

Fourier transforms

Created using Maple 18.00

Jake Bobowski

```
> restart;
with(StringTools) :
with(Statistics) :
with(plots) :
```

```
FormatTime("%m-%d-%Y, %H:%M");
"01-08-2018, 09:15"
```

(1)

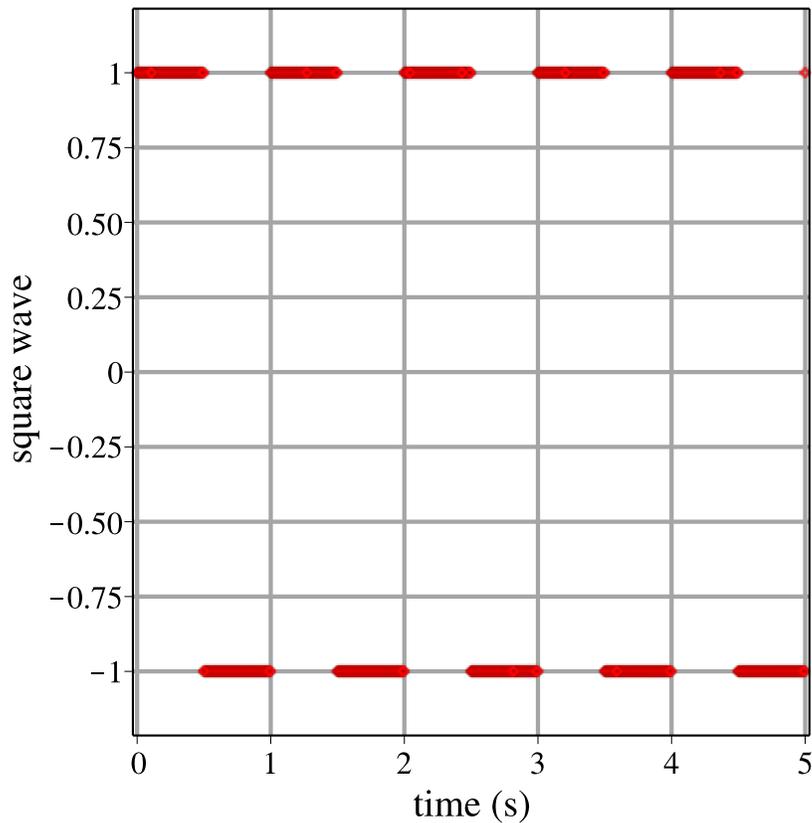
This example requires that the *DiscreteTransforms* package be loaded.

```
> with(DiscreteTransforms) :
```

Import and plot some square wave data that will be used in this example. The fundamental frequency is 1 Hz.

```
> square := readdata("G:\\UBCO\\2017-2018\\people page\\2017\\Maple\\squareWave.dat",
    [float, float]) :
nops(square);
y := [seq(square[i, 2], i = 1 ..nops(square)) ] :
Time := [seq(square[i, 1], i = 1 ..nops(square)) ] :
squarePlot := ScatterPlot(Time, y, color = red, axes = boxed, view = [0 .. 14, 0 .. 175], labels
    = [typeset("time (s)", typeset("square wave")), labeldirections = ["horizontal",
    "vertical"], tickmarks = [8, 8], axesfont = [Times, 12], labelfont = [Times, 14], axis
    = [gridlines = [thickness = 2]]) :
display(squarePlot, view = [0 ..5, -1.2 ..1.2]);
```

10000



Recall that:

$$Ae^{i\theta} = A (\cos \theta + i \sin \theta) = x + iy$$

such that:

$$A = \sqrt{x^2 + y^2}$$

Below Maple is used to evaluate the Fourier transform of the square wave data to examine the frequency content of the signal. When evaluating the Fourier transform Maple outputs the x and y values and you must evaluate A as shown above to find the amplitude of the frequencies present in the signal.

Let's see an example below. The Maple command is *FourierTransform* and argument of the command are the y -values input as a *Vector* rather than a simple list.

```
> Squarefft := FourierTransform(Vector(y), normalization = full);
Squarefft[23];
Re(Squarefft[23]);
```

```
Im(Squarefft[23]);
```

```
Squarefft := [ 1 .. 10000 Vectorcolumn
               Data Type: anything
               Storage: rectangular
               Order: Fortran_order ]
```

```
7.95128765548198 10-19 - 9.18051253957727 10-19 I
7.95128765548198 10-19
-9.18051253957727 10-19 (2)
```

Here is the magnitude (A) calculation.

```
> magnitude := [seq(sqrt((Re(Squarefft[i]))2 + (Im(Squarefft[i]))2), i = 1 .. nops(y))]:
nops(magnitude);
magnitude[23];
```

```
10000
1.21451548310244 10-18 (3)
```

Now we need to generate a list of the frequencies that the magnitudes correspond to. The frequency span of the Fourier transformed data (or the max frequency) is determined by the time interval Δt between the measured data points. The smaller the Δt , the larger the frequency span. The frequency resolution of the Fourier transformed data (or the spacing between the frequency data points) are determined by the total number of data points N that are collected.

Below Δt is determined.

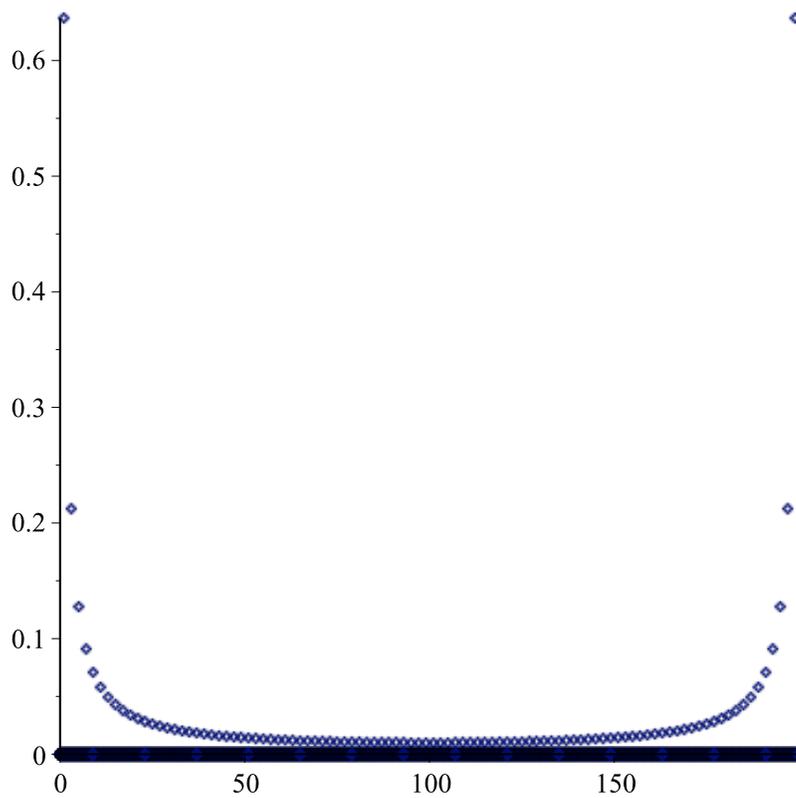
```
>  $\Delta t := Time[2] - Time[1];$ 
 $\Delta t := 0.005$  (4)
```

The max frequency is $1/\Delta t$ and the frequency resolution is $1/N\Delta t$. Below the list of frequencies is generated.

```
> freq := [evalf(seq( (i / ((nops(Time)) *  $\Delta t$ ), i = 0 .. nops(Time) - 1)))] :
nops(freq);
freq[51];
10000
1.000000000 (5)
```

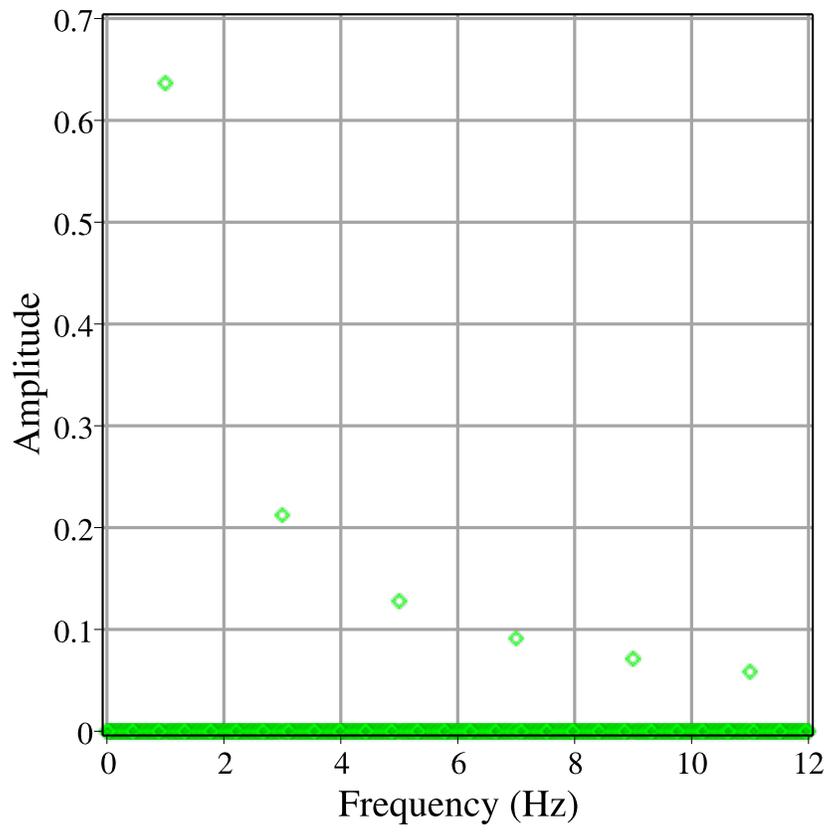
Here's the scatter plot of the Fourier transformed data. Notice that the Fourier transform high-frequency data is just a mirror-image of the low-frequency data. For interpreting our square waveform, we will look only from 0 to 100 Hz. The high-frequency data (above 100 Hz) is due to the negative frequency components. Consider $2 \cos(\omega t) = e^{j\omega t} + e^{-j\omega t}$ and $2j \sin(\omega t) = e^{j\omega t} - e^{-j\omega t}$, from this one can interpret cos and sin as being made up of positive and "negative" frequency componets.

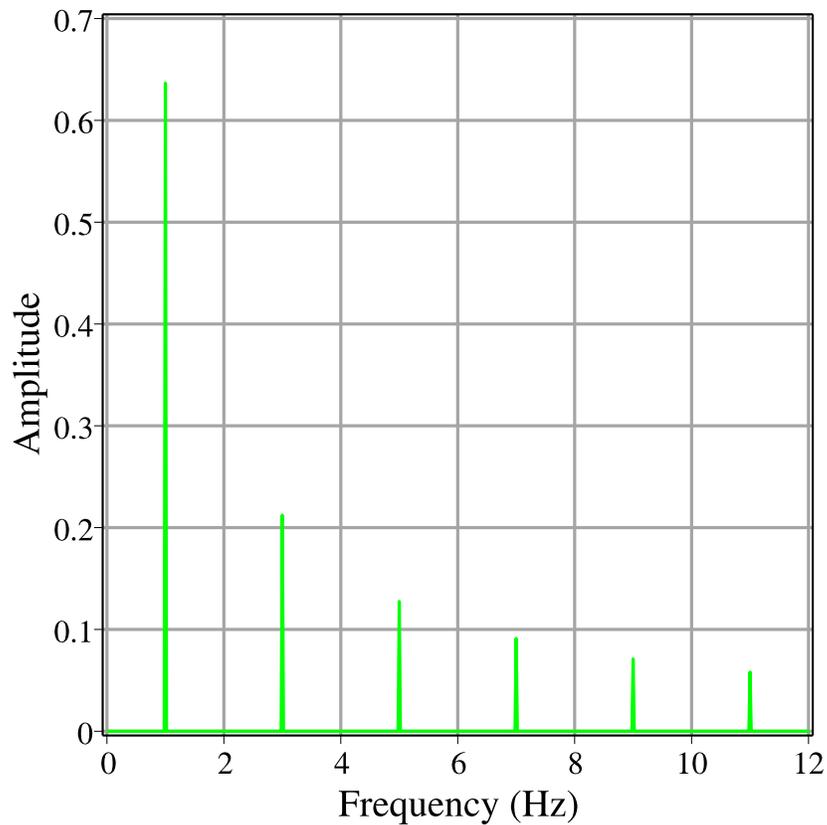
```
> ScatterPlot(freq, magnitude);
```



Let's format the scatter plot so that we may gain additional insight. In this case, one may wish to connect the points with lines using *plot*.

```
> ScatterPlot(freq, magnitude, color = green, axes = boxed, view = [0 .. 12, 0 .. 0.7], labels
  = [typeset("Frequency (Hz)", typeset("Amplitude")), laeldirections = ["horizontal",
  "vertical"], tickmarks = [7, 8], symbolsize = 12, axesfont = [Times, 12], laelfont = [Times,
  14], axis = [gridlines = [thickness = 1]]);
plot(freq, magnitude, color = green, axes = boxed, view = [0 .. 12, 0 .. 0.7], labels = [typeset(
  "Frequency (Hz)", typeset("Amplitude")), laeldirections = ["horizontal", "vertical"],
  tickmarks = [7, 8], axesfont = [Times, 12], laelfont = [Times, 14], axis = [gridlines
  = [thickness = 1]]);
```





As expected, the Fourier transform of the square wave consists of the fundamental frequency (1 Hz), and all of the odd harmonics (3, 5, 7, 9 Hz, ...). The amplitude of the harmonics fall as $1/N$, where N is odd and 1 represents the first harmonic, 3 is the third harmonic... In our example, the square wave has an amplitude of 1. As a result, the expected amplitude of fundamental frequency component is $4/\pi = 1.2732$. The result we get from Maple is exactly half that (0.6366). The factor of two difference comes from the fact that Maple has calculated the Fourier series in terms of complex exponentials rather than trig functions (sines and cosines). The complex exponential series has both positive and negative frequency terms and the amplitude of each term is half the amplitude that one gets for the Fourier series in terms of sines and cosines. The ratios of the harmonic amplitudes are right.

```
> freq[51];
   magnitude[51];
   freq[151];
   magnitude[151];
   freq[251];
   magnitude[251];
   freq[351];
   magnitude[351];
   print("ratios");
    $\frac{\text{magnitude}[51]}{\text{magnitude}[151]}$ ;
```

```
magnitude[51] ,  
magnitude[251] ,  
magnitude[51] ,  
magnitude[351] ,
```

```
1.000000000  
0.636645953060006  
3.000000000  
0.212285150958168  
5.000000000  
0.127454948431824  
7.000000000  
0.0911292001614984  
"ratios"  
2.99901312073146  
4.99506657758801  
6.98619050679416
```

(6)

