DFT (theory and lab example).
Discrete Fourier Transform (Reprise)

**Issue**: Often discretely sample a waveform and need to know: its shape and/or its frequency characteristics.

Sample (i.e., measure w/ an instrument) every $\Delta$ seconds. ($1/\Delta$ is the “sampling rate”)

$$\cos(2\pi f t) \quad f = 2$$

sampled every 1 second

“Nyquist critical frequency” $f_c \equiv \frac{1}{2\Delta}$
Discrete Fourier Transform

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

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

make_fourier_data.cc

dft.cc

inv_dft.cc

Q: \( \sin(2\pi f t) \): where is peak \( n \) in \( H_n \)?
Nyquist critical frequency \( f_c \equiv \frac{1}{2\Delta} \)

Good news:
- Continuous \( h(t) \) samples at intervals \( \Delta \)
- “bandwidth” limited to frequencies \(< f_c \), i.e., \( H(f) = 0 \ \forall \ |f| > f_c \)

\[ h(t) = \Delta \sum_{n=-\infty}^{+\infty} h_n \frac{\sin[2\pi f_c(t-n\Delta)]}{\pi(t-n\Delta)} \]

Bad news:
- If \( h(t) \) not bandwidth limited to \( f < f_c \)
- All of PSD outside \(-f_c < f < f_c\) ("aliasing")
Limit sampled frequencies to $< f_c$
Fast Fourier Transform

DFT is slow. Execution time $\propto N^2$.
Compare $N = 1000$ to $N = 5000$ (yes, try it now…)

Fast Fourier Transform (FFT) to the rescue.
Not a new type of transform, but a new way to calculate DFT.
Uses a “divide and conquer” strategy.

$$W \equiv e^{2\pi i/N} \quad \text{“twiddle factor” (Nth root of 1)}$$

$$H_n \equiv \sum_{k=0}^{N-1} W^{nk} h_k \quad \text{index } n: 0 \rightarrow N-1$$

$$F_k = \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \quad \text{index } k: 0 \rightarrow N-1$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi i(2j)k/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi i(2j+1)k/N} f_{2j+1}$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi ijk/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ijk/(N/2)} f_{2j+1}$$

repeat…

$$= F_k^e + W^k F_k^o \quad 0 \leq k \leq N-1$$

Execution time $\propto N \log_2 N \quad \text{w/ } N = \text{power of 2}$
Fast Fourier Transform (2)

make_fourier_data.cc
gsl_fft.cc
gsl_inv_fft.cc

For optimum efficiency, make N = power of 2
e.g., N = 1024, N = 2048, …, 1 048 576 , …

GSL FFT routines store Imag(H_n) differently from DFT
H(k) = H(N-k)* (Not all 2N H(k) numbers independent.)

- Compare DFT output w/ FFT output.
- Compare execution times: DFT v. FFT w/ N = 4096
FFT (3)

square wave data

dft.dat

<table>
<thead>
<tr>
<th>n</th>
<th>a(n)</th>
<th>b(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-27.3487</td>
<td>1.72063</td>
</tr>
<tr>
<td>2</td>
<td>13.846</td>
<td>-1.74915</td>
</tr>
<tr>
<td>3</td>
<td>-0.32786</td>
<td>0.0625427</td>
</tr>
<tr>
<td>4</td>
<td>-6.52502</td>
<td>1.67534</td>
</tr>
<tr>
<td>5</td>
<td>5.41695</td>
<td>-1.76007</td>
</tr>
<tr>
<td>6</td>
<td>-0.311563</td>
<td>0.123357</td>
</tr>
</tbody>
</table>

inefficient storage

Why the relative minus (-) sign for Imag H(n) ?

gsl_fft.dat

<table>
<thead>
<tr>
<th>n</th>
<th>a(n)</th>
<th>b(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-27.3487</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>13.846</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.32786</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.72063</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5.41695</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.311563</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-6.52502</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-1.67534</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.76007</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.32786</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.76007</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-0.311563</td>
<td></td>
</tr>
</tbody>
</table>
Summary

Sampling theory and aliasing.

FFT (theory and exercises).

Don’t suffer in silence. Scream for help!!!