# Fourier Notes

On discrete and continuous Fourier analysis, or how to add up waves to get anything you like.

**Jim Mahoney, Marlboro College, Fall 1999**

## Discrete Fourier Series

Let $x_n$ be coordinates on a uniform grid with spacing $\Delta x$ over a length $L = N\,\Delta x$, *e.g.*

$$x_n = n\,\Delta x\,, \text{ where } 0 < n \le N \text{ and} \qquad \Delta x = \frac{L}{N}. \tag{1}$$

Likewise, let $y_n$ be a discrete set of points defined on the grid, $Y_n = f(x_n)$, like this.

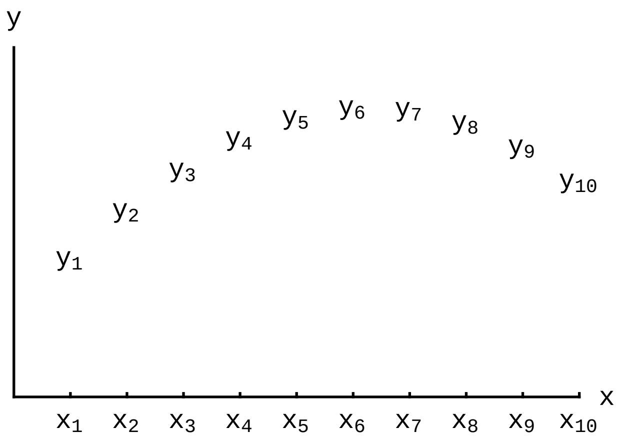
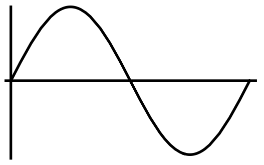


**Figure 1**

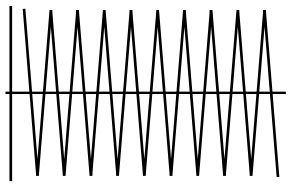
We would like to express this function $f$ as a sum of waves - sin and cosine functions, or complex exponentials.

The longest wavelength $\lambda$ we will allow is $L$, the length of the $x$ interval, since that will always give us solutions which are periodic in $L$. Essentially that means we are only considering functions such that $f(x+L) = f(x)$. (For some problems, such as a rope which is fixed at both ends, one chooses instead $\frac{\lambda}{2} = L$; however, that is not what we are doing here.)



Longest wavelength.

The shortest wavelength is $2\,\Delta x$, which corresponds to an oscillation up and down as fast as possible on the grid, like this.

 ... Shortest wavelength.

Choosing the other possible wavelengths to match the other available grid spacings ($3\,\Delta x$, $4\,\Delta x$, ...), we find $\frac{N}{2}$ discrete wavelengths. Since we also have two independent phases (either sin and cosine or positive and negative complex exponentials), the total number of discrete waves is just $N$, the same as the number of independent $y$ values. More on this in a moment.

It is convenient to label each wave by its wavenumber $k = \frac{2\pi}{\lambda}$ with a spacing $\Delta k = \frac{2\pi}{L}$. Then the possible values of $k$ are

$$k_j = \frac{2\pi}{\lambda_j} = j\,\Delta k = j\,\frac{2\pi}{L} = j\,\frac{2\pi}{N\,\Delta x}\text{, where } -\frac{N}{2} < j \leq \frac{N}{2} \text{ and} \qquad (2)$$
$$\Delta k = \frac{2\pi}{L}.$$

Let's look at the number of $x_n$ and $k_j$ values a bit more closely and count the number of independent quantities. In the first place, since $f(x)$ is periodic, $y_0$ is the same as $y_N$; therefore, we only need $n = 0, 1, 2, \ldots, N-1$. Second, the highest possible frequency is $k_{\max} = \frac{\pi}{\Delta x} = \frac{N\pi}{L}$ has only one phase, not two. That is because it is not possible to shift sideways by a half cycle at that frequency; the grid spacing is not small enough. Thus $j = \frac{N}{2}$ and $j = -\frac{N}{2}$ refer to the same wave. This would appear to imply that there are only N-1 different values of $k_j$. However, there is one more value of $k$ which we have not yet mentioned : $k = 0$, which allows us to include a "zero frequency" or "DC offset" term.

Furthermore, since $x$ is periodic, we could have instead chosen $x$ symmetric about $x = 0$ with

$$x_n = n\,\Delta x\text{, where } -N/2 < n \leq N/2\,, \qquad (3)$$

which is the way we'll usually write these limits. But equation (1) is just as correct, since $x_N = x_0$ and $x_{-N/2} = x_{N/2}$.

With those details out of the way, we're now ready to define our **discrete Fourier transform** :

Please note that different authors put the factors of $N$ and $2\pi$ in different places; check your conventions before plugging into someone else's equations. I'm choosing conventions which (a) make the transition to the integral forms clear, and (b) are commonly seen among physicists. (Note that engineers tend to use the symbal "$j$" for $\sqrt{-1}$ , rather than "$i$", and also tend to use $f$ (frequency in units of cycles/sec) rather than the physicist's $k$ or $\omega$ (radians/meter or radians/sec).

$$\boxed{\begin{aligned} Y_n &= \sum_{j=-N/2+1}^{N/2} C_j\, e^{i k_j x_n} \frac{\Delta k}{2\pi} \\ C_j &= \sum_{n=-N/2+1}^{N/2} Y_n\, e^{-i k_j x_n} \Delta x \end{aligned}}$$

The symmetry of these two equations is quite striking. Essentially what we have is a discrete description of the function *f* in terms of N coefficients, which we can choose as either the numbers $Y_n$or $C_j$, which specify *f* in terms of x coordinate values or sinusoidal waves.

By inserting our definitions of $x_n$, Δx, $k_j$, and Δk, we can also write this as

$$\boxed{\begin{aligned} Y_n &= \frac{1}{L} \sum_{j=-N/2+1}^{N/2} C_j\, e^{2\pi i\,(j\,n)/N} \\ C_j &= \frac{L}{N} \sum_{n=-N/2+1}^{N/2} Y_n\, e^{-2\pi i\,(j\,n)/N} \;. \end{aligned}} \tag{5}$$

If we then also re-define the Fourier coefficients to include the length *L*,

$$G_j = C_j \big/ L \tag{6}$$

we get this even simpler form,

$$\boxed{\begin{aligned} Y_n &= \sum_{j=-N/2+1}^{N/2} G_j\, e^{2\pi i\,(j\,n)/N} \\ G_j &= \frac{1}{N} \sum_{n=1}^{N} Y_n\, e^{-2\pi i\,(j\,n)/N}, \end{aligned}} \tag{7}$$

which is the kind of expression that actually gets coded into computer programs. It turns out that there's a partiuclarly fast algorithm for calculating this when *N* is a power of 2, called the "Fast Fourier Transform".

Are these two really inverses of each other? To check, substitute one inside the other - without forgetting that the sum is over a dummy variable. Replacing, for example, *j* with *j*' in the first expression, and substituting into the second gives

$$G_j \overset{?}{=} \frac{1}{N} \sum_{n=1}^{N} \left( \sum_{j'=-N/2+1}^{N/2} G_{j'}\, e^{2\pi i\,(j' n)/N} \right) e^{-2\pi i\,(j n)/N}$$

which should be an identify if all this is really working. Well, of course it is. To see this, swap the order of the summation signs, putting everything that depends on *n* on the inside. Then

$$G_j \overset{?}{=} \frac{1}{N} \sum_{j'=-N/2+1}^{N/2} G_{j'} \left( \sum_{n=1}^{N} e^{2\pi i\,(j'-j)\,n/N} \right)$$

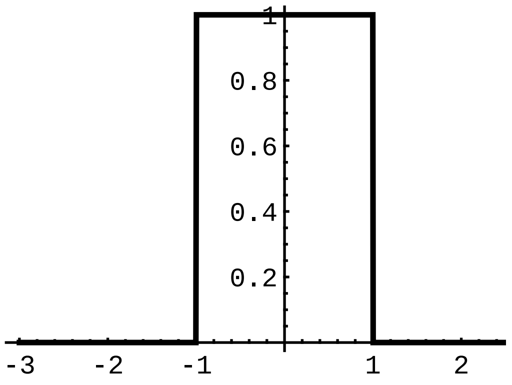
The part in the parenthesis is 0 if $j \neq j'$, since in that case we have the sum of complex numbers spaced equally around the unit circle, which all cancel out. On the other hand, if $j = j'$, then it's just $N$, since we're adding $N$ copies of $e^0$which is 1. So our result checks.

$$G_j \overset{?}{=} \frac{1}{N} \sum_{j'=-N/2+1}^{N/2} G_{j'}\, N\, \delta_{j-j'} = G_j \;.$$

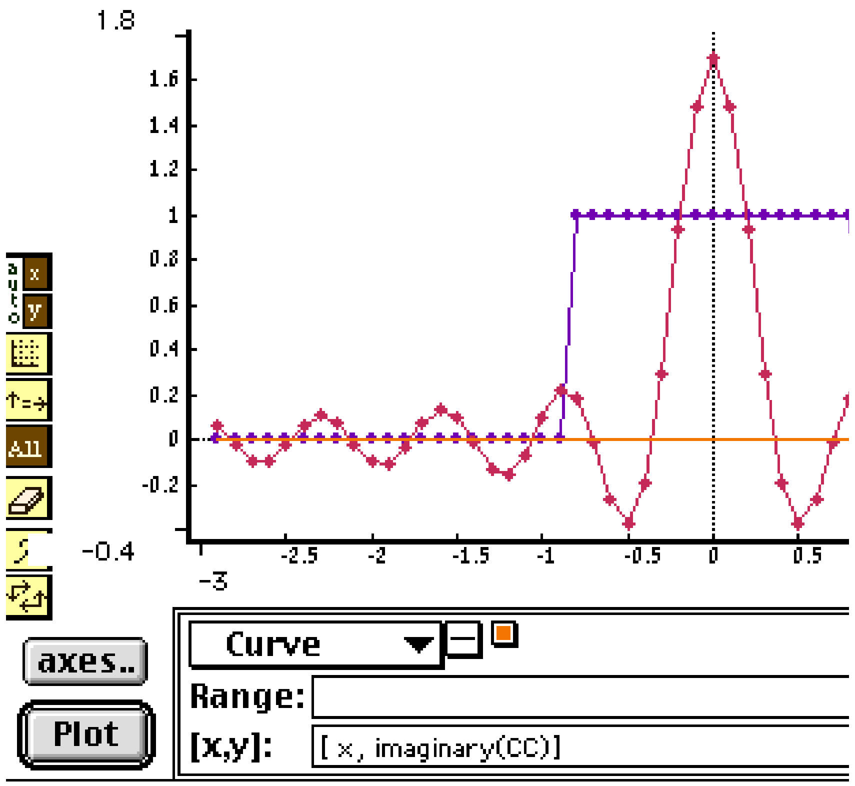
## An Example

So now that we have the discrete formulas, let's do an example. A typical one is the "hat" function,

$$f(x) = \begin{cases} B & \text{for } -a \leq x \leq a \\ 0 & \text{otherwise} \end{cases} :$$



To discretize this, let's set set values like shown in this picture, with $B = 1$, $\Delta x = 0.1$, $N = 60$, $a = 1.0$. Applying the formulas above gives the following Fourier Transform:



Range:
[x,y]: [ x, imaginary(CC)]

Here the purple shows the original function $Y_n$ while the red and orange show the real and imaginary parts of the Fourier components $C_j$, that is, the amplitude of the various sinusoids which when added up give the purple curve. Cool, eh? As the purple $Y$ gets wider, the red $C$ gets narrower, and vice-versa.

You can in fact do this sum analytically - but I'll leave that up to you to try. (Hint: the imaginary part of the complex exponential cancels out becuase it's odd, while the even parts add up in a series which you can sum with the same formula you use to sum power series sums like 1/2 + 1/4 + 1/8 + ... ).

## Linear Algebra

Now all that's well and good, but what's going on and where did those formulas come from anyway? I'm glad you asked. We're actually looking at a change of basis here, in just exactly the same way that the co-ordinate description of a vector changes when you rotate your co-ordinate system. Once we have the equations for that kind of transformation, we'll see that in fact they look just like the ones we just wrote down - and in fact that explains why the Fourier Transform and Inverse Fourier Transform look alike, becuase they're both "rotations" (forward and back) to a new basis.

So let's say we have a vector $\vec{r} = [x,\ y,\ z]$ in the $\hat{\boldsymbol{i}}$, $\hat{\boldsymbol{j}}$, $\hat{\boldsymbol{k}}$ basis, which we would like to write in terms of a new basis, say $\hat{\boldsymbol{e}}_1$, $\hat{\boldsymbol{e}}_2$, $\hat{\boldsymbol{e}}_3$. In other words, we want to find $a_1$, $a_2$, $a_3$ such that $\vec{r} = a_1\,\hat{\boldsymbol{e}}_1 + a_2\,\hat{\boldsymbol{e}}_2 + a_1\,\hat{\boldsymbol{e}}_3$. How do we get the a's? Well if the **e**'s are orthonormal - that is, if their dot product is 1 with themself and 0 with each other - then it turns out to be pretty easy: we just dot both sides with each **e** in turn, like this:

$$\vec{r} = x\,\hat{\boldsymbol{i}} + y\,\hat{\boldsymbol{j}} + z\,\hat{\boldsymbol{k}}$$
$$\hat{\boldsymbol{e}}_1 \cdot \vec{r} = \hat{\boldsymbol{e}}_1 \cdot \left(x\,\hat{\boldsymbol{i}} + y\,\hat{\boldsymbol{j}} + z\,\hat{\boldsymbol{k}}\right) =$$

$$\hat{\boldsymbol{e}}_1 \cdot \left(a_1 \hat{\boldsymbol{e}}_1 + a_2 \hat{\boldsymbol{e}}_2 + a_1 \hat{\boldsymbol{e}}_3\right) = a_1 \hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{e}}_1 + a_2\, 0 + a_3\, 0 = a_1$$

or

$$a_1 = x\, \hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{i}} + y\, \hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{j}} + z\, \hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{k}}$$

How does this apply to the Fourier Series? Quite simply: the original vector is our set of number $[Y_n]$ that describe the function, and the new basis vectors (as written in the $x$ basis) are the sinusoids $e^{ikx}$. Thus the formula for the Fourier $C_j$'s is just the same as the last line in equation (8) above ; the complex exponentials are the basis dot products, there's a sum over the various components in the original basis, and we get out one coefficient of the vector in the new basis.

(This needs to be fleshed out.)

All right, now on to what this looks like for continuous functions.

## Continuous Fourier Integral

Well, the short version is that it's just the same as equation (4) in the limit where $dx \to 0$ and $L \to \infty$. The equations look like this.

$$\boxed{\begin{aligned} Y(x) &= \int_{-\infty}^{+\infty} C(k)\, e^{i k x} \frac{dk}{2\pi} \\ C(k) &= \int_{-\infty}^{+\infty} Y(x)\, e^{-i k x}\, dx \end{aligned}} \qquad (9)$$

Here the Fourier Coefficients $C(k)$ give the (complex) amplitude of a continuous range of possible sinusoidal functions, each with wavenumber k, which when added together give the original function $Y(x)$.

A common variation on this theme is a mixed continuous/discrete case, in which typically $Y(x)$ is continuous over a finite range of $x$ values with a fixed boundry condition (*i.e.* a string). This kind of situation leads to a discrete set of Fourier Coefficients $C_j$ which typically give the modes of vibration of the string. Since this case is well covered in most texts (including ours), I'll leave it out here.

## More Examples

One particularly useful function is the Dirac Delta function $\delta(x)$ ...

- end -

## Scratch Space

```
<< DiscreteMath`KroneckerDelta`
```

```
j = 0;
        Sum[ Exp[2 Pi I j k / N], {k, 1, N}]
```

N

```
j = 1;          Sum[ Exp[2 Pi I j k / N], {k, 1, N}]
```

0

```
j =.;          Sum[ Exp[2 Pi I j k / N], {k, 1, N}]
```

$$\frac{E^{\frac{2 I j \pi}{N}} \left(-1 + E^{I j \pi}\right) \left(1 + E^{I j \pi}\right)}{\left(-1 + E^{\frac{I j \pi}{N}}\right) \left(1 + E^{\frac{I j \pi}{N}}\right)}$$

```
a = (1 / N) Sum[ G[j] Exp[2 Pi I j n], {j, -N / 2 + 1, N / 2}]
```

$$\frac{\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} G[j]\ \mathrm{Exp}[2 \pi I j n]}{N}$$

```
Sum[ a Exp[ -2 Pi I j n] , {n, 1, N}]
```

$$\sum_{n=1}^{N} a\ \mathrm{Exp}[-2 \pi I j n]$$

$$\sum_{n=1}^{N} \left( (1 / N)\ \mathrm{Sum}[\ G[j]\ \mathrm{Exp}[2\ \mathrm{Pi}\ I\ j\ n], \{j, -N / 2 + 1, N / 2\}] \right)$$

$$\mathrm{Exp}[-2 \pi I j' n]$$

$$\sum_{n=1}^{N} \frac{\left(\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} G[j]\ \mathrm{Exp}[2 \pi I j n]\right) \mathrm{Exp}[-2 \pi I j' n]}{N}$$

**Doesn't seem that Mathematica can convert
this sum of sums into a single sum by itself (! ?)
Disappointing.**

```
dx = L / N
```

$$\frac{L}{N}$$

```
dk = 2 Pi / L
```

$\frac{2\pi}{L}$

```
(N / 2) (dk)
```

$\frac{N\pi}{L}$

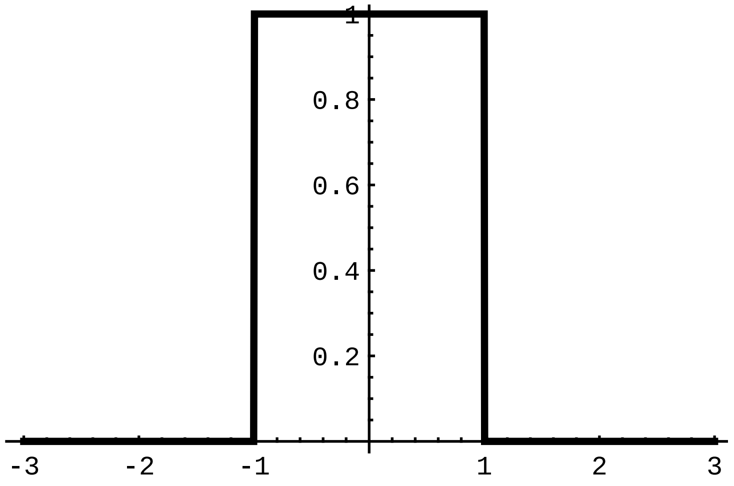
```
dk dx
```

$\frac{2\pi}{N}$

## sigma plot

```
sigma[x_] := (1 + Sign[a - x]) (1 + Sign[a + x])/4

Plot[sigma[x] /. a -> 1, {x, -3, 3}, PlotStyle -> Thickness[0.01]]
```



- Graphics -

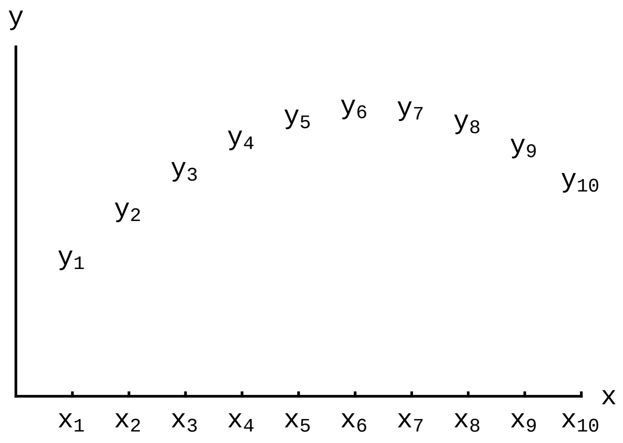
### ■ $y_n$ vs $x_n$plot

```
xNums = Table[{j, x_j}, {j, 1, 10}]
```

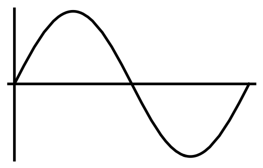
$\{\{1, x_1\}, \{2, x_2\}, \{3, x_3\}, \{4, x_4\}, \{5, x_5\},$
$\{6, x_6\}, \{7, x_7\}, \{8, x_8\}, \{9, x_9\}, \{10, x_{10}\}\}$

```
yNums = Table[{Sin[.3 + j * 0.2], y_j}, {j, 1, 10}]
```

```
yText = Table[
   {Text[yNums[[j, 2]], {xNums[[j, 1]], yNums[[j, 1]]}]}, {j, 1, 10} ]
Show[Graphics[yText], PlotRange -> {{0, 10}, {0, 1.2}},
 Axes -> {True, True}, Ticks -> {xNums, None}, AxesLabel -> {x, y}]
```

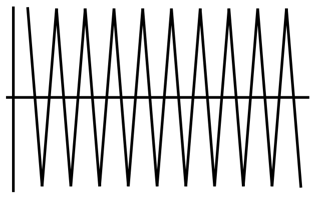


```
Plot[Sin[x], {x, 0, 2 Pi}, Ticks -> False, ]
```



```
- Graphics -
```

```
ListPlot[Level[Table[{1, -1}, {10}], {2}], Ticks → False, Joined → True]
```



```
- Graphics -
```

```
t = Table[{1, 2}, {20}]
```

```
{{1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2},
 {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2},
 {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}}
```

```
t[[1]]
```

```
{1, 2}
```

```
Level[t, {2}]
```

```
{1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2}
```

**--- end of scratch ---**