`), {x, 0, xmax, dx}];
```

$$\texttt{FData5 = 2 Abs[Fourier[noisyData1]]} \Big/ \sqrt{\texttt{Npts}}\ ;$$

We generate the coordinates of our plot in terms of angular frequency

```
angularfreq = Table[(k / (Npts dx)) (2 π), {k, 0, Npts - 1}];
```

```
FPlot5 = ListPlot[Transpose[{angularfreq, FData5}],
  Joined → False, AxesOrigin → {0, 0}, PlotRange → {{0, 30}, All},
  PlotStyle → {RGBColor[0, 0, 1], PointSize[0.02`]},
  Frame → True, FrameLabel → {"ωk", "|f(xk)|"},
  Epilog → {Dashing[Small], Line[{{4, 0}, {4, 1}}], Line[{{8, 0}, {8, 1}}]}]
```



The peak information is

```
{{angularfreq[[5]], FData5[[5]]}, {angularfreq[[9]], FData5[[9]]}}
```

```
{{3.96875, 0.992697}, {7.9375, 0.794037}}
```

Thus the spectrum shows that there are two prominent frequencies $\omega \approx 4$ (3.9688) and $\omega \approx 8$ (7.938) in our noisy sample with amplitudes 1 (0.993) and 0.8 (0.794).

Sometimes it is nice to modify the plots for a specific application. For example, in the spectral plot shown above, it would be nice to show each data as a "spectral line". In *Mathematica* this can be readily done by writing a function that does precisely this. I have called this function SpectralPlot with the following syntax

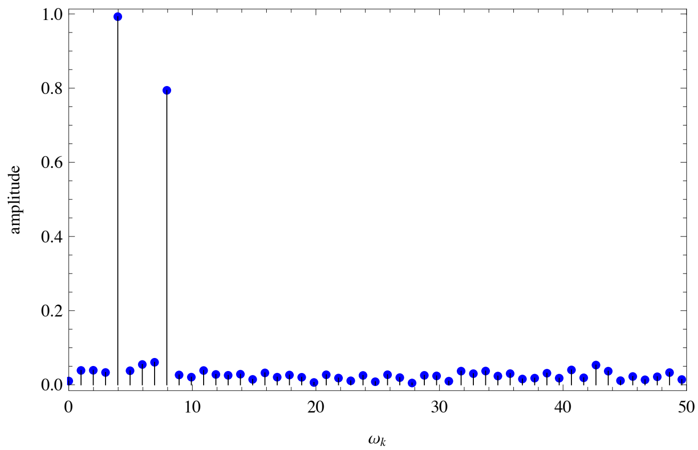
```
SpectralPlot[data, x_AxisLabel, y_AxisLabel, x_AxisRange]
```

Here is the function

```
SpectralPlot[data_, xAxisLabel_String, yAxisLabel_String, dataRange_] :=
 Module[{datapts}, ListPlot[data, PlotRange → {dataRange, All},
   PlotStyle → {Blue, PointSize[Medium]}, Frame → True, FrameLabel ->
    {xAxisLabel, yAxisLabel}, Epilog → data /. {x_, y_} -> Line[{{x, y}, {x, 0}}]]]
```

Here is an example using that function

```
SpectralPlot[Transpose[{angularfreq, FData5}], "ωk", "amplitude", {0, 50}]
```



Final Comments

In these notes we have covered the main ideas behind the Fourier Transform and how one interprets the data. There is a lot more that we have not covered but the reader should now have a clear understanding of the principal manipulations that one does using Fourier Transforms. These ideas can be readily extended to higher dimensions.

References

There is a vast literature on Fourier Transforms and the discrete Fourier transform. One source that I have found to be excellent is given below

- E. Oran Brigham, *The Fast Fourier Transform*, Prentice Hall, 1974