Lecture 11

Austin Baird
UNC Department of Mathematics
UNC Department of Biology

May 21, 2014

Summary

Today we will:

Try and understand series approximations and transforms
(Fourier Analysis!)

Periodic Functions

A period function satisfies the relationship:

$$f(t + nP) = f(t)$$
$$n \in \mathcal{Z}$$

Here the period is denoted P and so this relations shows that for any chosen n the graph will repeat itself with period P. Think of reconstructing a graph to have as large of a domain as you'd like by reflecting a function over and over again about the period!

Series Expansion of Periodic Functions

Sometimes, Periodic functions can be written as an infinite series of sin and cos's. This can be referred to as the Fourier series of a function:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right)\right]$$

Here our coefficients a_n and b_n can be written as:

$$a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\left(\frac{n\pi x}{L}\right)dx$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\left(\frac{n\pi x}{L}\right)dx$$

Let's Try It!

Find the Fourier series of the sawtooth function:

$$f(x) = x \quad -L < x < L$$
$$f(x + 2L) = f(x)$$

Now we want to graphically show that if we truncate this series it gives a good approximation. Do It!

Let's Try It!

Find the Fourier series of the sawtooth function:

$$f(x) = x \quad -L < x < L$$
$$f(x + 2L) = f(x)$$

Now we want to graphically show that if we truncate this series it gives a good approximation. Do It!

$$a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\left(\frac{n\pi x}{L}\right)dx = 0 \quad n = 0, 1, 2\dots$$
$$b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\left(\frac{n\pi x}{L}\right)dx = \frac{-2L}{n\pi}(-1)^n \quad n = 1, 2, 3\dots$$

Example Continued

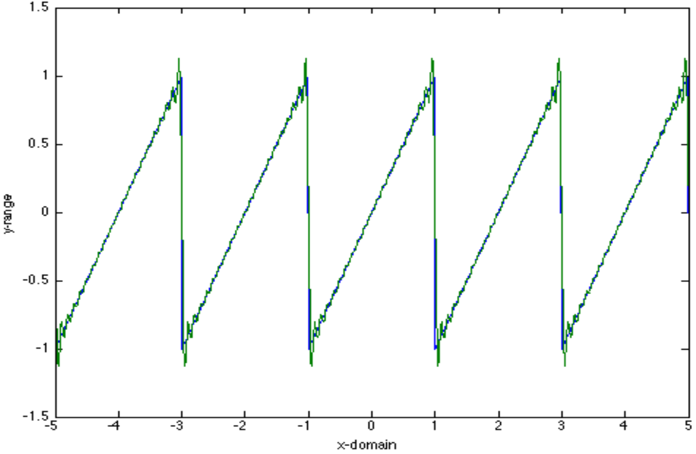
We now have a full formula for the series of our sawtooth function:

$$f(x) = \sum_{n=1}^{\infty} \left[\frac{-2L}{n\pi} (-1)^n \sin\left(\frac{n\pi x}{L}\right) \right]$$

This function is an exact approximation to our function except at the discontinuities!

Comparison

Sampling a finite number of series terms we can get a pretty good representation:



Fourier Integral Representation

Unfortunately the fourier series representation of a function only works for periodic functions, what about non-periodic functions? For this we use the Fourier integral representation:

$$f(x) = \int_0^\infty a(k)\cos(kx)dk + \int_0^\infty b(k)\sin(kx)dk$$
$$a(k) = \frac{1}{\pi}\int_\infty^\infty f(x)\cos(kx)dx$$
$$b(k) = \frac{1}{\pi}\int_\infty^\infty f(x)\sin(kx)dx$$

This definition now has a continuous frequency resolution and is defined for non-period functions defined on $(-\infty, \infty)$

Fourier transform!

We can now write this representation using a singular integral and using Eulers equation: $e^{i\theta} = \cos(\theta) + i \sin\theta$.

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} f(x) e^{-2\pi i x k} dx$$

With an inverse defined to be:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} \hat{f}(k) e^{2\pi i x k} dk$$

Similarly we could transform a function defined with time (t). Also note that k is the frequency of the function.

Discrete Transform

Now assume that we aren't transforming a function but a sequence of data, $\{f_k\}_{k=0}^{N-1}$ which we know contains some oscillations: (sample the sin wave) we get:

$$\hat{F}_n = \sum_{n=0}^{N-1} f_k e^{\frac{-2\pi i k n}{N-1}}$$

Here $\hat{F}_n$ is the Fourier transform of the data f_k sampled at N data points. Here k is the frequency. Note that $\hat{F}_n$ is a sequence of numbers: $n = 0, 1, \ldots N-1$ which is the same size as the number of times we sampled the data: f_k

Note: We cannot compute zero or negative indices in matlab, so we will shift this formula by 1:

$$\hat{F}_n = \sum_{n=1}^{N} f_k e^{\frac{-2\pi i (k-1)(n-1)}{N-1}}$$

Discrete Transform

Turns out we can write this as matrix vector multiplication! If we define an NxN matrix for $\omega_n = e^{\frac{-2\pi i}{n}}$ we can define it to be:

$$\begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega_n^{1\cdot 1} & \dots & \omega_n^{1\cdot(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{(n-1)\cdot 1} & \dots & \omega_n^{(n-1)\cdot(n-1)} \end{pmatrix} \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \end{pmatrix}$$

Now we have what we need to get $[F_n]$. Since we know f_k, sampled data, and we can compute the matrix!

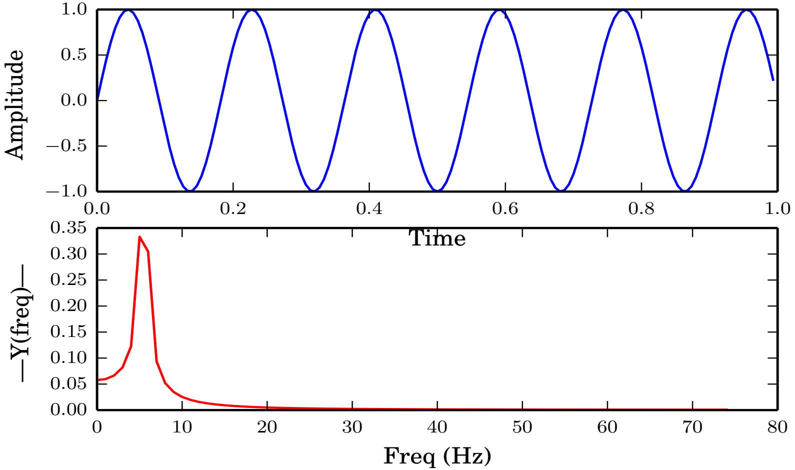
Fast Fourier Transform

So what is wrong with what we just computed? Computational Cost! To compute this directly $= \mathcal{O}(n^2)$, but there is a new way! FFT! Idea:

- Use symmetry of the matrix to compute the multiplication without recomputing matrix terms!
- Speeds up computational time to $\mathcal{O}(n\log(n))$

Examples

Periodic data = frequency analysis:



But how do we get the frequency!?

Frequency

To get the frequency spectrum of the data you are analyzing, do the following (in Matlab) :

```
Fs = 150;    sampling frequency
t = 0:1/Fs:1;    discretized time domain
f = 5;    frequency of our sin wave
y = sin(2*pi*f*t);
nfft = 1024;    length of the fft domain, power of 2
x = fft(y,nfft);
x = abs(x(1:nfft/2)); Half of the data, fft is symmetric
f = (0:nfft/2-1)*Fs/nfft;    frequency vector
```

Homework 10

We want to gain an understanding of Fouier analysis a variety of ways:

- Compute the series expansion for the top hat periodic function:

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

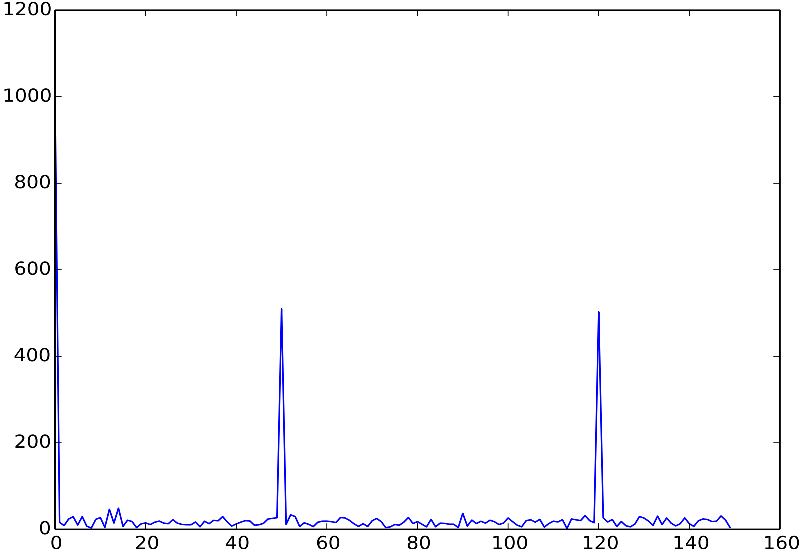
$$
f(x+2) = f(x) \quad \text{(periodic condition)}
$$

Graph the actual function and the series representation truncated at the first 2,4,8,10 terms.

Homework 10

- Program the Discrete Fourier transform for a periodic function and compute it's frequency.
- Use the periodic function $f(t) = sin(2\pi 50t) + sin(2\pi 120t)$ in your computations (it has what frequencies?)
- Vary the number of sampling points and test the time it takes to compute.
    - Let sampling be power of 2 and compute the time it takes to compute the problem (graph the data x-axis = sample points, y-axis = compute time)
    - Watch this movie: `http://blogs.mathworks.com/pick/2008/02/12/timing-code-in-matlab/`, it will help you!

This is what it should look like



Homework 10

- Use the data provided
- The data is a loaded elastic material extended into a sin wave, immersed into a fluid. The deformed boundary is released and then vibrates to rest.
- Goal: Extract the frequency of the vibrating boundary!
  - Load it into matlab
  - make a graph of it over time (it contains two columns, position and time)
  - Take an fft of the data
  - Graph the frequency spectrum
  - Explain issues and why there are several predominant frequencies.