Week 4 – Fourier series and analysis

Periodic motion is by far the most common type of motion in the universe.

Three basic categories:

1. Rotations: nuclei, electrons in atoms, molecules,…, earth, planets.
2. Vibrations: nuclei, molecules,…, heart, tides, sun.

The fundamental equation that governs all periodic motion is that of a simple harmonic oscillator (SHO)

\[ \ddot{x} = -\omega^2 x \]

With solution: \[ x(t) = A \cos \omega t + B \sin \omega t \]
Most periodic motions, however, are more complicated than SHO because they involve many coupled oscillators. For n-coupled oscillators, there are n-normal modes with frequencies

$$\omega_i, i = 1, n$$

The general solution is obtained by superposing all the normal modes

$$x_k(t) = \sum_{i=1}^{n} A_{ki} \cos \omega_i t + B_{ki} \sin \omega_i t$$

This is a considerably more complicated function than the SHO.

To get a feeling for a particular solution, we can use Matlab to synthesize it and plot the result. Conversely, given some observed motion, we can perform a Fourier analysis to determine whether it is periodic and find the dominant frequencies in its spectrum.
A simple example of coupled oscillators: n beads on a string

Consider the motion of bead 2:

Force due to tension

\[ F_2 = -T \left( \frac{x_2 - x_1}{l} \right) + T \left( \frac{x_3 - x_2}{l} \right) \]

Equation of motion

\[ m \ddot{x}_2 = -\frac{T}{l} (-x_1 + 2x_2 - x_3) \]

\[ \dot{x}_2 = -\omega_0^2 (-x_1 + 2x_2 - x_3), \quad \omega_0^2 = \frac{T}{ml} \]
The same equation of motion applies to all the beads with the provision that

\[ x_0 = x_{n+1} = 0 \]

To find the normal modes, assume \[ \dot{x}_k = -\omega^2 x_k \quad \text{for all } k \]

\[
\omega_0^2 \begin{pmatrix}
2 & -1 & 0 & & & \\
-1 & 2 & -1 & & & \\
0 & -1 & 2 & \ddots & & \\
& & & \ddots & -1 & 0 & \\
& & & & -1 & 2 & -1 & \\
& & & & 0 & -1 & 2
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_{n-1} \\
x_n
\end{pmatrix}
= \omega^2
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_{n-1} \\
x_n
\end{pmatrix}
\]
Solution by inspection:
Consider the equation of motion for the $k$th bead

$$\omega_0^2 (-x_{k-1} + 2x_k - x_{k+1}) = \omega^2 x_k$$

Rewrite it in the form

$$\frac{x_{k-1} + x_{k+1}}{x_k} = \frac{2\omega_0^2 - \omega^2}{\omega_0^2}$$

This equation is satisfied for all $k$ provided $x_0 = x_{n+1} = 0$

The only functions that satisfy such a relationship are exp, sin and cos.

$$\frac{e^{(k+1)\theta} + e^{(k-1)\theta}}{e^{k\theta}} = e^\theta + e^{-\theta}$$

Exp and cos does not vanish at $k=0$, so it must be sin
(sinh is also possible but it does not vanish at $k=n+1$)
Thus the amplitude of $x_k$ is proportional to $\sin(k\theta)$.

To find $\theta$, we use the second boundary condition:

$$\sin((n + 1)\theta) = 0 \quad \rightarrow \quad (n + 1)\theta = m\pi, \quad m = 1, 2, \ldots, n$$

So the amplitudes in the $m$'th normal mode are given by

$$A_{mk} = C_m \sin \frac{mk\pi}{n + 1}$$

To find the eigen frequencies, use this result in

$$\frac{x_{k-1} + x_{k+1}}{x_k} = \frac{2\omega_0^2 - \omega^2}{\omega_0^2} = 2\cos\left(\frac{m\pi}{n + 1}\right)$$

$$\omega_m = 2\omega_0 \sin \frac{m\pi}{2(n + 1)}$$
Using these values, we can construct the general solution as

\[ x_k(t) = \sum_{m=1}^{n} \sin\left(\frac{mk\pi}{n+1}\right) \left[ C_m \cos \omega_m t + C'_m \sin \omega_m t \right] \]

The amplitudes are determined from the initial conditions

E.g. if \( x_k(0) = X_k \), \( \dot{x}_k(0) = Y_k \)

\[ \sum_{m=1}^{n} \sin\left(\frac{mk\pi}{n+1}\right) C_m = X_k, \quad k = 1, \ldots, n \]

\[ \sum_{m=1}^{n} \sin\left(\frac{mk\pi}{n+1}\right) \omega_m C'_m = Y_k, \quad k = 1, \ldots, n \]

This system of linear equations can be solved using Matlab.
Example: n=4 beads, generate the sine matrix

```matlab
>> for k=1:4; for m=1:4;
    A(m,k)=sin(m*k*pi/5);
end; end;

>> A
A =
 0.5878   0.9511   0.9511   0.5878
 0.9511   0.5878  -0.5878  -0.9511
 0.9511  -0.5878  -0.5878   0.9511
 0.5878  -0.9511   0.9511  -0.5878

>> X=[1 1 1 1]'; Y=0

>> C=(A\X)'
C =
 1.2311   0.0000   0.2906   0.0000
```
Frequencies of the normal modes

>> for m=1:4
w(m)=2*sin(m*pi/10);
end

>> w
w =

0.6180  1.1756  1.6180  1.9021

So the solution for this initial condition is:

\[
\begin{pmatrix}
  x_1(t) \\
  x_2(t) \\
  x_3(t) \\
  x_4(t)
\end{pmatrix} =
\begin{pmatrix}
  0.5878 & 0.9511 & 0.9511 & 0.5878 \\
  0.9511 & 0.5878 & -0.5878 & -0.9511 \\
  0.9511 & -0.5878 & -0.5878 & 0.9511 \\
  0.5878 & -0.9511 & 0.9511 & -0.5878
\end{pmatrix}
\begin{pmatrix}
  1.2311 \cos \omega_1 t \\
  0 \\
  0.2906 \cos \omega_3 t \\
  0
\end{pmatrix}
\]
Solution for the first bead:

\[ x_1(t) = 0.7236 \cos 0.618t + 0.2764 \cos 1.618t \]

To visualize the solution, plot it in Matlab

\[
\begin{align*}
&>> t=1:0.5:400; \\
&>> f=0.7236*\cos(0.618*t)+0.2906*\cos(1.618*t); \\
&>> \text{plot}(t,f)
\end{align*}
\]

A better way to get an intuitive feeling for the solutions is to animate the positions of the 4 beads as a function of time, e.g., using a bar graph.
Continuum limit \((n \to \infty)\)

When the number of beads are very large, we can take a continuum limit

Introduce total mass and length: \(M = n\mu, \quad L = nl\)

\[
\omega_m^2 = 4 \frac{T}{\mu l} \sin^2 \frac{m\pi}{2(n+1)} \quad \xrightarrow{n \to \infty} \quad 4T \frac{m^2\pi^2}{\mu l 4n^2} = \frac{\pi^2 m^2 T}{ML}
\]

Thus the normal mode frequencies become multiples of the fundamental frequency

\[
\omega_m = m\omega_1, \quad \omega_1 = \sqrt{\frac{\pi^2 T}{ML}}
\]

\[
y_m(x,t) = C_m \sin \left( \frac{m\pi x}{L} \right) \cos \omega_m t
\]

General solution: \(y(x,t) = \sum_{m=1}^{\infty} C_m \sin \left( \frac{m\pi x}{L} \right) \cos m\omega_1 t\)
Vibrations in a two-dimensional system (e.g. membranes) can be analysed in a similar manner

$$z(x, y, t) = C(n_1, n_2) \sin \left( \frac{n_1 \pi x}{L_x} \right) \sin \left( \frac{n_2 \pi y}{L_y} \right) \cos \omega_{12} t$$

The normal mode frequencies are given by

$$\omega_{12}^2 = \frac{SL_x L_y}{M} \left[ \left( \frac{n_1 \pi}{L_x} \right)^2 + \left( \frac{n_2 \pi}{L_y} \right)^2 \right]$$

Where $S$ is the surface tension, and $M$ is the mass.

For equal lengths

$$\omega_{12}^2 = \omega_0^2 (n_1^2 + n_2^2), \quad \omega_0^2 = \frac{\pi^2 S}{M}$$
Fourier series

Any function $f(x)$ in a finite interval $[-L/2, L/2]$ can be represented by a series of sines and cosines

$$f(x) = \frac{1}{2} A_0 + \sum_{m=1}^{\infty} \left[ A_m \cos\left(\frac{2\pi mx}{L}\right) + B_m \sin\left(\frac{2\pi mx}{L}\right) \right]$$

The sines and cosines form an orthogonal set, that is

$$\int_{-L/2}^{L/2} \sin\left(\frac{2\pi mx}{L}\right) \sin\left(\frac{2\pi nx}{L}\right) \, dx = \begin{cases} L/2 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}$$

$$\int_{-L/2}^{L/2} \cos\left(\frac{2\pi mx}{L}\right) \cos\left(\frac{2\pi nx}{L}\right) \, dx = \begin{cases} L/2 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}$$

$$\int_{-L/2}^{L/2} \sin\left(\frac{2\pi mx}{L}\right) \cos\left(\frac{2\pi nx}{L}\right) \, dx = 0$$
Thus we can determine the coefficients by integrating sin (or cos) times \( f(x) \)

\[
\int_{-L/2}^{L/2} f(x) \begin{cases} \sin & \text{if } m \neq 0 \\ \cos & \text{if } m = 0 \end{cases} \, dx = \int_{-L/2}^{L/2} \left\{ \frac{1}{2} A_0 + \sum_{m=1}^{\infty} A_m \cos \left( \frac{2\pi m x}{L} \right) + B_m \sin \left( \frac{2\pi m x}{L} \right) \right\} \begin{cases} \sin & \text{if } m \neq 0 \\ \cos & \text{if } m = 0 \end{cases} \, dx
\]

The right hand side is non-zero only for one value of \( m \), which yields

\[
A_m = \frac{2}{L} \int_{-L/2}^{L/2} f(x) \cos \left( \frac{2\pi m x}{L} \right) \, dx
\]

\[
B_m = \frac{2}{L} \int_{-L/2}^{L/2} f(x) \sin \left( \frac{2\pi m x}{L} \right) \, dx
\]

The fourier series arise in solutions of differential equations with a finite boundary, e.g.:

Laplace and Poisson equations (E&M), Heat equation (thermodynamics, Schröedinger equation (quantum mechanics)
Discrete Fourier transform (DFT)

Very commonly, an observable is sampled at evenly spaced time intervals, e.g., its value is recorded at every $\Delta t$ seconds, which is called the sampling rate.

Inverse of $2\Delta t$ is called the Nyquist critical frequency ($f = \omega / 2\pi$)

$$f_c = \frac{1}{2\Delta t}$$

Sampling theorem: If a continuous function is band-width limited to frequencies smaller than $f_c$, then it is completely determined by its samples.

Aliasing effect: Conversely, if a function is not band-width limited to frequencies smaller than $f_c$, then the frequency components greater than $f_c$ are falsely translated.
Consider a data set \( x_k \) sampled at equal intervals \( N \) times:

\[
\{x_0, x_1, \ldots, x_{N-1}\}
\]

Its discrete Fourier transform is given by

\[
X_n = \sum_{k=0}^{N-1} e^{2\pi i nk/N} x_k, \quad n = 0, 1, \ldots, N - 1
\]

This can be written in terms of sines and cosines as

\[
X_n = \sum_{k=0}^{N-1} \cos\left(\frac{2\pi nk}{N}\right)x_k + i \sum_{k=0}^{N-1} \sin\left(\frac{2\pi nk}{N}\right)x_k
\]

Inverse DFT \( x_k = \frac{1}{N} \sum_{n=0}^{N-1} e^{-2\pi i kn/N} X_n \)

Thus DFT transforms \( N \) (complex) numbers into \( N \) (complex) numbers
Introduce the n’th root of unity: $z = e^{2\pi i / N}$

The DFT can be written as

$$X_n = \sum_{k=0}^{N-1} z^{nk} x_k$$

In matrix form:

$$
\begin{pmatrix}
X_0 \\
X_1 \\
X_2 \\
\vdots \\
X_{N-1}
\end{pmatrix} = 
\begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & z & z^2 & \cdots & z^{N-1} \\
1 & z^2 & z^4 & \cdots & z^{2N-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & z^{N-1} & z^{2N-2} & \cdots & z^{(N-2)^2}
\end{pmatrix} 
\begin{pmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{N-1}
\end{pmatrix}
$$
Fast Fourier transform (FFT)

The matrix multiplication in DFT requires about $N^2$ operations.

In FFT the number of operations is reduced to $N \cdot \log N$.

Assume $N$ is even, so that $N/2$ is an integer

$$X_n = \sum_{k=0}^{N-1} e^{2\pi ink/N} x_k$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi in(2j)/N} x_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi in(2j+1)/N} x_{2j+1}$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi inj/(N/2)} x_{2j} + z^n \sum_{j=0}^{N/2-1} e^{2\pi inj/(N/2)} x_{2j+1}$$

$$= X_n^e + z^n X_n^o$$

If $N/2$ is also even, we can apply the same procedure to the new set.
In FFT, the number of data points is chosen as N=2^p.
Then the above process can be carried out p times, which results in one-point DFT for each of the N values generated.
Working ones way back requires repeated addition and subtraction of these values weighted by the powers of z (p times).
Matlab has special functions for this purpose:
fft: finds the DFT of a given sample using FFT
ifft: finds the inverse DFT
Assuming x is a sample of N data point
>> y=fft(x)
returns N complex numbers that correspond to DFT.
The absolute square of y plotted against the frequency provides the spectral information (periodogram).