Fourier Series and Transform

Overview

- Why Fourier transform?
- Trigonometric functions
- Who is Fourier?
- Fourier series
- Fourier transform
- Discrete Fourier transform
- Fast Fourier transform
- 2D Fourier transform
- Tips

Why Fourier transform?

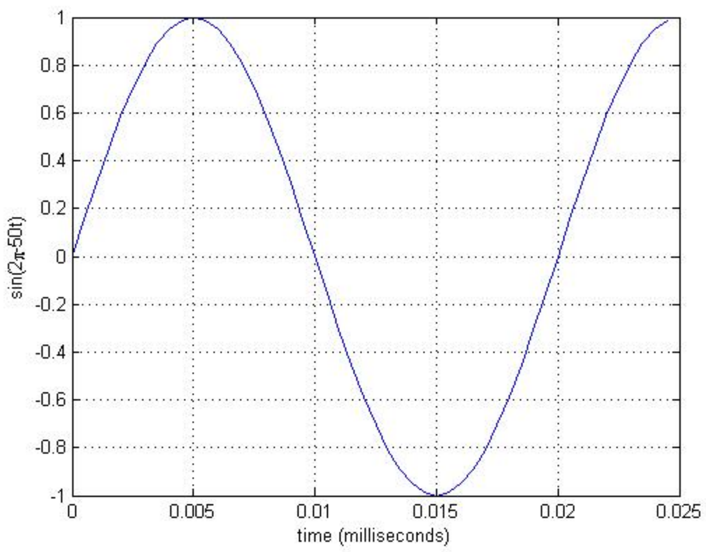
Fourier, not being noble, could not enter the artillery, although he was a second Newton.

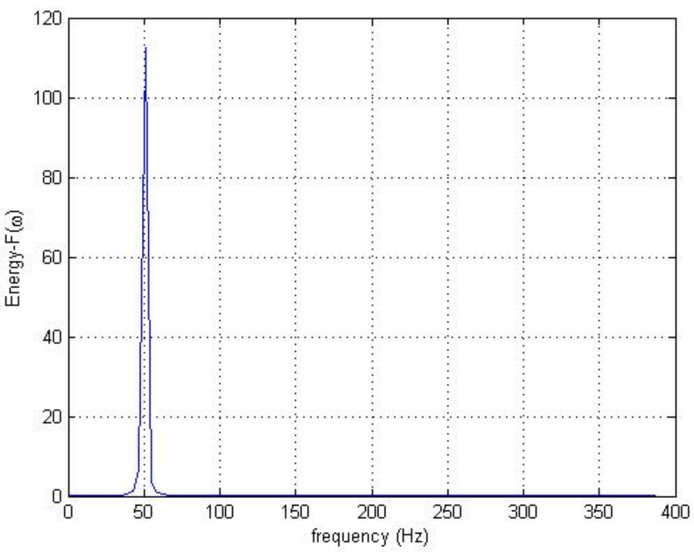
— Francois Jean Dominique Arago

- For signal processing, Fourier transform is the tool to connect the time domain and frequency domain.

- Why would we do the exchange between time domain and frequency domain? Because we can do all kinds of useful analytical tricks in the frequency domain that are just too hard to do computationally with the original time series in the time domain. Simplify the calculation.

Why Fourier transform?

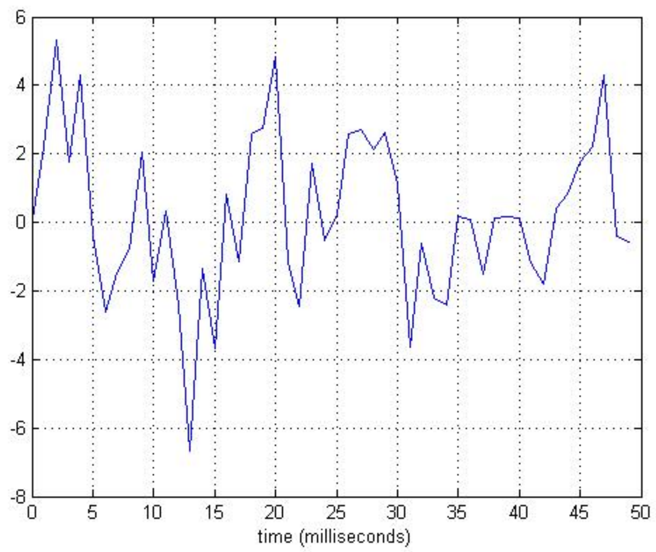


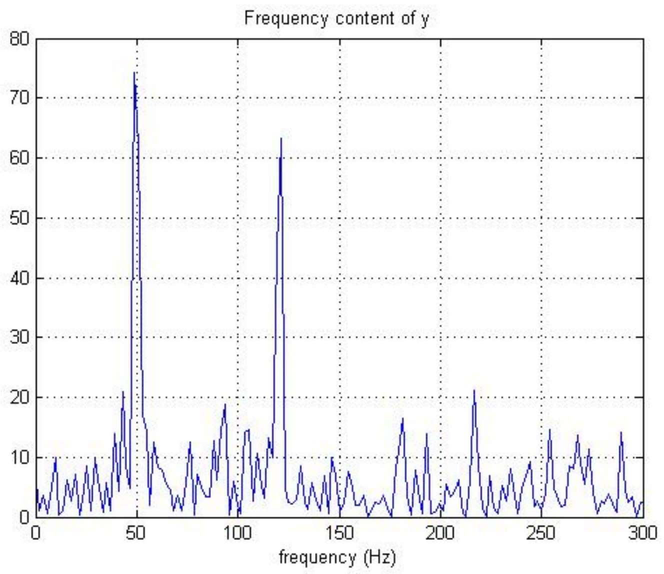


$$f(t) = \sin(2\pi * 50t)$$

Why Fourier transform?

- This example is a sound record analysis. The left picture is the sound signal changing with time. However, we have no any idea about this sound by the time record. By the Fourier transform, we know that this sound is generated at 50Hz and 120Hz mixed with other noises.





Trigonometric functions (ex.1)

- Trigonometric system is the periodic functions as:

  1, sinx, cosx, sin2x, cos2x, …, sinnx, cosnx, ….

- The properties of trigonometric system

$$\cos\theta\cos\phi = \cos(\theta-\phi)/2 + \cos(\theta+\phi)/2$$

$$\sin\theta\sin\phi = \cos(\theta-\phi)/2 - \cos(\theta+\phi)/2$$

$$(\sin\theta)' = \cos\theta, \qquad (\cos\theta)' = -\sin\theta$$

- Trigonometric system is the orthogonal system

$$\int_{-\pi}^{\pi}\cos nx\,dx = 0, \qquad \int_{-\pi}^{\pi}\sin nx\,dx = 0 \qquad \int_{-\pi}^{\pi}\sin mx\cos nx\,dx = 0$$

$$\int_{-\pi}^{\pi}\cos mx\cos nx\,dx = \begin{cases} 0 & m \neq n \\ \pi & m = n \end{cases} \qquad \int_{-\pi}^{\pi}\sin mx\sin nx\,dx = \begin{cases} 0 & m \neq n \\ \pi & m = n \end{cases}$$

In Matlab

- Int(f)
  Int(f,a,b)

  if f is a symbolic expression.

$$\int_a^b f(x)dx$$

- q=quad(fun,a,b)
  q=quad(fun,a,b,tol)
  [q,fcnt]=quad(fun,a,b,…)

Quadrature is a numerical method used to find the area under the graph of a function, that is, to compute a definite integral.

$$q = \int_a^b f(x)dx$$

q=quad(fun,a,b) tries to approximate the integral of function fun from a to b to within an error of 1e-6 using recursive adaptive Simpson quadrature. fun is a function handle for either an M-file function or an anonymous function. The function y=fun(x) should accept a vector argument x and return a vector result y, the integrand evaluated at each element of x.

Who is Fourier?

Jean Baptiste Joseph Fourier (France, 1768~1830).

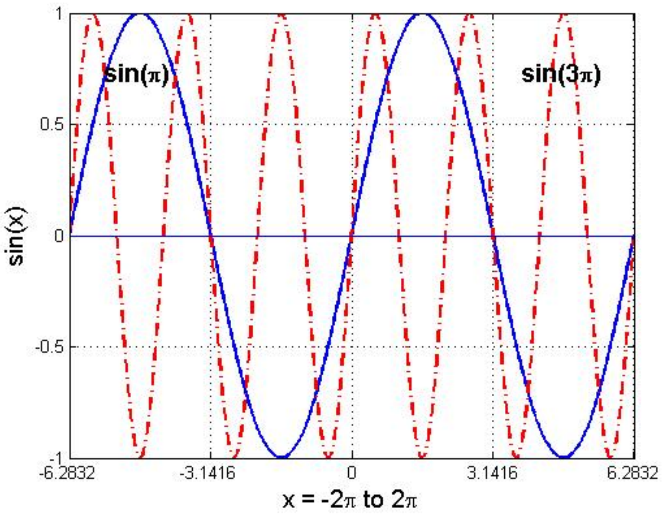
- Fourier is one of the France's greatest administrators, historians, and mathematicians.
- He graduated with honors from the military school in Auxerre and became a teach of math when he was 16 years old.
- Later he joined the faculty at Ecole Normale and then the Polytechnique in Paris when he is 27.
- He went to Egypt with Napoleon as the Governor of Lower Egypt after the 1798 Expedition.
- He was secretary of the Academy of Sciences in 1816 and Fellow in 1817.
- **Don't believe it?**

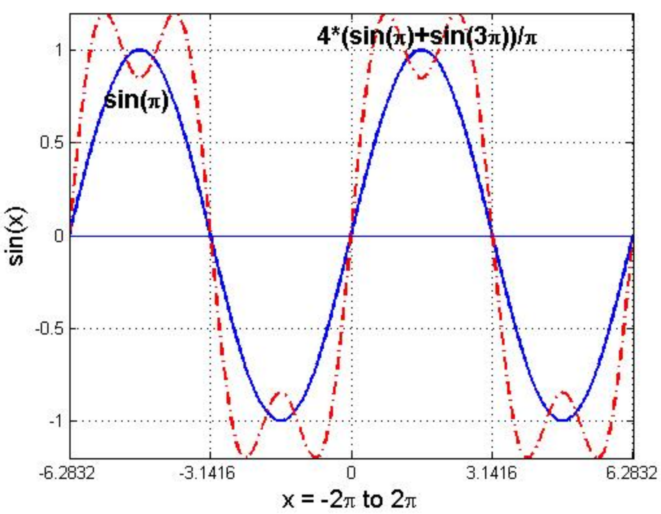
  Neither did Lagrange, Laplace, Poisson and other big wigs.

  Not translated into English until 1878!
- **But it's true!!**

Fourier's basic idea

- Trigonometric functions: sin(x) and cos(x) has the period 2π.
- sin(nx) and cos(nx) have period $2\pi/n$.
- The linear combination of these functions or multiply each by a constant, the adding result still has a period 2π.





Fourier series

- For any function f(x) with period 2π (f(x) = f(2π +x)), we can describe the f(x) in terms of an infinite sum of sines and cosines

$$f(x) = \frac{a_0}{2} + \sum_{m=1}^{\infty}(a_m \cos mx + b_m \sin mx),$$

- To find the coefficients a, b and a, we multiply above equation by cosmx or sinmx and integrate it over interval -π<x<π. By the orthogonality relations of sin and cos functions, we can get

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

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

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

Fourier series→ example (ex.2)

- Period function $f(x)=\begin{cases} 1 & 0<x<\pi \\ -1 & -\pi<x<0 \end{cases}$
- The parameters are

$$a_0=\frac{1}{\pi}\int_0^{\pi}1dx+\frac{1}{\pi}\int_{-\pi}^{0}(-1)dx=0$$

$$a_1=\frac{1}{\pi}\int_0^{\pi}\cos xdx+\frac{1}{\pi}\int_{-\pi}^{0}(-1)\cos xdx$$
$$=-\frac{1}{\pi}\sin x\Big|_0^{\pi}-\frac{1}{\pi}\sin x\Big|_{-\pi}^{0}$$
$$=0$$

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

$$b_n=\frac{4}{n\pi},\qquad n=1,3,5,\ldots$$

$$a_m=\frac{1}{\pi}\int_0^{\pi}\cos mxdx+\frac{1}{\pi}\int_{-\pi}^{0}(-1)\cos mxdx$$
$$=-\frac{1}{m\pi}\sin mx\Big|_0^{m\pi}+\frac{1}{m\pi}\sin mx\Big|_{-m\pi}^{0}$$
$$=0$$

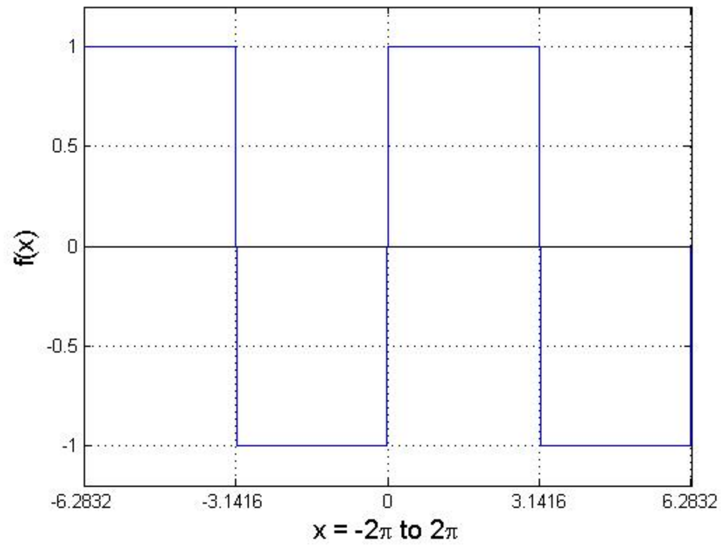
$$b_k=0,\qquad k=2,4,6\ldots$$

$$\boxed{f(x)=\frac{4}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots\right)}$$

Fourier series→ example (cont…)

- Period function $f(x)=\begin{cases} 1 & 0<x<\pi \\ -1 & -\pi<x<0 \end{cases}$
- The Fourier series is:

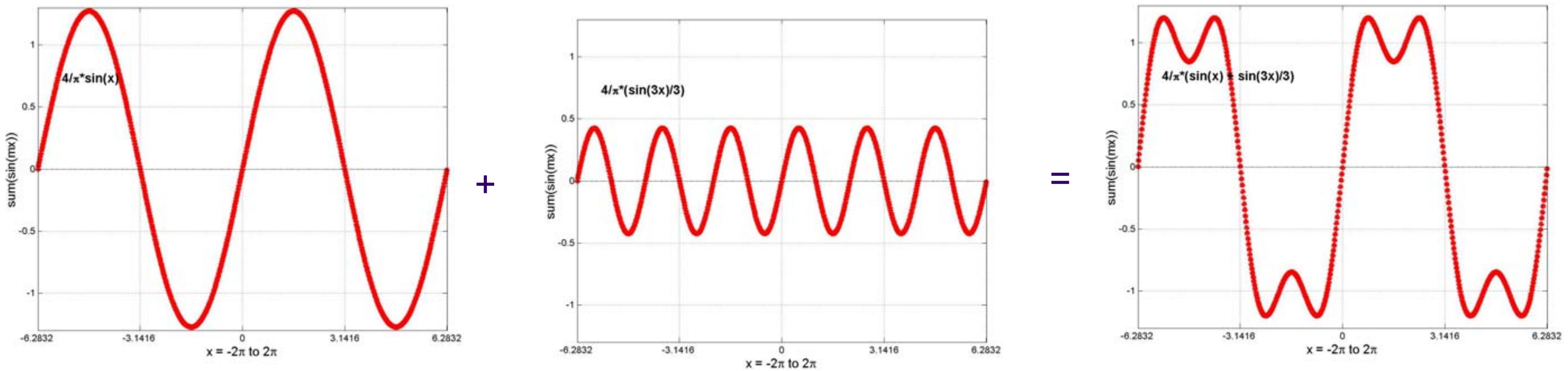
$$\boxed{f(x)=\frac{4}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots\right)}$$



Fourier series→ example (cont…)

- Period function $f(x)=\begin{cases} 1 & 0<x<\pi \\ -1 & -\pi<x<0 \end{cases}$
- The Fourier series is:

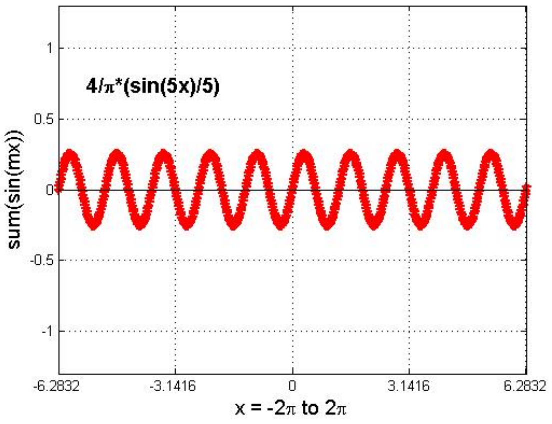
$$\boxed{f(x)=\frac{4}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots\right)}$$



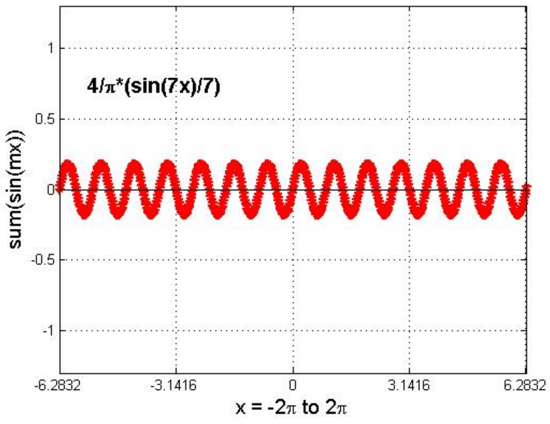
Fourier series→ example (cont…)

- Period function $f(x) = \begin{cases} 1 & 0 < x < \pi \\ -1 & -\pi < x < 0 \end{cases}$
- The Fourier series is:

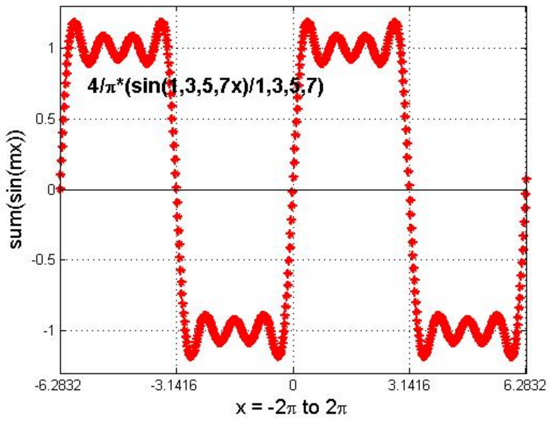
$$\boxed{f(x) = \frac{4}{\pi}\left(\frac{\sin x}{1} + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \ldots\right)}$$



+



=

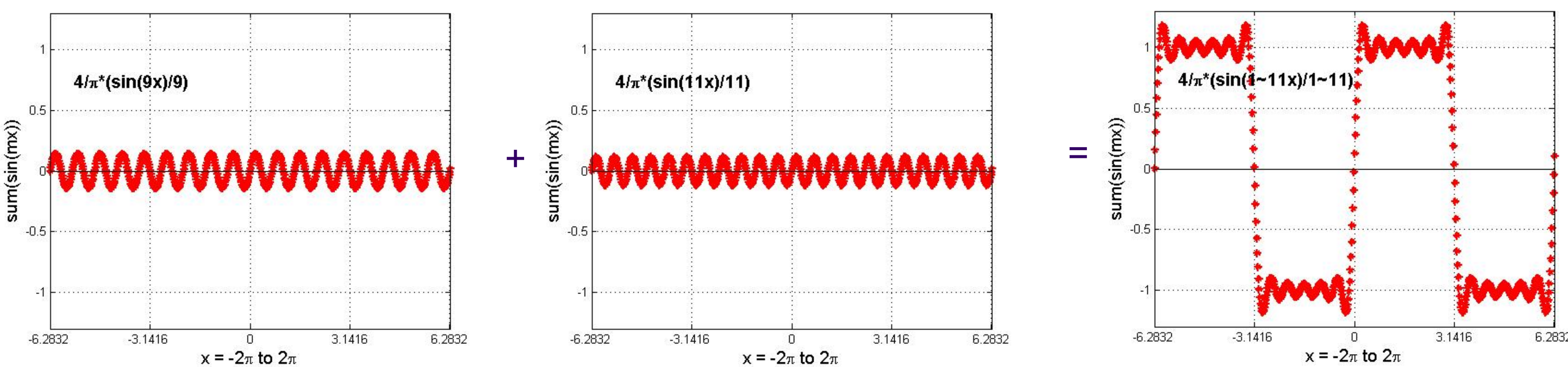


Fourier series→ example (cont…)

- Period function $f(x)=\begin{cases} 1 & 0<x<\pi \\ -1 & -\pi<x<0 \end{cases}$

- The Fourier series is:

$$\boxed{f(x)=\frac{4}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots\right)}$$

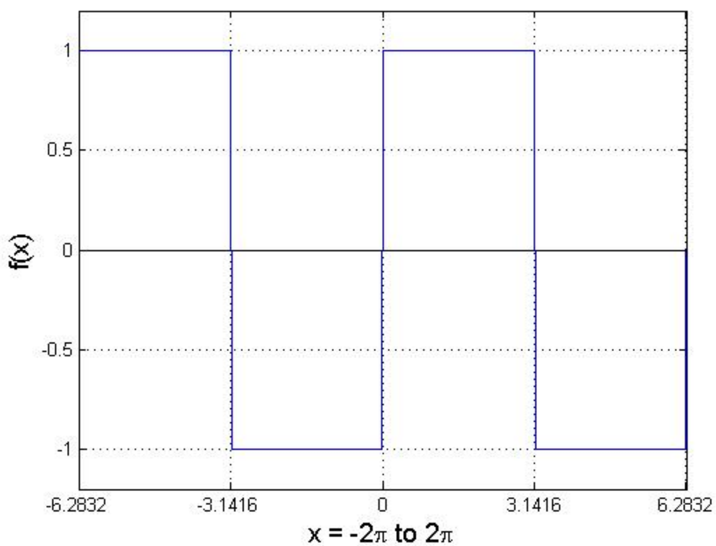


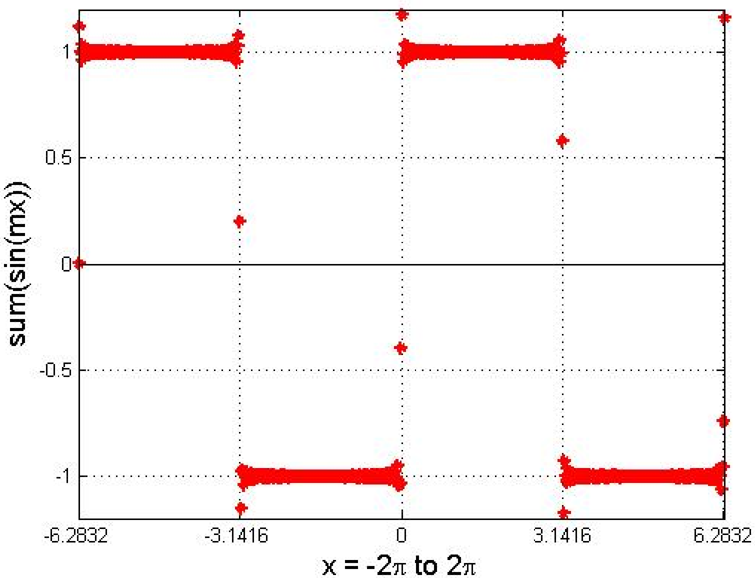
Fourier series→ example (cont…)

- Period function $f(x)=\begin{cases}1 & 0<x<\pi \\ -1 & -\pi<x<0\end{cases}$

- The Fourier series is:

$$\boxed{f(x)=\frac{4}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\ldots\right)}$$

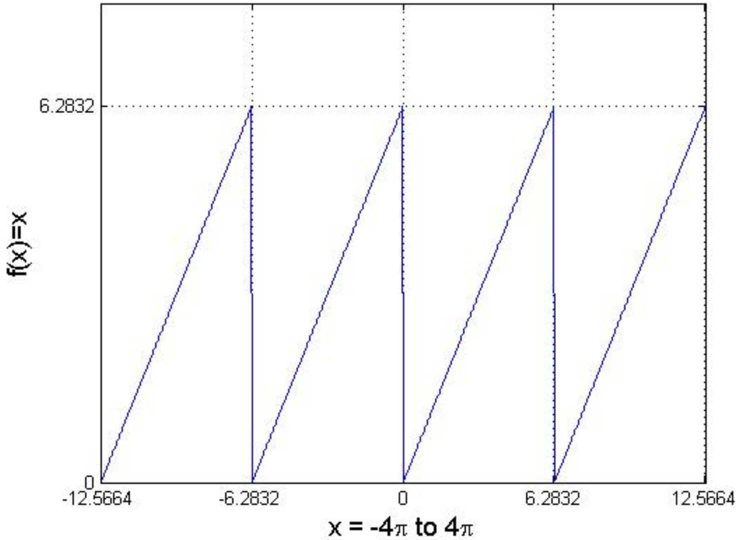


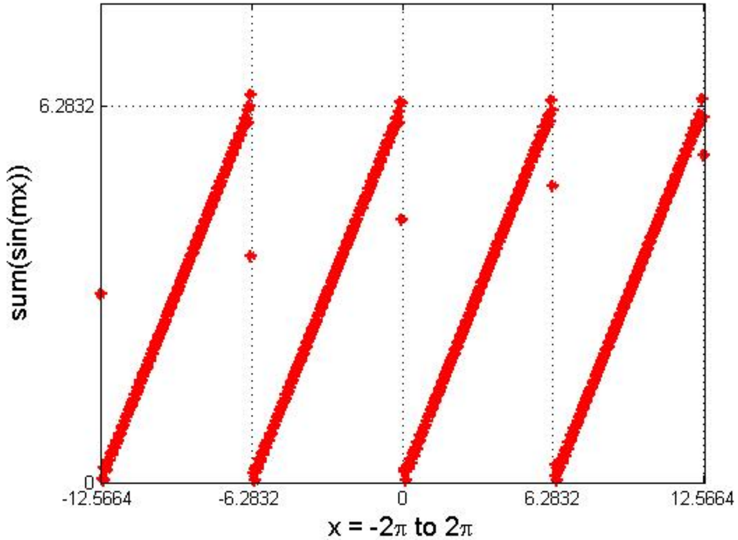


Fourier series→ example

- Period function $f(x) = x \qquad 0 < x < 2\pi$
- The parameters are

$$\boxed{f(x) = \pi - 2\left(\frac{\sin x}{1} + \frac{\sin 2x}{2} + \frac{\sin 3x}{3} + \ldots\right)}$$





Fourier series→ general

- For any function f(x') with arbitrary period T , a simple change of variables can be used to transform the interval of integration from [-π, π] to [-T/2,T/2] as

$$x = \frac{2\pi}{T} x', \qquad dx = \frac{2\pi}{T} dx'$$

- The f(x') can be described by the Fourier series as

$$f(x') = \frac{a_0}{2} + \sum_{m=1}^{\infty} (a_m \cos(m \frac{2\pi}{T} x') + b_m \sin(m \frac{2\pi}{P} x')), \qquad m = 1,2,\ldots$$

- where

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(x') dx'$$

$$a_m = \frac{2}{T} \int_{-T/2}^{T/2} f(x') \cos(m \frac{2\pi}{T} x') dx'$$

$$b_m = \frac{2}{T} \int_{-T/2}^{T/2} f(x') \sin(m \frac{2\pi}{T} x') dx'$$

- Replace ω→2π/T and x'→t,

$$f(t) = \frac{a_0}{2} + \sum_{m=1}^{\infty} (a_m \cos m\omega t + b_m \sin m\omega t), \qquad m = 1,2,\ldots$$

The complex form of Fourier series

- Euler formulae

$$\left.\begin{aligned} e^{ix} &= \cos x + i\sin x \\ e^{-ix} &= \cos x - i\sin x \end{aligned}\right\} \Rightarrow \quad \begin{aligned} \cos x &= (e^{ix}+e^{-ix})/2 \\ \sin x &= (e^{ix}-e^{-ix})/2i \end{aligned}$$

- Fourier series

$$f(t) = \frac{a_0}{2} + \sum_{m=1}^{\infty}\left(\frac{a_m}{2}(e^{im\omega t}+e^{-im\omega t}) + \frac{b_m}{2i}(e^{im\omega t}-e^{-im\omega t})\right)$$

- For certain m=k,

$$a_k \cos k\omega t + b_k \sin k\omega t = a_k \frac{e^{ik\omega t}+e^{-ik\omega t}}{2} + b_k \frac{e^{ik\omega t}-e^{-ik\omega t}}{2i}$$

$$= \frac{a_k - ib_k}{2} e^{ik\omega t} + \frac{a_k + ib_k}{2} e^{-ik\omega t}$$

- Denoting that as

$$c_0 = \frac{a_0}{2}, c_k = \frac{a_k - ib_k}{2}, c_{-k} = \frac{a_k + ib_k}{2}$$

- The complex form of Fourier series

$$f(t) = c_0 + \sum_{k=1}^{\infty}(c_k e^{ik\omega t} + c_{-k} e^{-ik\omega t}) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega t}$$

$$c_k = \frac{2}{T}\int_{-T/2}^{T/2} f(t) e^{-ik\omega t} dt$$

Fourier transform

- For any non-periodic function and assume T→∞, rewrite previous general Fourier series equation and get:

$$f(t)=\sum_{k=-\infty}^{\infty}\left(\frac{2}{T}\int_{-T/2}^{T/2} f(t)e^{-ik\omega t}dt\right)e^{ik\omega t}$$

$$\xrightarrow{T=2\pi/\omega} \quad \frac{1}{\pi}\sum_{k=-\infty}^{\infty}\omega\int_{-T/2}^{T/2} f(\xi)e^{ik\omega(t-\xi)}d\xi$$

$$\to \quad \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}d\omega\int_{-\infty}^{\infty} f(\xi)e^{-i\omega\xi}d\xi$$

- Define

$$\boxed{\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \\ f(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \end{aligned}}$$

- Here, F(ω) is called as the *Fourier Transform* of f(t). Equation of f(t) is called the *inverse Fourier Transform*.

Fourier transform→ Parseval's law

- The time signal squared $f^2(t)$ represents how the energy contained in the signal distributes over time t, while its spectrum squared $F^2(\omega)$ represents how the energy distributes over frequency (therefore the term power density spectrum). Obviously, the same amount of energy is contained in either time or frequency domain, as indicated by Parseval's formula:

$$\int_{-\infty}^{\infty} \left| f(t) \right|^2 dt = \int_{-\infty}^{\infty} \left| F(\omega) \right|^2 d\omega$$

Fourier transform → properties

- The properties of Fourier transfrom

  - *Linearity property*: given f(x), g(x),

$$FT(af(x)+bg(x)) = aF(\omega)+bG(\omega)$$

  - Similarity property: g(x)=f(ax)

$$G(\omega) = \frac{1}{a}F(\frac{\omega}{a})$$

  -*Shift formula*: given g(x) = f(x+b)

$$G(\omega) = e^{i\omega b}F(\omega)$$

  - *Derivative formula*:

$$FT(f'(x)) = i\omega F(\omega)$$

In Matlab

- F = fourier(f)

  This is the Fourier transform of the symbolic scalar f with default independent variable x. The default return is a function of ω. This represents

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

- f = ifourier(F)

  This is the inverse Fourier transform of the symbolic scalar F with default independent variable ω. The default return is a function of x. This represents

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega$$

Discrete Fourier Transform (DFT)

- Discrete Fourier Transform can be understood as a numerical approximation to the Fourier transform.

- This is used in the case where both the time and the frequency variables are discrete (which they are if digital computers are being used to perform the analysis).

- To convert the integral Fourier Transform (FT) into the Discrete Fourier Transform (DFT), we can do following steps:

1) Assume the sampling window is T. The number of sampling points is N. Define the *sample interval* $\Delta T = Ts = T/N$.

2) Define the *sample points* $t_k = k(\Delta T)$ for $k = 0, \ldots, (N-1)$.

3) Define the signal values at each sampling points as $f_k = f(t_k)$.

4) Define the frequency sampling points $\omega_n = 2\pi n/T$, where $2\pi n/T$ is termed as the *fundamental frequency*.

Discrete Fourier Transform (DFT)

5) Consider the problem of approximating the FT of f at the points $\omega_n=2\pi n/T$, The answer is

$$F(\omega_n) = \int_{-\infty}^{\infty} e^{-i\omega_n t} f(t)dt, \qquad n = 0,\dots,(N-1)$$

6) Approximate this integral by Riemann sum approximation using the points t_k since f~0 for t>T.

$$\boxed{F(\omega_n) = \sum_{k=0}^{N-1} f(t_k) e^{-i\omega_n t_k}, \qquad n = 0,1,2,\dots, N-1}$$

- This is the Discrete Fourier Transform.
- The inverse Discrete Fourier Transform is defined as

$$f(t_k) = \frac{1}{N}\sum_{n=0}^{N-1} F(\omega_n) e^{i\omega_n t_k}, \qquad k = 0,1,2,\dots, N-1$$

Fast Fourier Transform (FFT)

- Fast Fourier Transform (FFT) is a effective algorithm of Discrete Fourier Transform (DFT) and developed by Cooley and Tukey at 1965.

- This algorithm reduces the computation time of DFT for N points from N^2 to Nlog2(N) (This algorithm is called Butterfly algorithm.).

- The only requirement of this algorithm is that number of point in the series have to be a power of 2 (2^n points) such as 32, 1024, 4096.

- Zero padding at the end of the data set if the sampling number is not equal to the exact the power of 2.

In Matlab

- Y = fft(X)

  This command returns the discrete Fourier transform (DFT) of X, computed with a fast Fourier transform (FFT) algorithm.

- Y = fft(X,n)

  This command returns the n-point DFT of X. If the length of X is less than n, X is padded with trailing zeros to length n. If the length of X is greater than n, the sequence X is truncated.

- Y = ifft(X)

  This command returns the inverse discrete Fourier transform (DFT) of X, computed with a fast Fourier transform (FFT) algorithm.

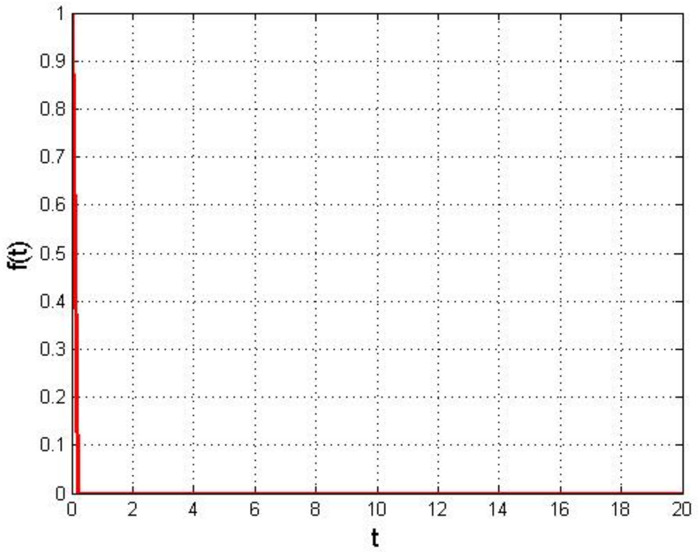
- Y = ifft(X,n)

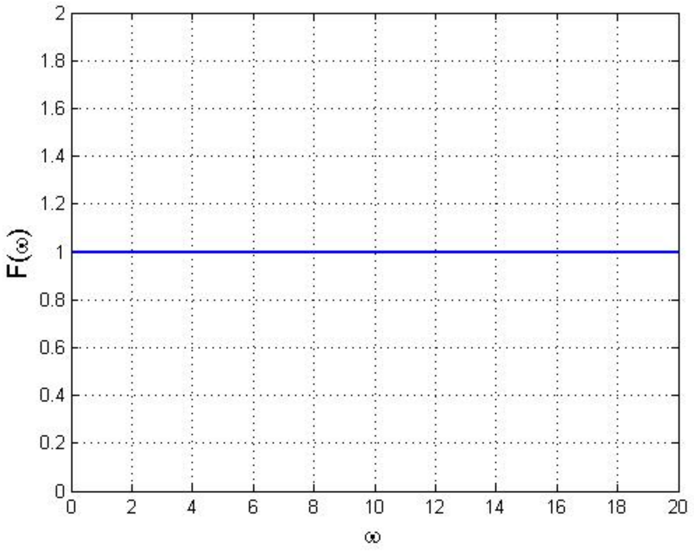
  This command returns the n-point inverse DFT of X.

Fourier transform → Delta function

- Delta function

$$f(t) = \delta(t) = \begin{cases} 1 & t = 0 \\ 0 & others \end{cases}$$

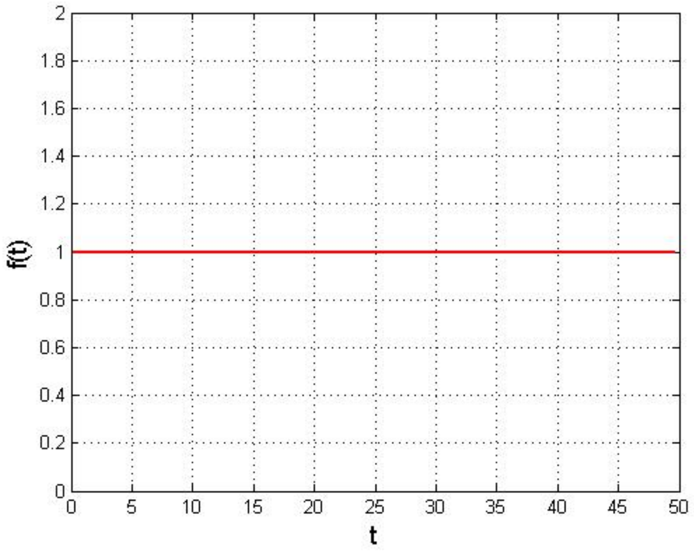


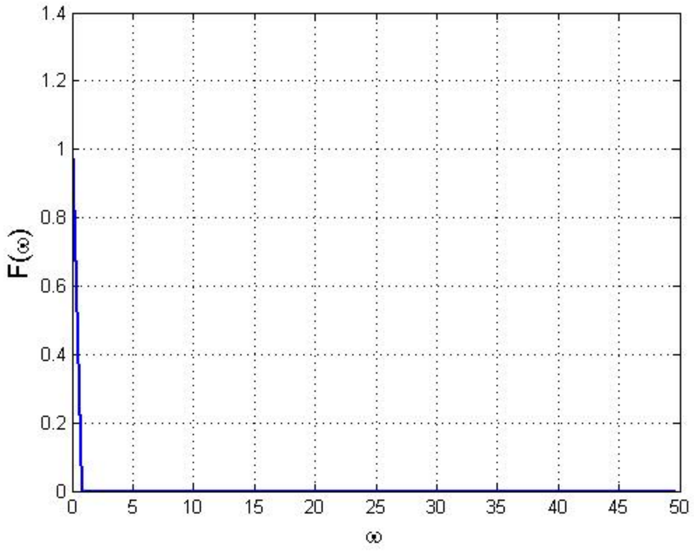


Fourier transform → Uniform function

- Unit function

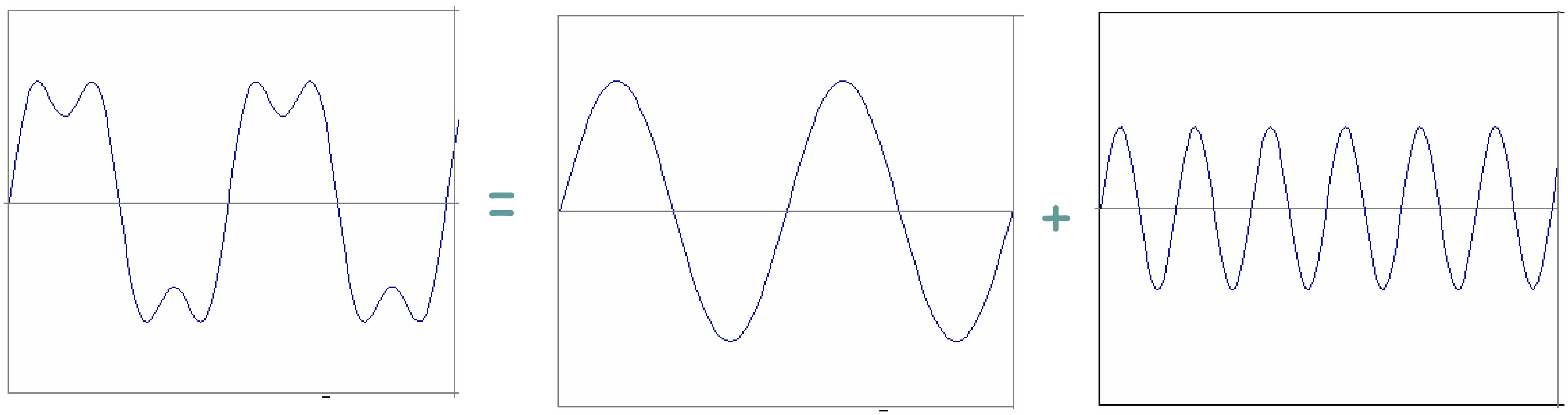
$$f(t) = 1$$





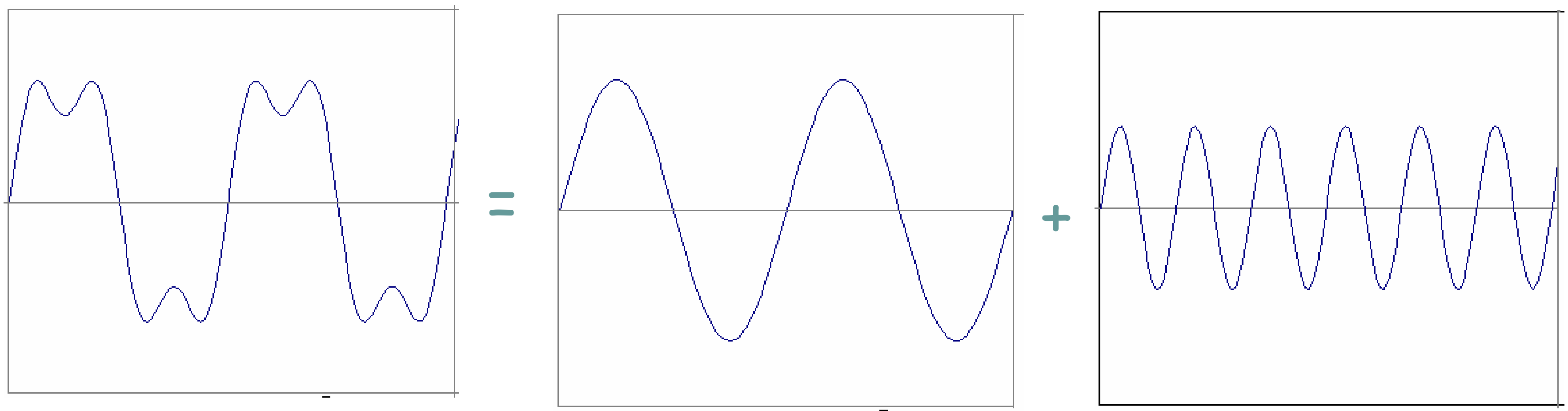
Fourier transform → Sin function

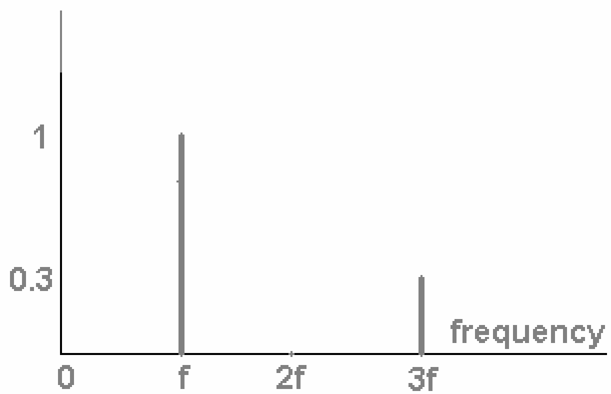
- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi 3 f\, t)$



Fourier transform → Sin function

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi\, 3f\, t)$

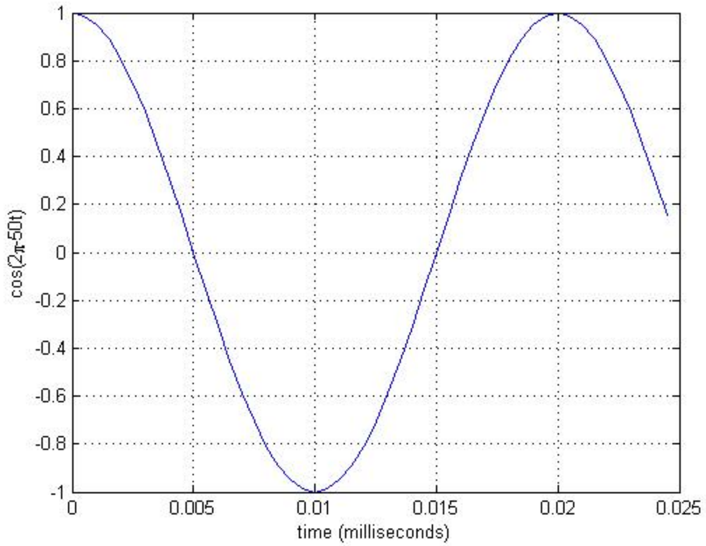


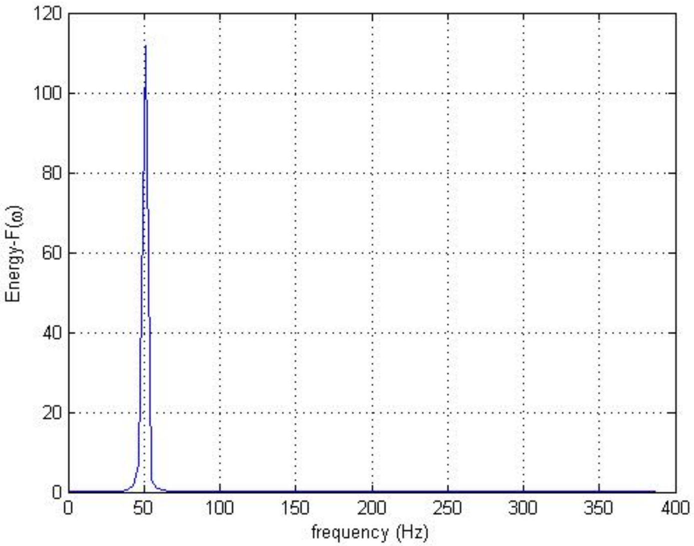


Fourier transform → Cos function

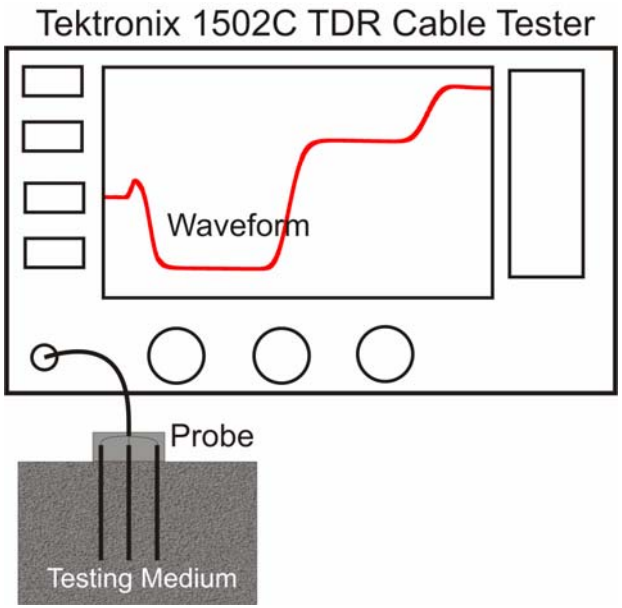
- cos functions

$$f(t) = \cos(2\pi * 50t)$$



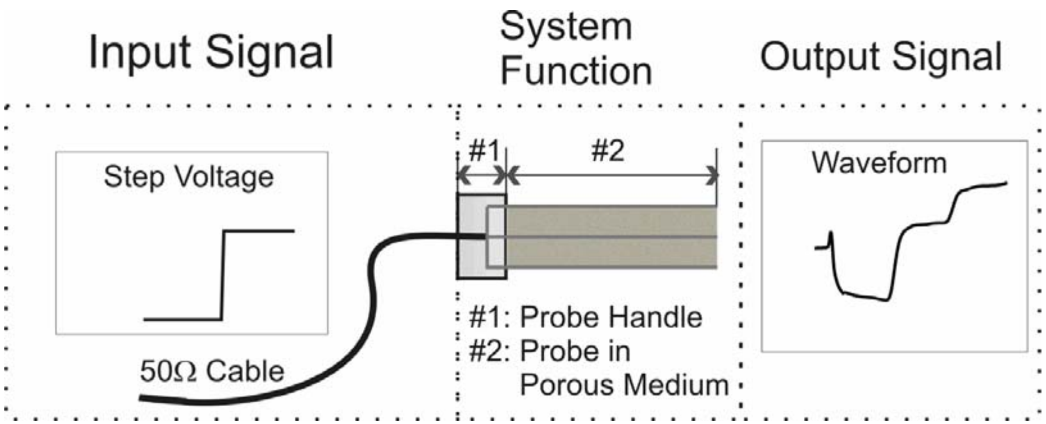


Applications → convolution



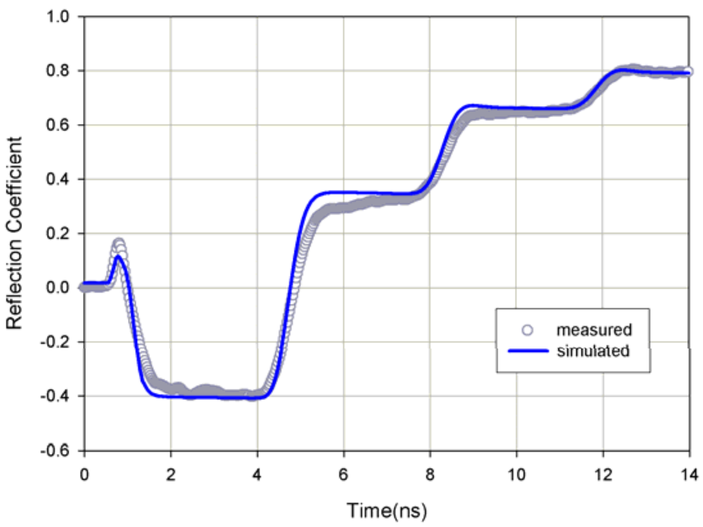
- The time domain recorded waveform is a convolution product:
- $$r(t) = \int_{-\infty}^{\infty} v_0(t-\tau)s(\tau)d\tau$$
- Simplify the complex convolution product into the direct multiply in the frequency domain by Fourier transform.

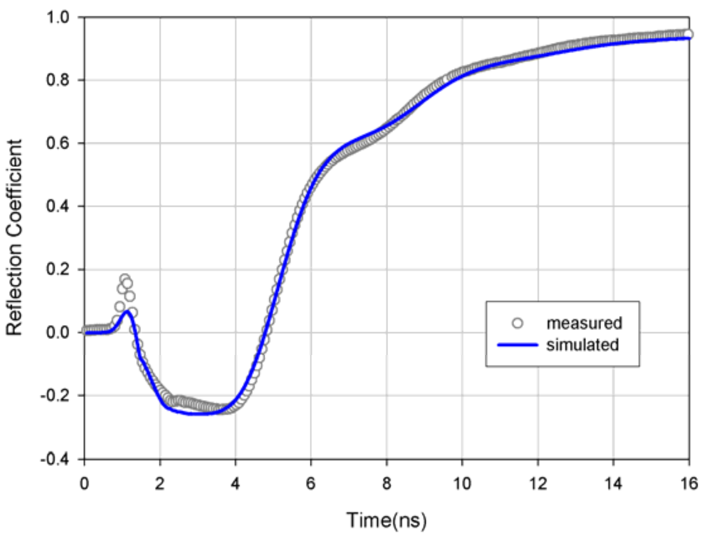
$$\overset{FFT}{\Rightarrow} R(\omega) = V_0(\omega)S(\omega)$$



Applications → convolution

- The simulated DI water waveform and ethanol waveform at room temperature by FFT.





2D Integral Fourier transform

- 2D integral Fourier transform is

$$\boxed{F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i(ux+vy)}dxdy,}$$

- Inverse 2D Fourier transform is

$$f(x,y)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{i(ux+vy)}dudv\,,$$

2D Discrete Fourier transform

- For 2D function f(x,y), DFT is

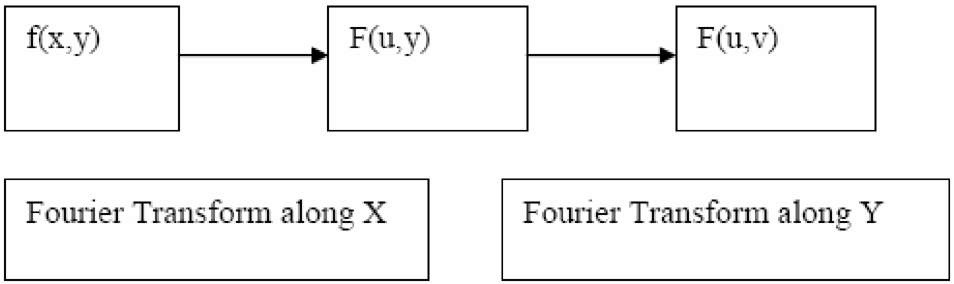
$$\boxed{F(u_m, v_n) = \sum_{xk=0}^{M-1} \sum_{yk=0}^{N-1} f(x_{xk}, y_{yk}) e^{-i(u_m x_{xk} + v_n y_{yk})}, \qquad \begin{matrix} m = 0,1,2,\ldots, M-1 \\ n = 0,1,2,\ldots, N-1 \end{matrix}}$$

- Rewrite above equation

$$F(u_m, v_n) = \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} f(x_{xk}, y_{yk}) e^{-iv_n y_{yk}} \right] e^{-iu_m x_{xk}}$$

$$F(u_m, v_n) = \sum_{m=0}^{M-1} F(x, v) e^{-iu_m x_{xk}}$$

- We can implement 2D Fourier transform as a sequence of 1-D Fourier transform operations:



2D Inverse Discrete Fourier transform

- For 2D function f(x,y), inverse DFT is

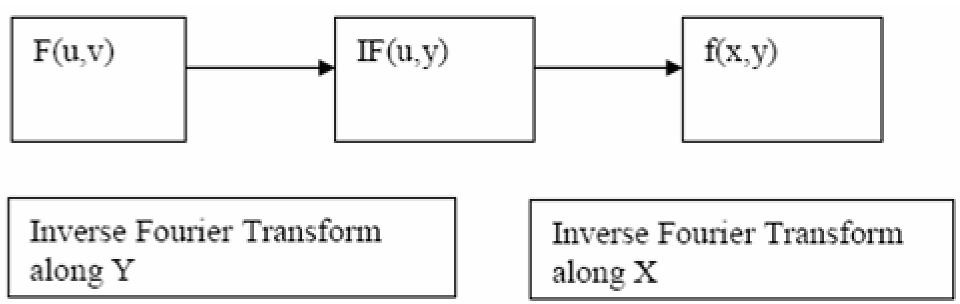
$$\boxed{f(x_{xk}, y_{yk}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(u_m, v_n) e^{-i(u_m x_{xk} + v_n y_{yk})}, \qquad \begin{aligned} xk &= 0,1,2,\ldots, M-1 \\ yk &= 0,1,2,\ldots, N-1 \end{aligned}}$$

- Rewrite above equation

$$f(x_{xk}, y_{yk}) = \frac{1}{MN} \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} F(u_m, v_n) e^{i v_n y_{yk}} \right] e^{i u_m x_{xk}}$$

$$f(x_{xk}, y_{yk}) = \frac{1}{MN} \sum_{m=0}^{M-1} F(x_{xk}, v_n) e^{-i u_m x_{xk}}$$

- We can implement 2D inverse Fourier transform as a sequence of 1-D inverse Fourier transform operations:



2D Fourier transform → properties

- The properties of Fourier transfrom

  - *Linearity property*: given f(x,y), g(x,y),

$$FT(af(x,y)+bg(x,y)) = aF(u,v)+bG(u,v)$$

  - *Shift formula*: given g(x,y) = f(x+a,y+b)

$$G(u,v) = e^{i(ua+vb)}F(u,v)$$

  - Similarity property: g(x,y)=f(ax,by)

$$G(u,v) = \frac{1}{|ab|}F(\frac{u}{a},\frac{v}{b})$$

  - Convolution

$$g(x,y)*h(x,y) = G(u,v)\cdot H(u,v)$$

In Matlab

- Y = fft2(X)

  This command returns the discrete Fourier transform (DFT) of X(x,y), computed with a fast Fourier transform (FFT) algorithm.

- Y = fft2(X,m,n)

  This command returns the n-point DFT of X(x,y) with the certain length of x at m and y at n.

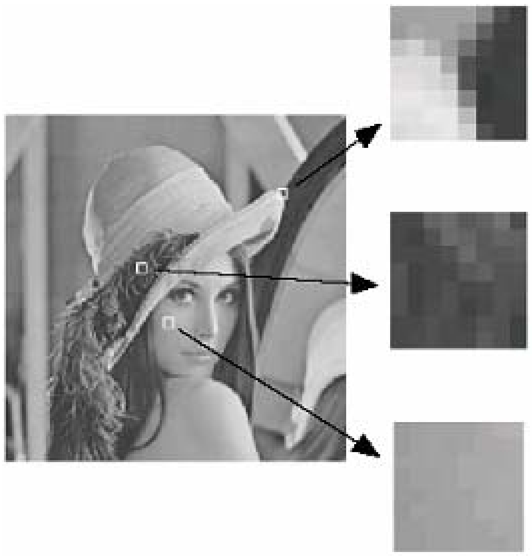
- Y = ifft2(X)

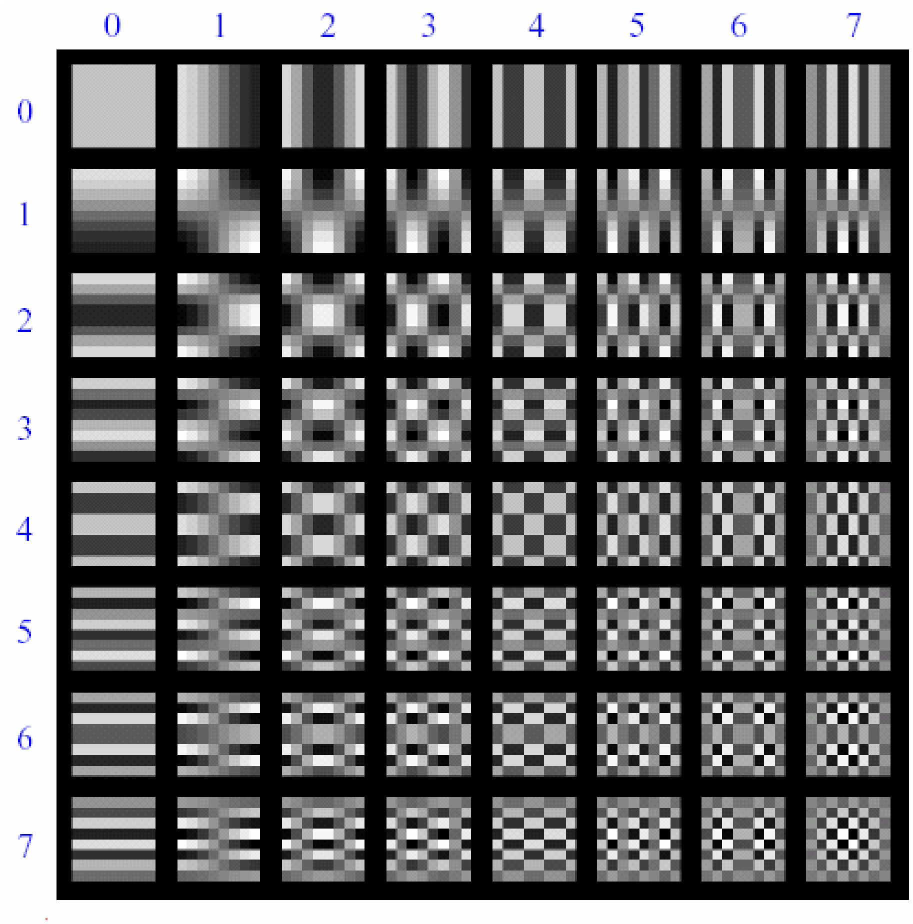
  This command returns the inverse discrete Fourier transform (DFT) of X, computed with a fast Fourier transform (FFT) algorithm.

- Y = ifft2(X,m,n)

  This command returns the m-point of x and n-point of y inverse DFT of X(x,y).

Image Compression (JPEG)





JPEG compression comparison

89k

12k

Tips → Nyquist frequency

- Nyquist frequency is called the highest frequency that can be coded at a given sampling rate in order to be able to fully reconstruct the signal.

$$f_{NF} = \frac{1}{2\Delta t}$$

- The total sampling period is T. Then, the base frequency is 2/T. This represent the lowest frequency of the signal we can see in the frequency domain.

- On the other hand, Nyquist frequency represents the highest frequency of the signal we can see in the frequency domain.

Tips → Sampling rate/sampling total time

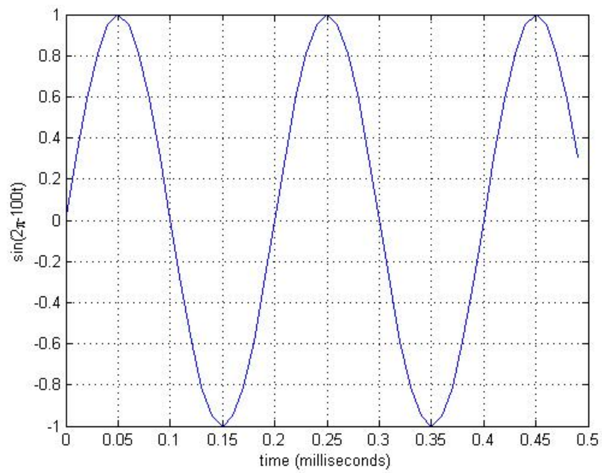
- Sampling rate is the sampling interval. This will control the highest frequency band.

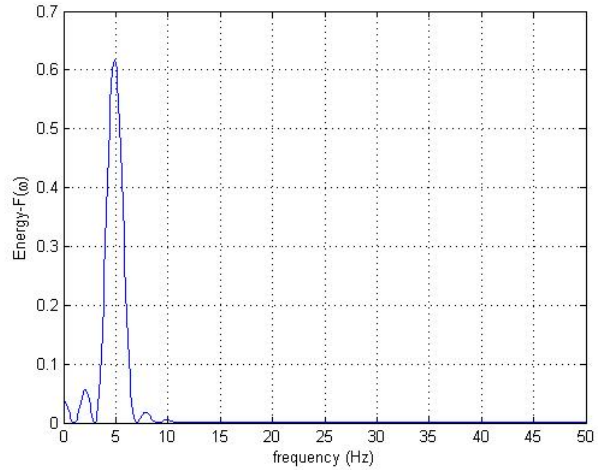
- Sampling total time is the total time period we looked to sampling the data. This will control the lowest frequency band.

- Please notice that we can extend the sampling total time by padding zero for FFT. This will change the lower frequency **resolution**. However, it **can not change the highest frequency**.

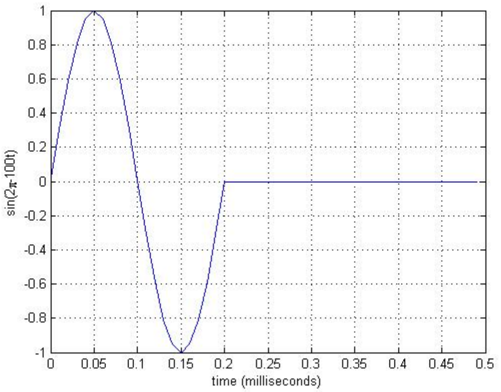
Tips → windows

$$f(t) = \sin(2\pi * 5t)$$

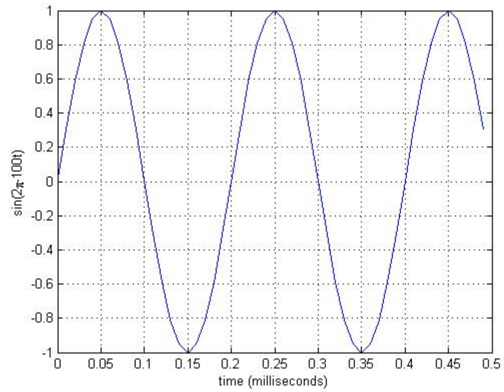




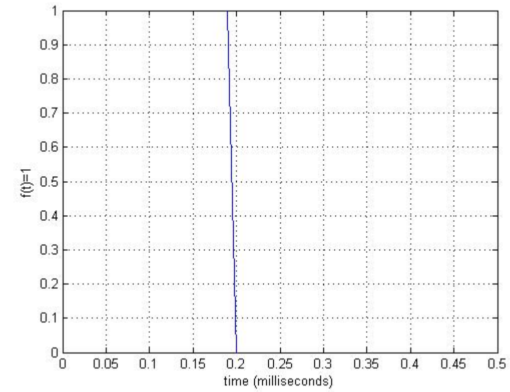
- Any sampling data range is limited/finite.



=

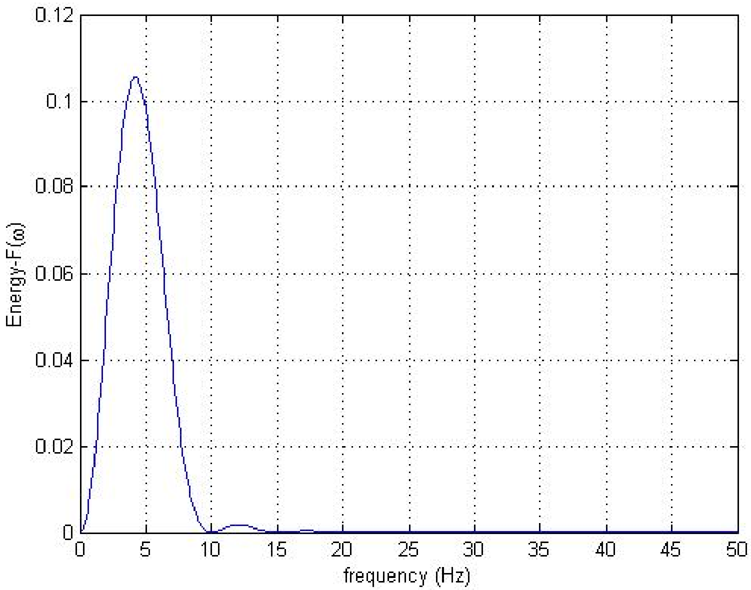


+



Tips → windows

$$f(t) = \sin(2\pi * 5t)$$



- The true sampling signal is f'(t) = f(t)·win(t).
- After the Fourier transform, the transformed the signal is the convolution products of F(ω) and WIN(ω).

$$F'(\omega) = F(\omega) * WIN(\omega)$$

- For the true transformed the signal, we have to de-convolution of the transformed the results.

The Homework of Fourier transform is using Matlab:

Do the Fourier transform of one simple harmonic function

f1(x)=sin(2*pi*500t);

and

f2(x) = cos(2*pi*500t)+2*sin(2*pi*1000t)+0.5*cos(2*pi*200t);

Please practice with:

1) **choosing two different sampling rate (how many points you sampled in total in time domain N);**
2) **choosing two different sampling interval (delta_t);**
3) **choosing two different window length to get the sense of how these different sampling will influence on your Fourier transform results in frequency domain.**