Fourier Series, Discrete Fourier Transforms and Fast Fourier Transforms

1. Complex Numbers

We start with some basics of complex numbers:

If $z = a + bi$ is a complex number, then the **complex conjugate** of z is the complex number $\overline{z} = a - bi$.

From Euler's theorem, we have a relationship between $e^x, \cos x$ and $\sin x$:

$$e^{ix} = \cos x + i \sin x$$

and

$$e^{-ix} = \cos x - i \sin x$$

We will define $\omega_N = e^{-\frac{2\pi i}{N}}$ for N a positive integer. ω_N is called a **primitive Nth root of unity**. In particular, ω_N is a root of $x^N - 1$ and is not a root of $x^k - 1$ for $k < N$. Some useful properties of ω_N are:

$$\omega_N^k = \omega_M \text{ if } N = kM$$
$$\omega_N^{N/2} = -1 \text{ if } N \text{ is even}$$
$$\omega_N^M = \omega_N^r \text{ if } M = kN + r$$

2. Fourier Series over $[0, 2\pi]$

In this section we consider the function space $C[0, 2\pi]$. Our initial goal will be to approximate a given function $f(x)$ in $C[0, 2\pi]$ using linear combinations of functions of the form $\cos kx$ and $\sin kx$ where k is a non-negative integer. This is obviously useful for periodic functions. Our setup is as follows:

$$\langle f(x), g(x) \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) g(x)\, dx$$

and

$$S_n = \text{Span}(\frac{1}{\sqrt{2}}, \sin x, \ldots, \sin nx, \cos x, \ldots, \cos nx)$$

We can check that the following are true:

$$\langle 1, \cos \alpha x \rangle = \frac{1}{\pi} \int_0^{2\pi} \cos \alpha x \, dx = 0$$
$$\langle 1, \sin \beta x \rangle = \frac{1}{\pi} \int_0^{2\pi} \sin \beta x \, dx = 0$$
$$\langle \cos \alpha x, \cos \beta x \rangle = \frac{1}{\pi} \int_0^{2\pi} \cos \alpha x \cos \beta x \, dx = \begin{cases} 1 \text{ if } \alpha = \beta \\ 0 \text{ if } \alpha \neq \beta \end{cases}$$
$$\langle \sin \alpha x, \sin \beta x \rangle = \frac{1}{\pi} \int_0^{2\pi} \sin \alpha x \sin \beta x \, dx = \begin{cases} 1 \text{ if } \alpha = \beta \\ 0 \text{ if } \alpha \neq \beta \end{cases}$$
$$\langle \cos \alpha x, \sin \beta x \rangle = \frac{1}{\pi} \int_0^{2\pi} \cos \alpha x \sin \beta x \, dx = 0$$

Therefore

$$\{\frac{1}{\sqrt{2}}, \sin x, \ldots, \sin kx, \cos x, \ldots, \cos kx\}$$

is an orthonormal basis for S_k.

If we define:

$$a_0 = \langle f(x), 1 \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x)\, dx$$

$$a_k = \langle f(x), \cos kx \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx\, dx$$

and

$$b_k = \langle f(x), \sin kx \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx\, dx$$

then the least squares approximation of $f(x)$ in S_n will be:

$$s_n(f(x)) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx)$$

This sum is an example of a **trigonometric polynomial** and is known as the n**th degree Fourier series for** $f(x)$ **over** $[0, 2\pi]$.

It is generally convenient to write this series in exponential form, which is what we proceed to do:

Let

$$c_k = \frac{1}{2}(a_k - ib_k)$$

and

$$c_{-k} = \overline{c_k}$$

With these definitions, we have the following relations:

$$a_k = c_k + c_{-k}$$

and

$$b_k = i(c_k - c_{-k})$$

So

$$c_k = \frac{1}{2}(a_k - ib_k) = \frac{1}{2\pi} \int_0^{2\pi} f(x)(\cos kx - i \sin kx)\, dx$$

Using Euler's theorem, we have:

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx}\, dx$$

Finally, notice that

$$c_k e^{ikx} + c_{-k} e^{-ikx} = c_k(\cos kx + i \sin kx) + c_{-k}(\cos kx - i \sin kx) = (c_k + c_{-k}) \cos kx + i(c_k - c_k) \sin kx$$

This gives:

$$c_k e^{ikx} + c_{-k} e^{-ikx} = a_k \cos kx + b_k \sin kx$$

Therefore the nth degree Fourier series:

$$s_n(f(x)) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx)$$

can be rewritten as:

$$s_n(f(x)) = \sum_{k=-n}^{n} c_k e^{ikx}$$

Fourier series have many applications. One of the most important is in the area of signal processing. The Fourier series of a noisy function can be used to reduce this noise. Noise in this case refers to the terms of $s_n(f(x))$ where $|k|$ is large. We can eliminate this noise by setting the corresponding c_k to 0.

3. Discrete Fourier Transform

In applications, we often do not know $f(x)$ explicitly, rather we have to approximate $f(x)$ by sampling. This will lead to an approximation of $s_n(f(x))$.

First, we partition the interval $[0, 2\pi]$ into N equal parts:

$$x_0 = 0, x_1 = \frac{2\pi}{N}, \ldots, x_N = 2\pi$$

In particular,

$$x_j = \frac{2\pi j}{N}$$

For each x_j we measure the output y_j and set

$$y_j = f(x_j)$$

Since we do not know $f(x)$, we can not compute $c_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\,dx$. Rather we will approximate this integral using the left endpoint Riemann sum given the partition above:

$$c_k \approx \sum_{j=0}^{N-1} f(x_j)e^{-ikx_j}\frac{1}{N}$$

Note that

$$e^{ikx_j} = e^{-\frac{2\pi ikj}{N}} = \omega_N^{jk}$$

where ω_N is the primitive Nth root of unity defined in section 1. Therefore

$$\sum_{j=0}^{N-1} f(x_j)e^{-ikx_j}\frac{1}{N} = \frac{1}{N}\sum_{j=0}^{N-1} y_j\omega_N^{jk}$$

We define

$$d_k = \sum_{j=0}^{N-1} y_j\omega_N^{jk}$$

The sequence $\{d_0, d_1, \ldots, d_{N-1}\}$ is called the **discrete Fourier transform** or **DFT** of $\{y_0, y_1, \ldots, y_{N-1}\}$

We will now show how to compute d_k using matrix multiplication. Let F_N be the $N \times N$ matrix whose k, j entry is $\omega_N^{(j-1)(k-1)}$ so that

$$F_N = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^6 & \cdots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \omega_N^9 & \cdots & \omega_N^{3(N-1)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \omega_N^{3(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix}$$

Note that F_N is a symmetric matrix (ie. $F_N^T = F_N$).

If we define $\mathbf{y}$ and $\mathbf{d}$ by:

$$\mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix} \text{ and } \mathbf{d} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{N-1} \end{pmatrix}$$

Then a direct calculation shows that

$$F_N\mathbf{y} = \mathbf{d}$$

4. Fast Fourier Transform

Using the matrix calculation above to compute the discrete Fourier transform of a sequence $\{y_0, y_1, \ldots, y_{N-1}\}$ requires approximately $8N^2$ operations. There are many algorithms which can be used to speed this processes up, they are collectively known as the **fast Fourier transform** or **FFT**. The best known of these is the Cooley-Tukey algorithm. This algorithm was developed by Cooley and Tukey in the 1960's, though it was later found to be originally developed by Gauss in the early 1800's. This algorithm works when N is even and relies upon reordering the columns of F_N.

To get started, we define two matrices as follows: Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N\}$ be the standard basis for $\mathbb{R}^n$. Then P_N is defined to be:

$$P_N = (\mathbf{e}_1\ \mathbf{e}_3\ \mathbf{e}_5 \cdots \mathbf{e}_{N-1}\ \mathbf{e}_2\ \mathbf{e}_4 \cdots \mathbf{e}_N)$$

Notice that P_N is a $N \times N$ permutation matrix, in particular $P_N^{-1} = P_N^T$. We also define $D_{N/2}$ to be the $N/2 \times N/2$ diagonal matrix whose j, j entry is ω_N^{j-1}.

Our first goal is to rearrange the columns of F_N so that all of the odd numbered columns occur before all of the even columns. This can be done by multiplying F_N on the right by P_N. This has the effect of arranging the columns so that the first $N/2$ columns contain only even powers of ω_N. We then use the relations $\omega_N^2 = \omega_{N/2}$ and $\omega_{N/2}^{k(N/2+1)} = \omega_{N/2}^k$ to reduce the entries to powers of $\omega_{N/2}$. These columns are now seen to consist of two copies of $F_{N/2}$. For the remaining columns, we can do a similar reduction, with the added complication that we have a copy of $D_{N/2}F_{N/2}$ and of $-D_{N/2}F_{N/2}$. Once we do this, we have the following block decomposition of F_N:

$$F_N P_N = \begin{pmatrix} F_{N/2} & D_{N/2}F_{N/2} \\ F_{N/2} & -D_{N/2}F_{N/2} \end{pmatrix}$$

So,

$$F_N\mathbf{y} = (F_N P_N)(P_N^T\mathbf{y})$$

If we write

$$P_N^T\mathbf{y} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}$$

where $\mathbf{w}_1$ is the first half of of $P_N^T\mathbf{y}$ and $\mathbf{w}_2$ is the second half of this vector and use the properties of block multiplication (see section 1.5 of Leon):

$$F_N\mathbf{y} = \begin{pmatrix} F_{N/2} & D_{N/2}F_{N/2} \\ F_{N/2} & -D_{N/2}F_{N/2} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix} = \begin{pmatrix} F_{N/2}\mathbf{w}_1 + D_{N/2}F_{N/2}\mathbf{w}_2 \\ F_{N/2}\mathbf{w}_1 - D_{N/2}F_{N/2}\mathbf{w}_2 \end{pmatrix}$$

Therefore, to compute $F_N\mathbf{y}$ we only need to compute $F_{N/2}\mathbf{w}_1$ and $D_{N/2}F_{N/2}\mathbf{w}_2$. This process can be used recursively, in stages: one stage for each power of 2 which divides N. This algorithm is most efficient when N is a power of 2. In this case the FFT takes approximately $5N\log_2 N$ operations (as opposed to $8N^2$ operations for the DFT). For example, if $N = 1024$ then the dft takes 2^{23} operations whereas the FFT will take $2^{11} * 25$ operations. This is speedup by a factor of $2^{12}/25$.

Example:

Let $N = 6$ and $\mathbf{y} = (1, 2, 3, 4, 5, 6)^T$.

Then

$$F_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_6 & \omega_6^2 & \omega_6^3 & \omega_6^4 & \omega_6^5 \\ 1 & \omega_6^2 & \omega_6^4 & \omega_6^6 & \omega_6^8 & \omega_6^{10} \\ 1 & \omega_6^3 & \omega_6^6 & \omega_6^9 & \omega_6^{12} & \omega_6^{15} \\ 1 & \omega_6^4 & \omega_6^8 & \omega_6^{12} & \omega_6^{16} & \omega_6^{20} \\ 1 & \omega_6^5 & \omega_6^{10} & \omega_6^{15} & \omega_6^{20} & \omega_6^{25} \end{pmatrix}$$

It is advantageous to reduce the various powers of ω_6 to smaller powers of ω_6 and powers of lower order primitive roots of unity (also note that $\omega_6^3 = -1, \omega_6^5 = -\omega_3$ and $\omega_6 = -\omega_3^2$):

$$F_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\omega_3^2 & \omega_3 & -1 & \omega_3^2 & -\omega_3 \\ 1 & \omega_3 & \omega_3^2 & 1 & \omega_3 & \omega_3^2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \omega_3^2 & \omega_3 & 1 & \omega_3^2 & \omega_3 \\ 1 & -\omega_3 & \omega_3^2 & -1 & \omega_3 & -\omega_3^2 \end{pmatrix}$$

$$P_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3^4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix}$$

$$D_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega_6 & 0 \\ 0 & 0 & \omega_6^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\omega_3^2 & 0 \\ 0 & 0 & \omega_3 \end{pmatrix}$$

and

$$D_3 F_3 = \begin{pmatrix} 1 & 1 & 1 \\ -\omega_3^2 & -1 & -\omega_3 \\ \omega_3 & 1 & \omega_3^2 \end{pmatrix}$$

Now compute $F_6 P_6$:

$$F_6 P_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 & -\omega_3^2 & -1 & -\omega_3 \\ 1 & \omega_3^2 & \omega_3 & \omega_3 & 1 & \omega_3^2 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & \omega_3 & \omega_3^2 & \omega_3^2 & 1 & \omega_3 \\ 1 & \omega_3^2 & \omega_3 & -\omega_3 & -1 & -\omega_3^2 \end{pmatrix} = \begin{pmatrix} F_3 & D_3 F_3 \\ F_3 & -D_3 F_3 \end{pmatrix}$$

Therefore

$$\mathbf{d} = F_6 \mathbf{y} = F_6 P_6 P_6^T \mathbf{y} = \begin{pmatrix} F_3 & D_3 F_3 \\ F_3 & -D_3 F_3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 5 \\ 2 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} F_3 \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} + D_3 F_3 \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \\ F_3 \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} - D_3 F_3 \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \end{pmatrix}$$

Finally,

$$F_3 \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} = \begin{pmatrix} 9 \\ 1 + 3\omega_3 + 5\omega_3^2 \\ 1 + 3\omega_3^2 + 5\omega_3 \end{pmatrix}$$

and

$$D_3 F_3 \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -\omega_3^2 & -1 & -\omega_3 \\ \omega_3 & 1 & \omega_3^2 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 12 \\ -2\omega_3^2 - 4 - 6\omega_3 \\ 2\omega_3 + 4 + 6\omega_3^2 \end{pmatrix}$$

So

$$\mathbf{d} = F_6 \mathbf{y} = \frac{1}{6} \begin{pmatrix} 9 + 12 \\ -3 - 3\omega_3 + 3\omega_3^2 \\ 5 + 7\omega_3 + 9\omega_3^2 \\ 9 - 12 \\ 5 + 9\omega_3 + 7\omega_3^2 \\ -3 + 3\omega_3 - 3\omega_3^2 \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 21 \\ -3.0000 + 5.1962i \\ -3.0000 + 1.7321i \\ -3.0000 \\ -3.0000 - 1.7321i \\ -3.0000 - 5.1962i \end{pmatrix}$$