10. Fourier Methods

In many diverse fields physical data is collected or analysed as Fourier components.

In this section we briefly discuss the mathematics of Fourier series and Fourier transforms.

1. Fourier Series

Any 'well-behaved' function $f(x)$ can be expanded in terms of an infinite sum of sines and cosines. The expansion takes the form:

$$\boxed{f(x) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx}$$

Joseph Fourier
(1768-1830)

The Fourier coefficients are given by the formulae:

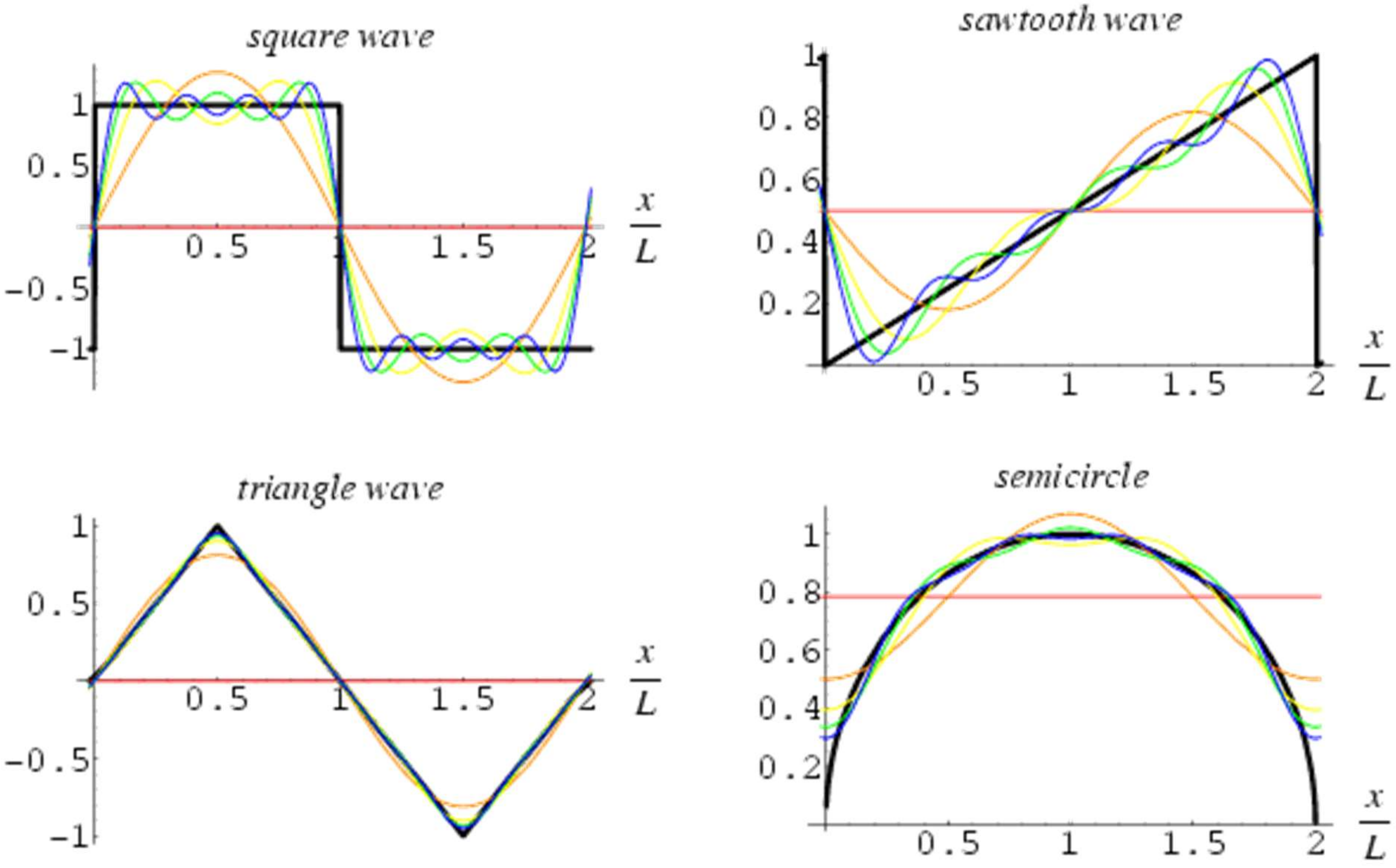
$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nxdx$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nxdx$$

These formulae follow from the **orthogonality properties** of sin and cos:

$$\int_{-\pi}^{\pi} \sin mx \sin nxdx = \pi\delta_{mn} \qquad \int_{-\pi}^{\pi} \cos mx \cos nxdx = \pi\delta_{mn} \qquad \int_{-\pi}^{\pi} \sin mx \cos nxdx = 0$$



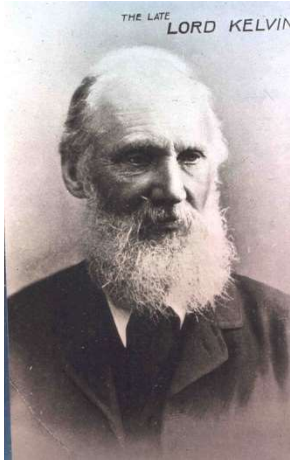
Some examples from Mathworld, approximating functions with a finite number of Fourier series terms

The Fourier series can also be written in complex form:

$$\boxed{f(x) = \sum_{n=-\infty}^{\infty} A_n e^{inx}}$$

where
$$A_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x) e^{-inx}\,dx$$

and recall that
$$e^{inx} = \cos nx + i \sin nx$$
$$e^{-inx} = \cos nx - i \sin nx$$



"Fourier's Theorem is not only one of the most beautiful results of modern analysis, but it is said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics"

Fourier Transform: Basic Definition

The **Fourier transform** can be thought of simply as extending the idea of a Fourier series from an infinite sum over discrete, integer Fourier modes to an infinite integral over continuous Fourier modes.

Consider, for example, a physical process that is varying in the **time domain**, i.e. it is described by some function of time $h(t)$.

Alternatively we can describe the physical process in the **frequency domain** by defining the Fourier Transform function $H(f)$.

It is useful to think of $h(t)$ and $H(f)$ as two different representations of the same function; the information they convey about the underlying physical process should be equivalent.

We define the Fourier transform as

$$\boxed{H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt}$$

and the corresponding inverse Fourier transform as

$$\boxed{h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df}$$

If time is measured in seconds then frequency is measured in cycles per second, or Hertz.

In many physical applications it is common to define the frequency domain behaviour of the function in terms of **angular frequency** $\omega = 2\pi f$

This changes the previous relations accordingly:

$$\boxed{H(\omega) = \int_{-\infty}^{\infty} h(t) e^{2\pi i \omega t} dt}$$

$$\boxed{h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{-2\pi i \omega t} d\omega}$$

Thus the symmetry of the previous expressions is broken.

Fourier Transform: Further properties

The FT is a **linear operation**:

(1) the FT of the sum of two functions is equal to the sum of their FTs

(2) the FT of a constant times a function is equal to the constant times the FT of the function.

If the time domain function $h(t)$ is a real function, then its FT is **complex**.

However, more generally we can consider the case where $h(t)$ is also a complex function – i.e. we can write

$$\boxed{h(t) = h_R(t) + ih_I(t)}$$

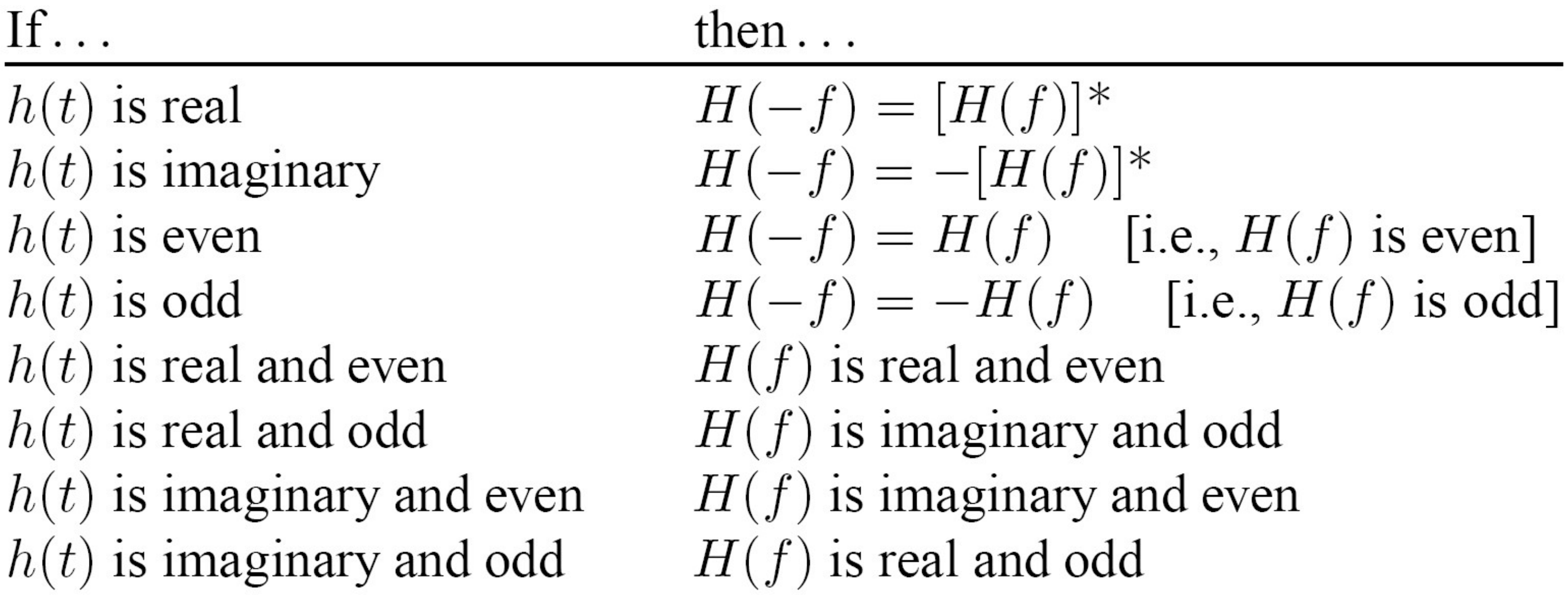
Real part

Imaginary part

$h(t)$ may also possess certain symmetries: **even function** $h(t) = h(-t)$

odd function $h(t) = -h(-t)$

The following properties then hold:



If...	then...
$h(t)$ is real	$H(-f) = [H(f)]^*$
$h(t)$ is imaginary	$H(-f) = -[H(f)]^*$
$h(t)$ is even	$H(-f) = H(f)$ [i.e., $H(f)$ is even]
$h(t)$ is odd	$H(-f) = -H(f)$ [i.e., $H(f)$ is odd]
$h(t)$ is real and even	$H(f)$ is real and even
$h(t)$ is real and odd	$H(f)$ is imaginary and odd
$h(t)$ is imaginary and even	$H(f)$ is imaginary and even
$h(t)$ is imaginary and odd	$H(f)$ is real and odd

See Numerical Recipes, Section 12.0

Note that in the above table a star $(*)$ denotes the **complex conjugate**,

i.e. if $z = x + i\,y$ then $z^* = x - i\,y$

For convenience we will denote the FT pair by $h(t) \Leftrightarrow H(f)$

It is then straightforward to show that

$$h(at) \Leftrightarrow \frac{1}{|a|} H(f/a)$$

"time scaling"

$$\frac{1}{|b|} h(t/b) \Leftrightarrow H(bf)$$

"frequency scaling"

$$h(t - t_0) \Leftrightarrow H(f)\, e^{2\pi i f t_0}$$

"time shifting"

$$h(t)\, e^{-2\pi i f t_0} \Leftrightarrow H(f - f_0)$$

"frequency scaling"

Suppose we have two functions $g(t)$ and $h(t)$

Their **convolution** is defined as

$$\boxed{(g*h)(t)=\int_{-\infty}^{\infty}g(s)h(t-s)\,ds}$$

We can prove the **Convolution Theorem** $(g*h)(t)\Leftrightarrow G(f)H(f)$

i.e. the FT of the convolution of the two functions is equal to the product of their individual FTs.

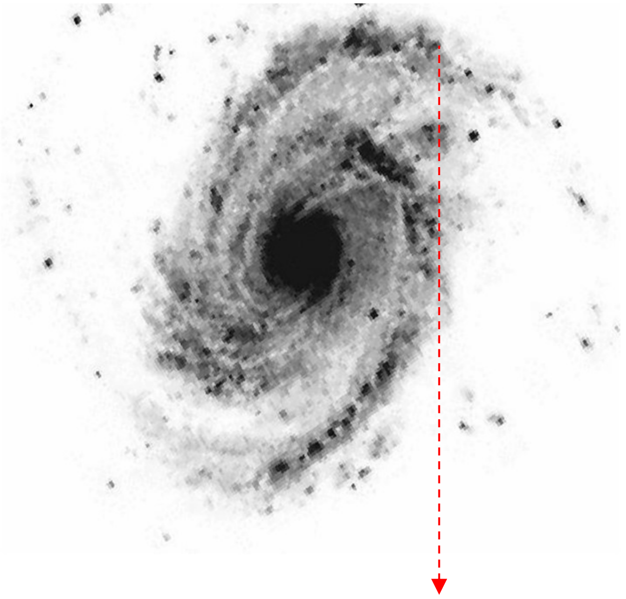
Example application: the observed spectrum of light from a galaxy is a convolution of stellar spectral templates ‘smeared out’ by the line of sight velocity distribution and the

$$G(u) \quad \propto \quad \int_{-\infty}^{\infty} F(\mathrm{v}_{\mathrm{LOS}})\,S(u-\mathrm{v}_{\mathrm{LOS}})\,d\,\mathrm{v}_{\mathrm{LOS}}$$

Galaxy spectrum

LOSVD

‘stellar spectra



This is a particular type of integral equation:

$$g(y) \quad = \int_{-\infty}^{\infty} f(x)\,s(y-x)\,dx$$

‘Data’ function

‘Source’ function

‘Kernel’ function

So by the convolution theorem:

$$\boxed{\tilde{G}(k) \quad = \quad \tilde{F}(k)\,\tilde{S}(k)}$$

and

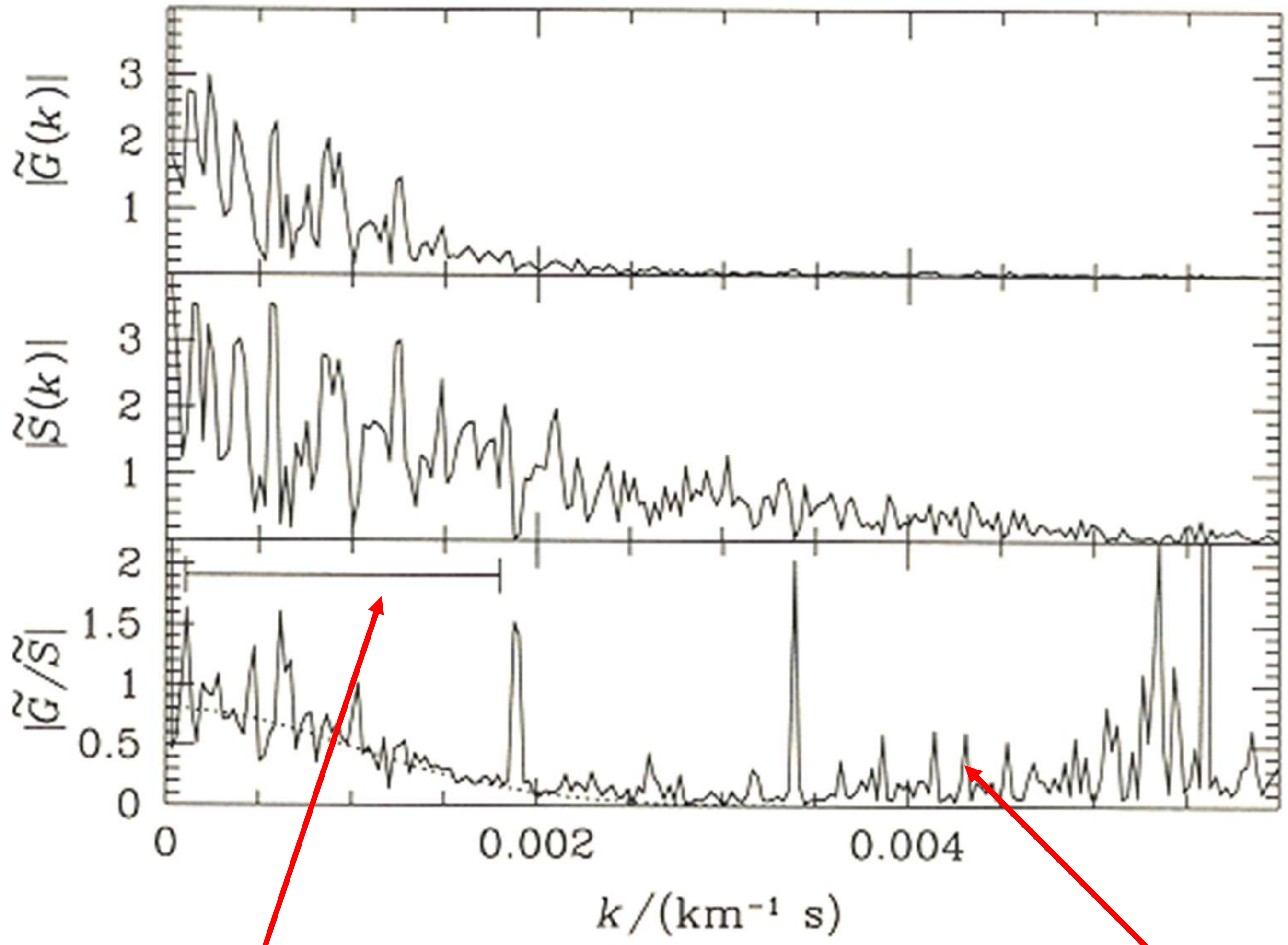
$$\boxed{F\left(\mathrm{v}_{\mathrm{LOS}}\right) \quad = \quad \tilde{F}^{-1}\left[\frac{\tilde{G}(k)}{\tilde{S}(k)}\right]}$$

Inverse Fourier transform

Hence, we can in *principle* invert the integral equation and reconstruct the LOSVD, $F\left(\mathrm{v}_{\mathrm{LOS}}\right)$.

In *practice*, this method is vulnerable to noise on the observed galaxy spectrum, $G(u)$, and uncertainties in the kernel $S(u)$.

Need to filter out high frequency (k) noise



Filter, denoting range of wavenumbers which give reliable inversion

Ratio of two small quantities: very noisy

Suppose we have two functions $g(t)$ and $h(t)$

Their **convolution** is defined as

$$\boxed{(g * h)(t) = \int_{-\infty}^{\infty} g(s)h(t-s)\,ds}$$

We can prove the **Convolution Theorem** $(g * h)(t) \Leftrightarrow G(f)H(f)$

i.e. the FT of the convolution of the two functions is equal to the product of their individual FTs.

Known as the ***lag***

Also their **correlation**, which is also a function of t, is defined as

$$\boxed{\mathrm{Corr}(g,h) = \int_{-\infty}^{\infty} g(s+t)h(s)\,ds}$$

We can prove the **Correlation Theorem** $\mathrm{Corr}(g,h) \Leftrightarrow G(f)H^*(f)$

i.e. the FT of the first time domain function, multiplied by the complex conjugate of the FT of the second time domain function, is equal to the FT of their correlation.

The correlation of a function with itself is called the **auto-correlation**

In this case $\mathrm{Corr}(g,g) \Leftrightarrow |G(f)|^2$

The function $|G(f)|^2$ is known as the **power spectral density**, or (more loosely) as the **power spectrum**.

Hence, the power spectrum is equal to the Fourier Transform of the auto-correlation function for the time domain function $g(t)$

The power spectral density

The power spectral density is analogous to the pdf we defined in previous sections.

In order to know how much power is contained in a given interval of frequency, we need to integrate the power spectral density over that interval.

The **total power** in a signal is the same, regardless of whether we measure it in the time domain or the frequency domain:

$$\boxed{\text{Total Power} \equiv \int_{-\infty}^{\infty} |h(t)|^2 \, dt = \int_{-\infty}^{\infty} |H(f)|^2 \, df}$$

Parseval's Theorem

Often we will want to know how much power is contained in a frequency interval without distinguishing between positive and negative values.

In this case we define the **one-sided power spectral density**:

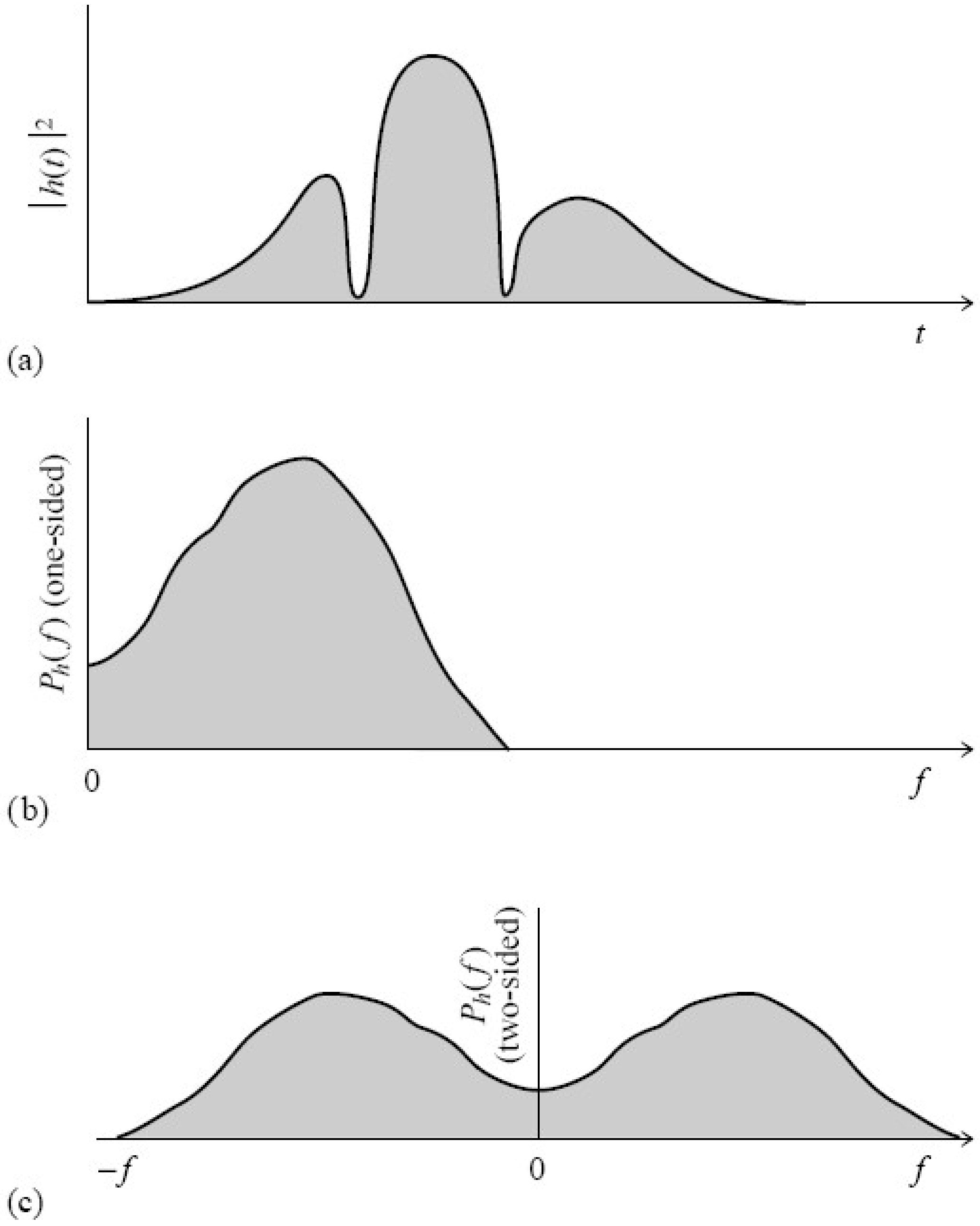
$$P_h(f) \equiv |H(f)|^2 + |H(-f)|^2 \qquad 0 \leq f < \infty$$

And $$\text{Total Power} \equiv \int_0^\infty P_h(f)\,df$$

When $h(t)$ is a real function $P_h(f) \equiv 2|H(f)|^2$

With the proper normalisation, the total power (i.e. the integrated area under the relevant curve) is the same regardless of whether we are working with the time domain signal, the power spectral density or the one-sided power spectral density.

From Numerical Recipes, Chapter 12.0



Time domain

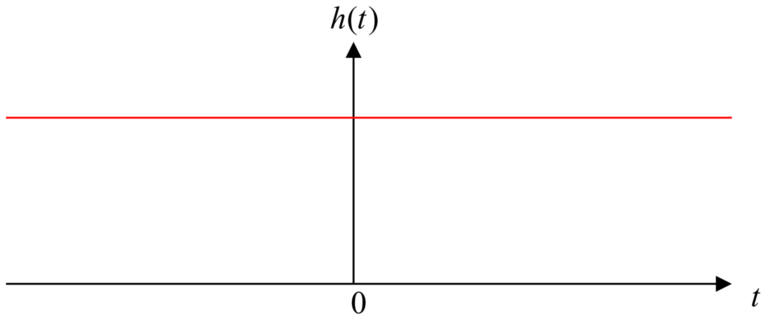
One-sided PSD

Two-sided PSD

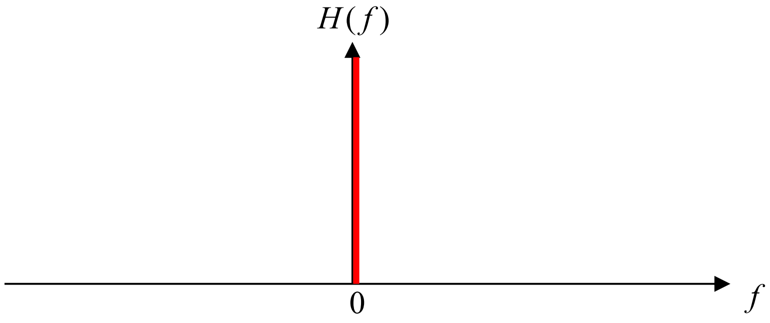
Examples

(1) $h(t) = \text{const.}$ $\qquad$ $H(f) = \delta_{\mathrm{D}}(0)$

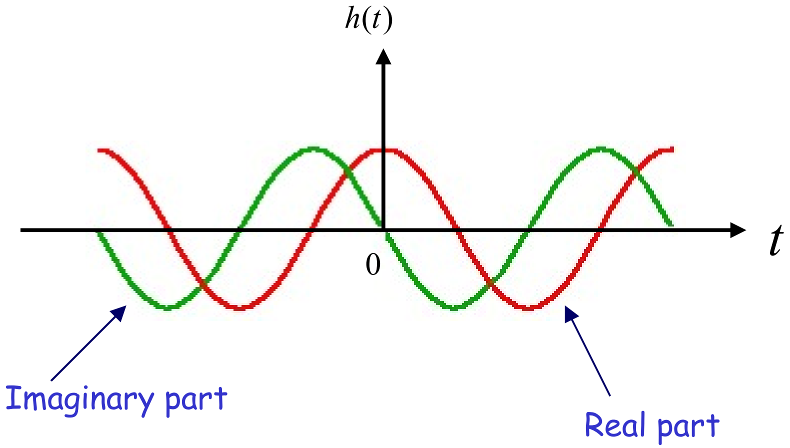
Dirac Delta function



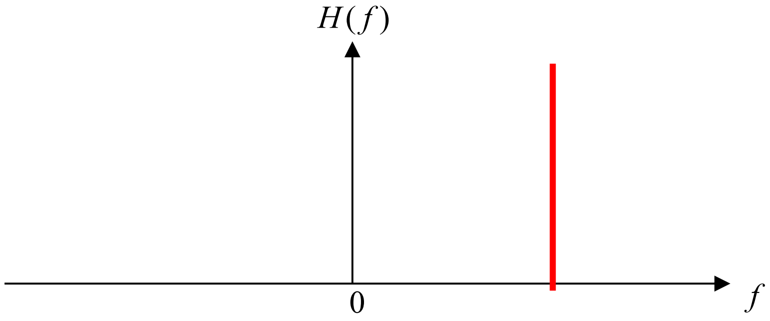
$\Leftrightarrow$



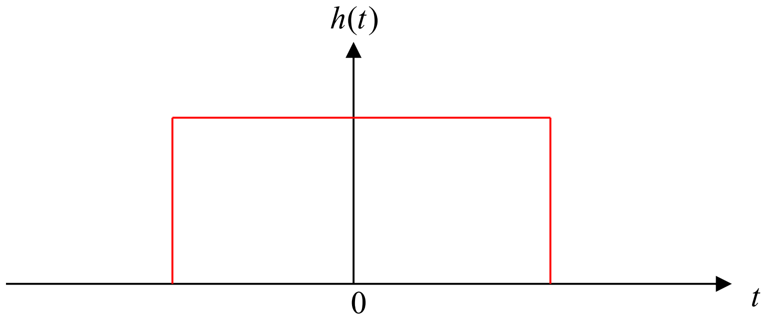
(2) $h(t) = e^{2\pi i f_0 t}$ $\qquad$ $H(f) = \delta_{\mathrm{D}}(f - f_0)$



$\Leftrightarrow$

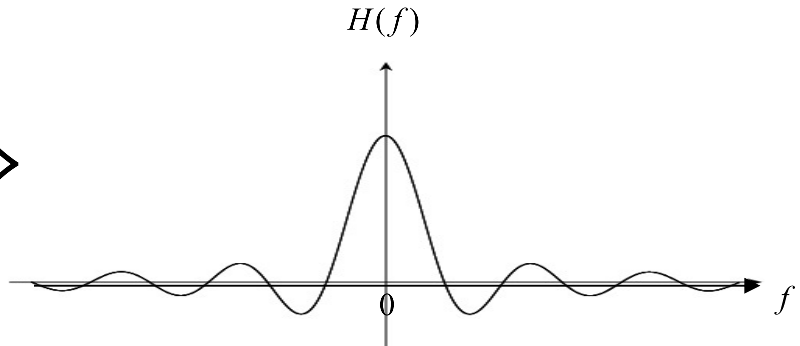


(3) $$h(t)=\begin{cases}1 & \text{for } -\frac{1}{2}\le t\le \frac{1}{2}\\ 0 & \text{otherwise}\end{cases}$$



$$\Leftrightarrow$$

$$H(f)=\text{sinc}(\pi f)$$

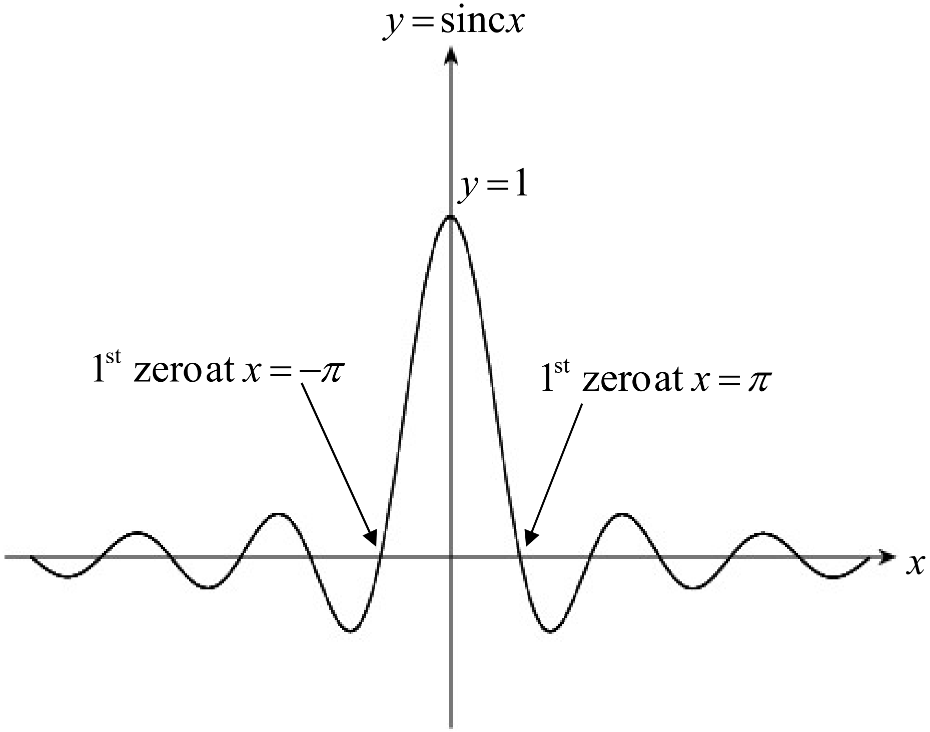


Imaginary part = 0

The sinc function occurs frequently in many areas of physics

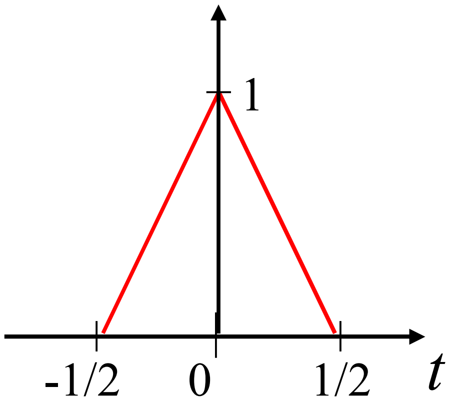
$$\text{sinc}\, x \equiv \frac{\sin x}{x}$$

The function has a maximum at $x=0$ and the zeros occur at $x=\pm m\pi$ for positive integer m



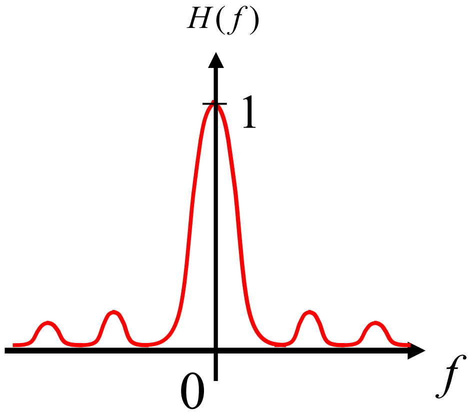
(4)

$$h(t) = \Delta(t)$$



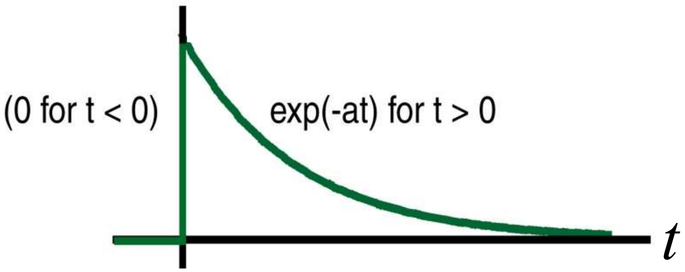
$$\Leftrightarrow$$

$$H(f) = \mathrm{sinc}^2(\pi f)$$



(5)

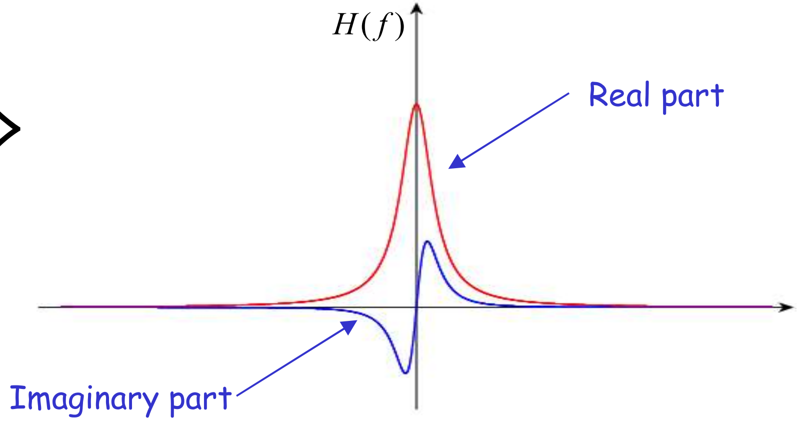
$$h(t) = e^{-at}$$



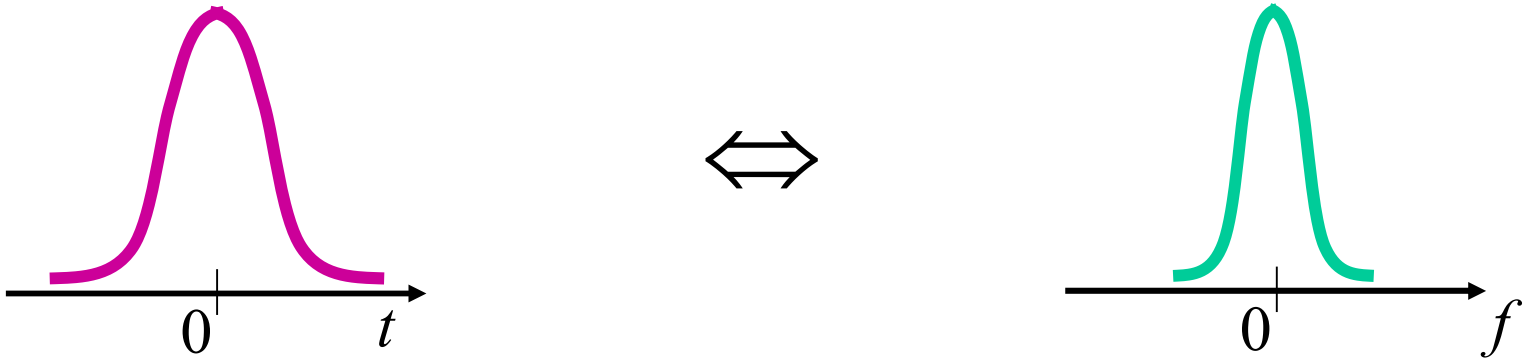
$$\Leftrightarrow$$

$$\mathrm{Re}\left[H(f)\right] = \frac{a^2}{a^2 + 4\pi^2 f^2}$$

$$\mathrm{Im}\left[H(f)\right] = -\frac{2a^2\pi f}{\left(a^2 + 4\pi^2 f^2\right)^2}$$



(6) $h(t) = \exp\left(-a^2t^2\right)$ $\qquad$ $H(f) \propto \exp\left(-\pi^2 f^2 / a^2\right)$



i.e. the FT of a Gaussian function in the time domain is *also* a Gaussian in the frequency domain.

Question 20: If the variance of a Gaussian is doubled in the time domain

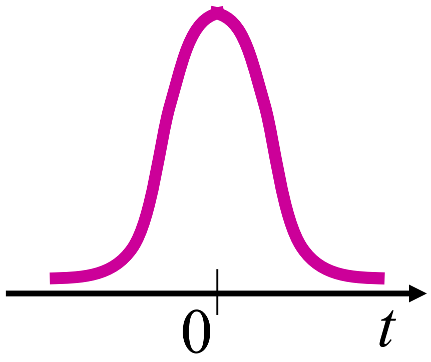
A the variance of its Fourier transform will be doubled in the frequency domain

B the variance of its Fourier transform will be halved in the frequency domain

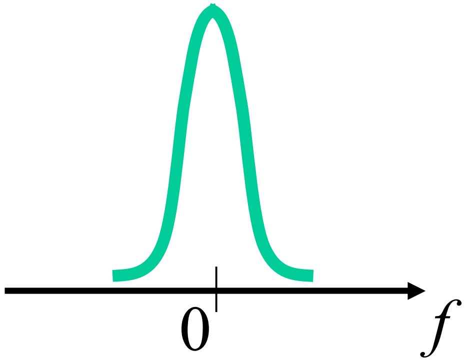
C the standard deviation of its Fourier transform will be doubled in the frequency domain

D the standard deviation of its Fourier transform will be halved in the frequency domain

(6) $h(t) = \exp\left(-a^2 t^2\right)$ $H(f) \propto \exp\left(-\pi^2 f^2 / a^2\right)$



$\Leftrightarrow$



i.e. the FT of a Gaussian function in the time domain is *also* a Gaussian in the frequency domain.

The broader the Gaussian is in the time domain, then the narrower the Gaussian FT in the frequency domain, and vice versa.

Discrete Fourier Transforms

Although we have discussed FTs so far in the context of a continuous, analytic function, $h(t)$, in many practical situations we must work instead with observational data which are sampled at a discrete set of times.

Suppose that we sample $h(t)$ in total $N+1$ times at evenly spaced time intervals Δ, i.e. (for N even)

$$h_k \equiv h(t_k) \quad \text{where} \quad t_k = k\Delta, \qquad k = -N/2,\ldots,0,\ldots,N/2$$

[If $h(t)$ is non-zero over only a finite interval of time, then we suppose that the $N+1$ sampled points contain this interval. Or if $h(t)$ has an infinite range, then we at least suppose that the sampled points cover a sufficient range to be representative of the behaviour of $h(t)$].

We therefore approximate the FT as

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt \approx \sum_{k=-N/2}^{k=N/2} h_k e^{2\pi i f t_k} \Delta$$

Since we are sampling $h(t)$ at $N+1$ discrete timesteps, in view of the symmetry of the FT and inverse FT it makes sense also to compute $H(f)$ only at a set of $N+1$ discrete frequencies:

$$f_n \equiv \frac{n}{N\Delta}, \qquad n = -N/2, \ldots, 0, \ldots, N/2$$

(The frequency $f_c = 1/2\Delta$ is known as the **Nyquist (critical) frequency** and it is a very important value. We discuss its significance later).

Discrete Fourier Transform of the h_k

Then

$$H(f_n) \;\approx\; \sum_{k=-N/2}^{k=N/2} h_k\, e^{2\pi i f_n t_k}\,\Delta \;=\; \Delta \sum_{k=-N/2}^{k=N/2} h_k\, e^{2\pi i k n/N}$$

Note that $e^{-\pi i n} = e^{\pi i n}$

Hence, there are only N independent values.

Also, note that $e^{2\pi i k n/N} = e^{2\pi i n + 2\pi i k n/N} = e^{2\pi i n(N+k)/N}$

So we can re-define the Discrete FT as:

$$H(f_n) \approx \Delta \sum_{k=0}^{N-1} h_k\, e^{2\pi i k n/N} = \Delta H_n$$

The discrete *inverse* FT, which recovers the set of h_k's from the set of H_n's is

$$h_k = \frac{1}{N}\sum_{k=0}^{N-1} H_n e^{-2\pi i k n / N}$$

Parseval's theorem for discrete FTs takes the form

$$\sum_{k=0}^{N-1} \left|h_k\right|^2 \quad = \quad \frac{1}{N}\sum_{n=0}^{N-1} \left|H_n\right|^2$$

There are also discrete analogues to the convolution and correlation theorems.

Fast Fourier Transforms

Consider again the formula for the discrete FT. We can write it as

$$H_n = \sum_{k=0}^{N-1} e^{2\pi i k n / N} h_k \equiv \sum_{k=0}^{N-1} W^{nk} h_k$$

This is a **matrix equation**: we compute the $(N \times 1)$ vector of H_n's by multiplying the $(N \times N)$ matrix $\left[W^{nk}\right]$ by the $(N \times 1)$ vector of h_k's.

In general, this requires of order N^2 multiplications (and the h_k's may be complex numbers).

e.g. suppose $N = 10^8 \Rightarrow N^2 = 10^{16}$. Even if a computer can perform (say) 1 billion multiplications per second, it would still require more than **115 days** to calculate the FT.

Fortunately, there is a way around this problem.

Suppose (as we assumed before) N is an even number. Then we can write

$$H_n = \sum_{k=0}^{N-1} e^{2\pi i k n/N} h_k \quad = \quad \underbrace{\sum_{j=0}^{N/2-1} e^{2\pi i (2j) n/N} h_{2j}}_{\text{Even values of } k} + \underbrace{\sum_{j=0}^{N/2-1} e^{2\pi i (2j+1) n/N} h_{2j+1}}_{\text{Odd values of } k}$$

$$= \sum_{j=0}^{M-1} e^{2\pi i j n/M} h_{2j} + W^n \sum_{j=0}^{M-1} e^{2\pi i j n/M} h_{2j+1}$$

where $M = N/2$

So we have turned an FT with N points into the weighted sum of **two** FTs with $N/2$ points. This would reduce our computing time by a factor of two.

Why stop there, however?...

If M is *also* even, we can repeat the process and re-write the FTs of length M as the weighted sum of two FTs of length $M/2$.

$$\vdots$$

$$\vdots$$

If N is a **power of two** (e.g. 1024, 2048, 1048576 etc) then we can repeat iteratively the process of splitting each longer FT into two FTs half as long.

The final step in this iteration consists of computing FTs of length unity:

$$H_0 = \sum_{k=0} e^{2\pi i k 0} h_k \equiv h_0$$

i.e. the FT of each discretely sampled data value is just the data value itself.

This iterative process converts $O(N^2)$ multiplications into $O(N \log_2 N)$ operations.

This notation means ‘of the order of’

So our 10^{16} operations are reduced to about 2.7×10^9 operations.

The Fast Fourier Transform (FFT) has revolutionised our ability to tackle problems in Fourier analysis on a desktop PC which would otherwise be impractical, even on the largest supercomputers.

Data Acquisition

Earlier we approximated the continuous function $h(t)$ and its FT $H(f)$ by a finite set of $N+1$ discretely sampled values.

How good is this approximation? The answer depends on the form of $h(t)$ and $H(f)$. In this short section we will consider:

1. under what conditions we can reconstruct $h(t)$ and $H(f)$ ***exactly*** from a set of discretely sampled points?

2. what is the minimum **sampling rate** (or density, if h is a spatially varying function) required to achieve this exact reconstruction?

3. what is the effect on our reconstructed $h(t)$ and $H(f)$ if our data acquisition does *not* achieve this minimum sampling rate?

The Nyquist – Shannon Sampling Theorem

Suppose the function $h(t)$ is **bandwidth limited**. This means that the FT of $h(t)$ is non-zero over a finite range of frequencies.

i.e. there exists a **critical frequency** f_c such that

$$H(f) = 0 \quad \text{for all} \quad |f| \geq f_c$$

The **Nyquist – Shannon Sampling Theorem** (NSST) is a very important result from information theory. It concerns the representation of $h(t)$ by a set of discretely sampled values

$$h_k \equiv h(t_k) \quad \text{where} \quad t_k = k\Delta, \qquad k = \ldots,-2,-1,0,1,2,\ldots$$

The NSST states that, provided the sampling interval Δ satisfies

$$\boxed{\Delta = 1/2 f_c}$$ or less

then we can **exactly** reconstruct the function $h(t)$ from the discrete samples $\{h_k\}$. It can be shown that

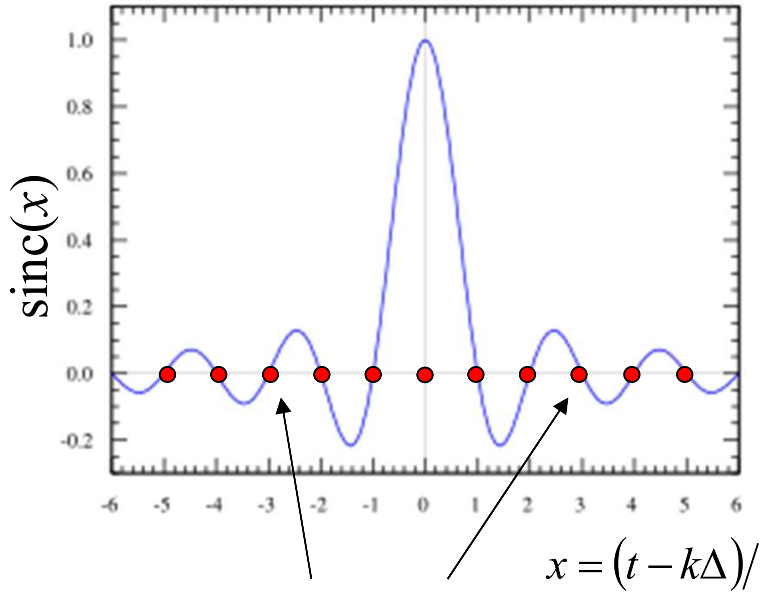
$$\boxed{h(t) = \Delta \sum_{\mathrm{k}=-\infty}^{+\infty} h_k \frac{\sin\left[2\pi f_c (t - k\Delta)\right]}{\pi (t - k\Delta)}}$$

f_c is also known as the **Nyquist frequency** and $\Delta^{-1} = 2 f_c$ is known as the **Nyquist rate**.

We can re-write this as

$$h(t) = \sum_{k=-\infty}^{+\infty} h_k \frac{\sin[\pi(t-k\Delta)/\Delta]}{\pi[(t-k\Delta)/\Delta]}$$

So the function $h(t)$ is the sum of the sampled values $\{h_k\}$, weighted by the ***normalised* sinc function**, scaled so that its zeroes lie at those sampled values.



$x = (t - k\Delta)/\Delta$

Sampled values

Normalised sinc function

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

(compare with previous section)

Note that when $t = k\Delta$ then

$h(t) = h_k$ since $\text{sinc}(0) = 1$

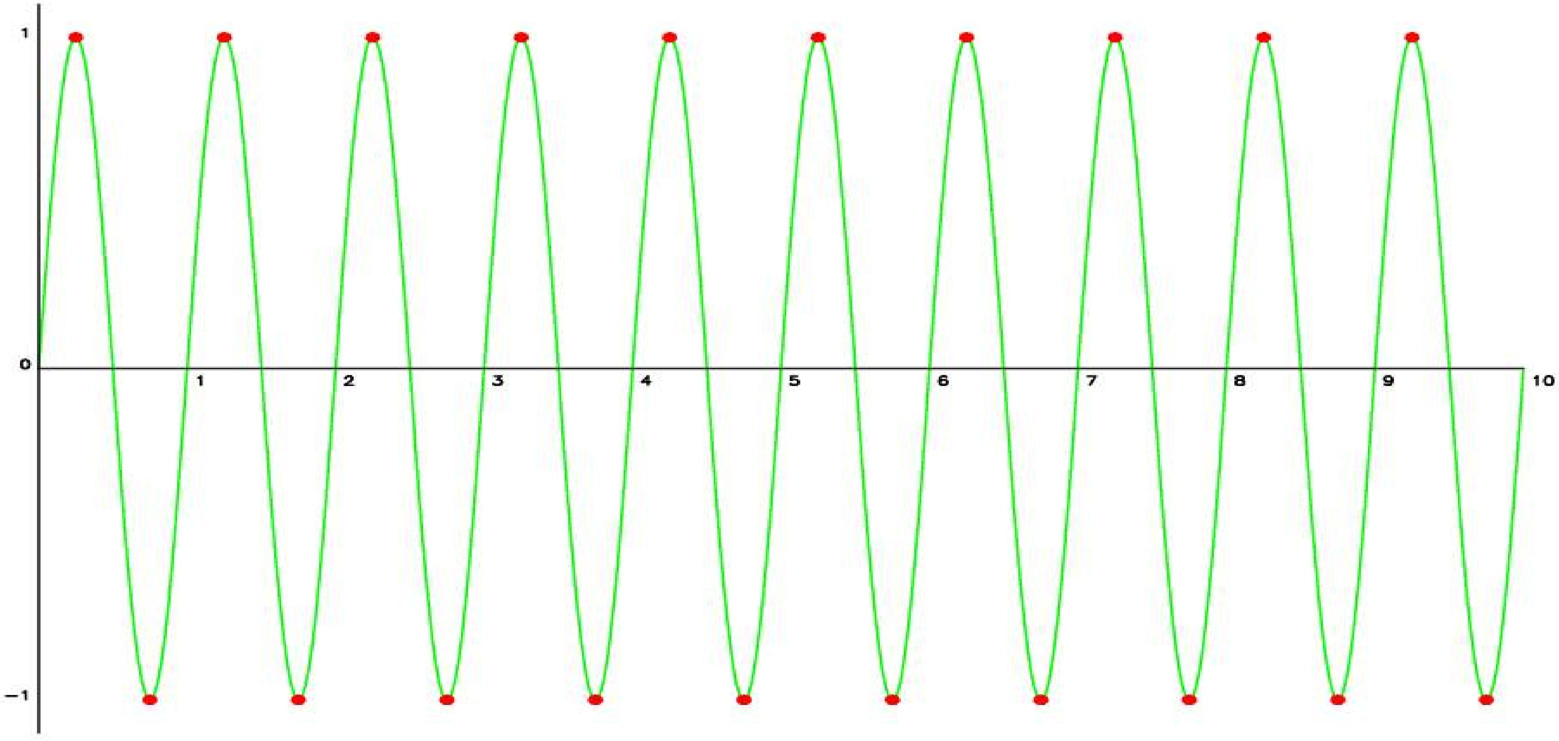
The NSST is a very powerful result.

We can think of the interpolating sinc functions, centred on each sampled point, as 'filling in the gaps' in our data. The remarkable fact is that they do this job *perfectly*, provided $h(t)$ is bandwidth limited. i.e. the discrete sampling incurs no loss of information about $h(t)$ and $H(f)$.

Suppose, for example, that $h(t) = \sin\left(2\pi f_c t\right)$. Then we need only sample $h(t)$ *twice* every period in order to be able to reconstruct the entire function exactly.

(Note that formally we do need to sample an *infinite number* of discretely spaced values, $\{h_k\}$. If we only sample the $\{h_k\}$ over a finite time interval, then our interpolated $h(t)$ will be approximate).

$y = \sin(2\pi f_c t)$



$x = f_c t$

$2\Delta = 1/f_c$

Sampling $h(t)$ at (infinitely many of) the **red** points is sufficient to reconstruct the function for all values of t, with no loss of information.

Aliasing

There is a downside, however.

If $h(t)$ is *not* bandwidth limited (or, equivalently, if we don't sample frequently enough – i.e. if the sampling rate $\Delta^{-1} < 2f_c$) then our reconstruction of $h(t)$ and $H(f)$ is badly affected by **aliasing**.

This means that all of the power spectral density which lies *outside* the range $-f_c < f < f_c$ is spuriously moved *inside* that range, so that the FT $H(f)$ of $h(t)$ will be computed **incorrectly** from the discretely sampled data.

Any frequency component outside the range $(-f_c, f_c)$ is falsely translated (**aliased**) into that range.

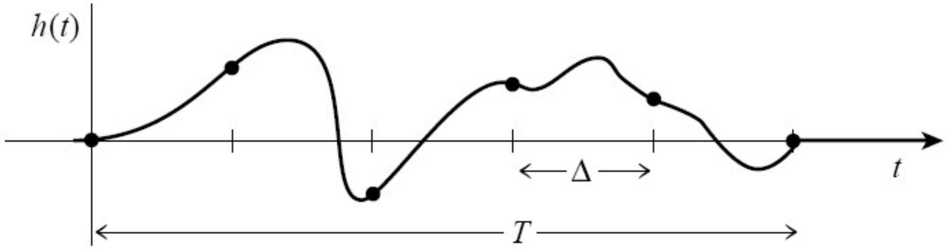
Consider $h(t)$ as shown.

Suppose $h(t)$ is zero outside the range T.

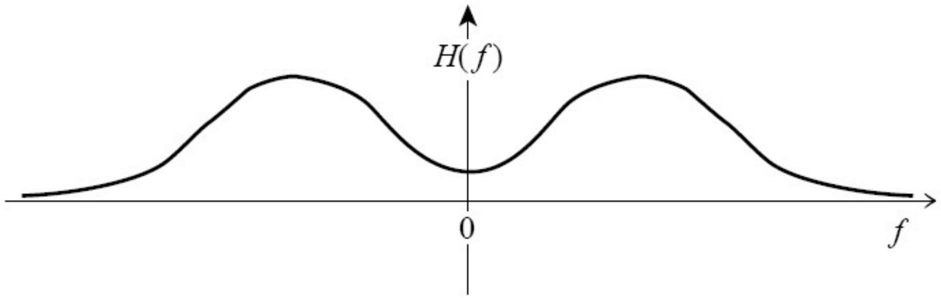
This means that $H(f)$ extends to $\pm\infty$.

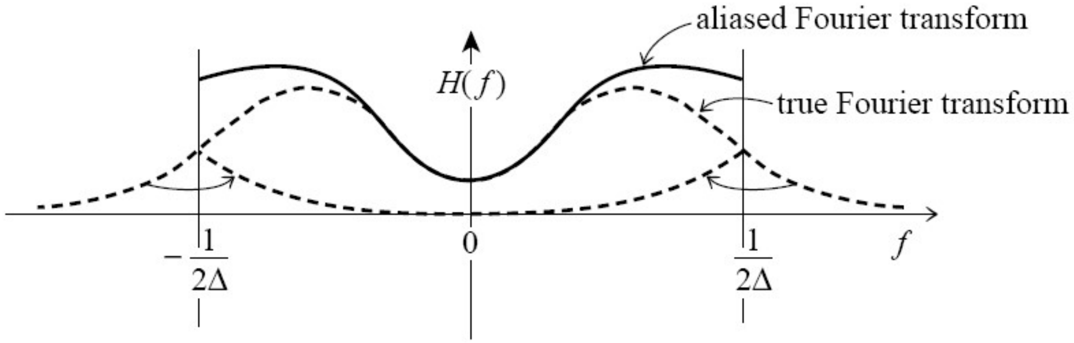
The contribution to the true FT from outside the range $(-1/2\Delta, 1/2\Delta)$ gets aliased into this range, appearing as a 'mirror image'.

Thus, at $f = \pm 1/2\Delta$ our computed value of $H(f)$ is equal to *twice* the true value.



$h(t)$ sampled at regular intervals Δ





From Numerical Recipes, Chapter 12.1

How do we combat aliasing?

- Enforce some chosen f_c e.g. by *filtering* $h(t)$ to remove the high frequency components $|f| > f_c$. *(Also known as anti-aliasing)*
- Sample $h(t)$ at a high enough rate Δ^{-1} so that $\Delta^{-1} \geq 2f_c$ - i.e. at least two samples per cycle of the highest frequency present

To check for / eliminate aliasing *without* pre-filtering:

- Given a sampling interval Δ, compute $f_{\lim} = 1/2\Delta$
- Check if discrete FT of $h(t)$ is approaching **zero** as $f \rightarrow f_{\lim}$
- If *not*, then frequencies outside the range $(-1/2\Delta, 1/2\Delta)$ are probably being folded back into this range.
- Try increasing the sampling rate, and repeat…

And finally….

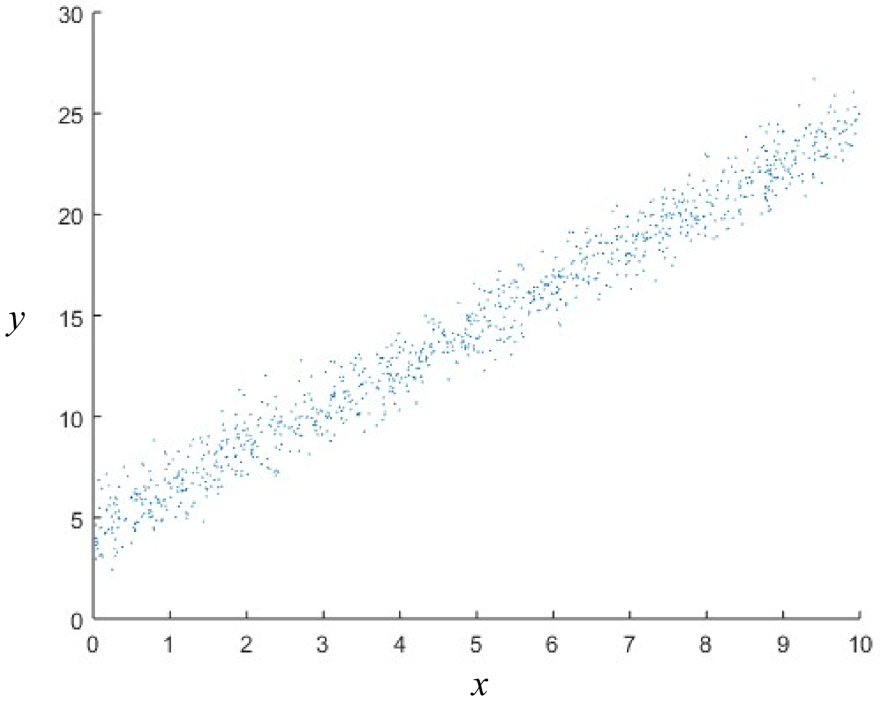
Mock Data Challenge

Part 1

1000 (x,y) data pairs, generated from an unknown model plus Gaussian noise.

Data posted on my.SUPA and on Moodle (for MSc course)

Three-stage challenge:



1. Fit a linear model to these data, using ordinary least squares;
2. Compute Bayesian credible regions for the model parameters, using a bivariate normal model for the likelihood function;
3. Write an MCMC code to sample from the posterior pdf of the model parameters, and compare their sample estimates with the LS fits;

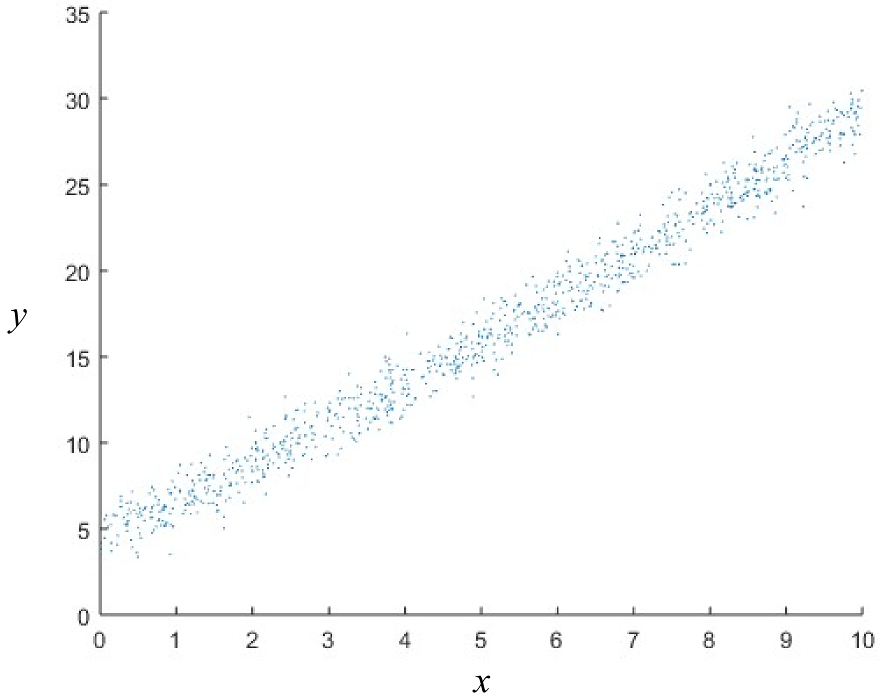
And finally….

Mock Data Challenge

Part 2

1000 (x,y) **new** data pairs, generated from an unknown model plus Gaussian noise.

Data posted on my.SUPA and on Moodle (for MSc course)



1. Fit a quadratic model to these data, using e.g. an MCMC code to sample from the posterior pdf of the model parameters, presenting plots of the marginal posterior for each pair of parameters;