Research | November 01, 2018

A Visual Way to Teach the Fast Fourier Transform

By Jithin D. George

The algorithm behind the fast Fourier transform (FFT) has a simple yet beautiful geometric interpretation that is frequently lost in translation in a classroom. Here I provide a visual perspective that aims to capture the algorithm's essence.

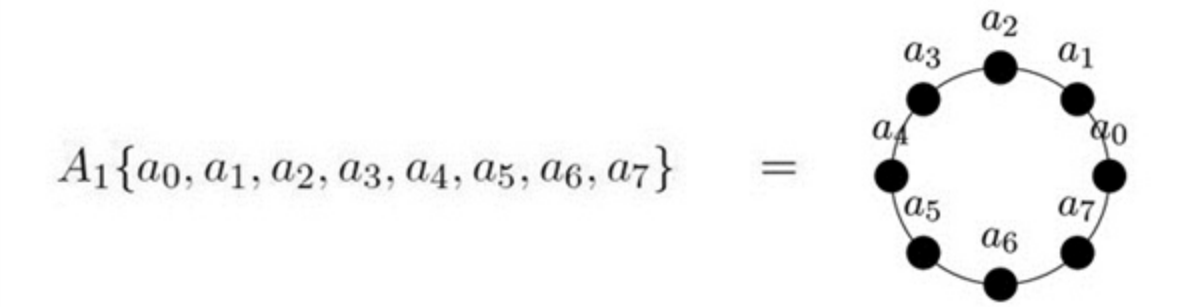

Figure 1. The "*n*"th term in a discrete Fourier transform can be expressed as the summation of points that lie on a circle separated by angle θ_n.

Students are often confused when they encounter the FFT for the first time. This confusion likely stems from two sources:

1. The belief that one needs to completely understand the Fourier transform to comprehend the FFT. This is not true; the FFT is simply an efficient way to compute sums of a special form, and the terms in the discrete Fourier transform (DFT) just happen to be in that form:

$$A_k = \sum_{n=0}^{N-1} a_n e^{-i\frac{2\pi n}{N}k}. \tag{1}$$

2. The standard presentation of the Cooley-Tukey algorithm [1]. This is the heart of the FFT, and indicates that it is possible to decompose the DFT of a sequence of terms into a DFT of even terms and a DFT of odd terms. When applied recursively, it results in a computational cost of $O(N \log N)$. Researchers generally use the following decomposition of A_k into odd and even terms to illustrate the idea:

$$\sum_{n=0}^{N-1} a_n e^{-i\frac{2\pi n}{N}k} = \sum_{n=0}^{N/2-1} a_{2n} e^{-i\frac{2\pi (2n)}{N}k} + \sum_{n=0}^{N/2-1} a_{2n+1} e^{-i\frac{2\pi (2n+1)}{N}k}. \tag{2}$$

Let us take a simplified look at the terms in a DFT:

$$A_k = \sum_{n=0}^{N-1} a_n e^{-in\frac{2\pi}{N}k} = \sum_{n=0}^{N-1} a_n e^{ik\theta_n}. \tag{3}$$

One can visualize $a_n e^{ik\theta_n}$ as the value a_n located at angle $k\theta_n$ on a unit circle in the complex plane. As n goes from 0 to $\mathrm{N}-1$, the θ_ns divide the circle into N arcs of angle $\frac{2\pi}{N}$. Each term in the summation in (3) is a multiple of a point on the unit circle in the complex plane (see Figure 1). With this geometric view, the Cooley-Tukey algorithm in (2) becomes obvious through Figure 2.

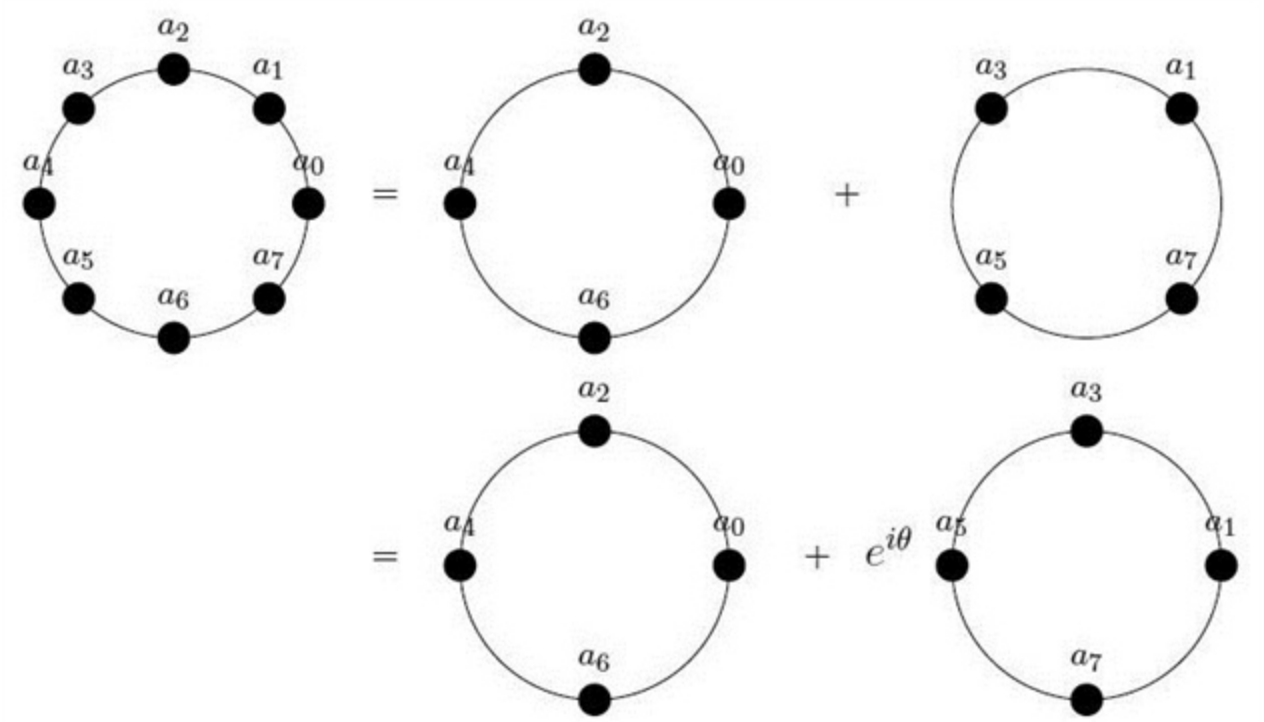

Figure 2. The circular representation of a discrete Fourier transform term can be split into the circular representations of its even and odd components (with a rotation).

We can compute sums like the FFT in this way because the odd terms are a “rotation” away from the even terms. This is quite elegant, but does not provide any new computational efficiency in itself. We are able to decompose a sum into two smaller sums of half the size, but still must calculate all of the sums. The smaller sums’ ability to be “recycled” into new sums gives the FFT its computational efficiency. We can recycle the two terms that when added yield A_1 by subtracting them to produce A_5 (see Figure 3). This certainly saves some computational cost, but how much? To obtain the finer details, we must work out a simple example.

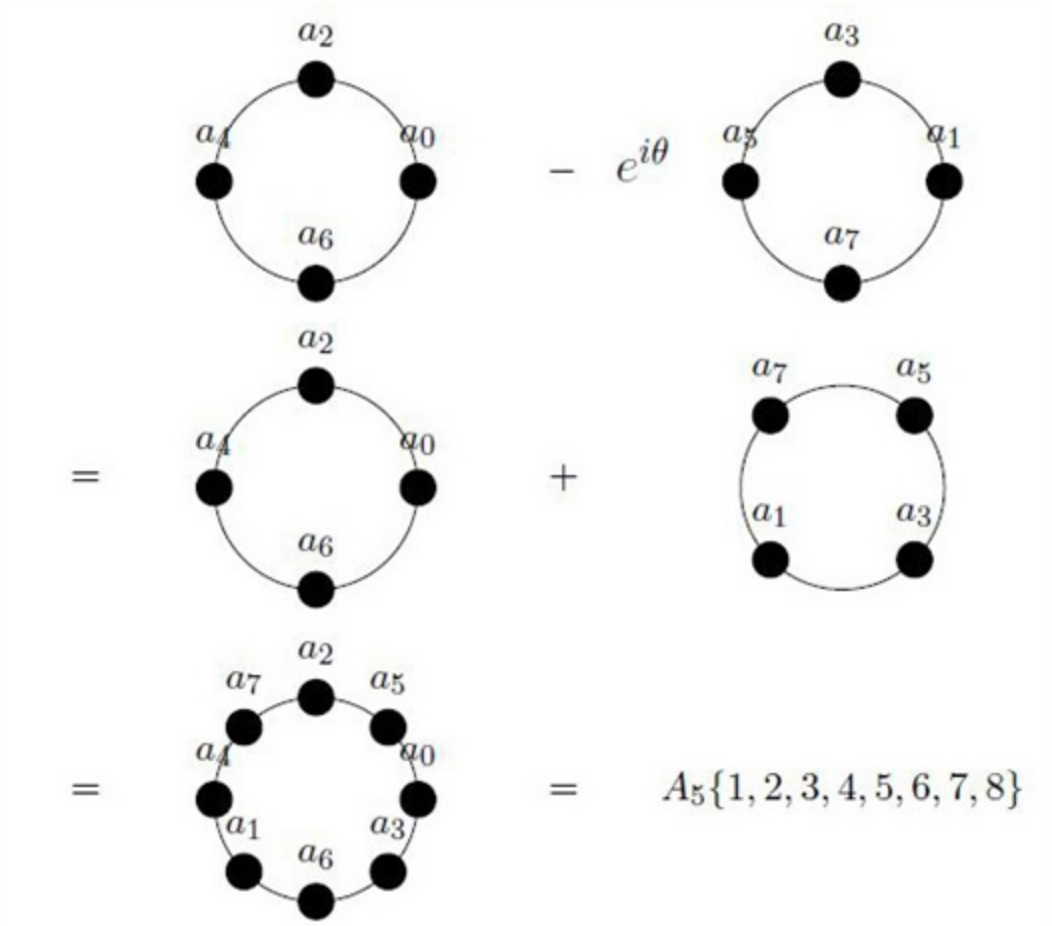

Figure 3. The even and odd circles from Figure 2 can be recycled to yield a completely different term of the discrete Fourier transform.

To that end, let us examine the DFT of the vector $\{a_0, a_1, a_2, a_3\}$ (see Figure 4). We can obtain the FFT of $\{a_0, a_1, a_2, a_3\}$ using the terms in Figure 5. The first two terms are the FFT of $\{a_0, a_2\}$, and the last two form the FFT of $\{a_1, a_3\}$. Thus, one can decompose an FFT into an FFT of even terms and an FFT of odd terms. This saves a lot of computational cost — almost half, since computing the DFT naively yields $\mathcal{O}(N^2)$. It also varies from the decomposition of sums with a cost of $\mathcal{O}(N)$, where decomposition did not help conserve computational cost.

FFT$\{a_1, a_2, a_3, a_4\}$ has the combined computational cost of FFT$\{a_1, a_3\}$, FFT$\{a_2, a_4\}$, and $4c_2$, where c_2 represents the cost of multiplication and addition for each A_n.

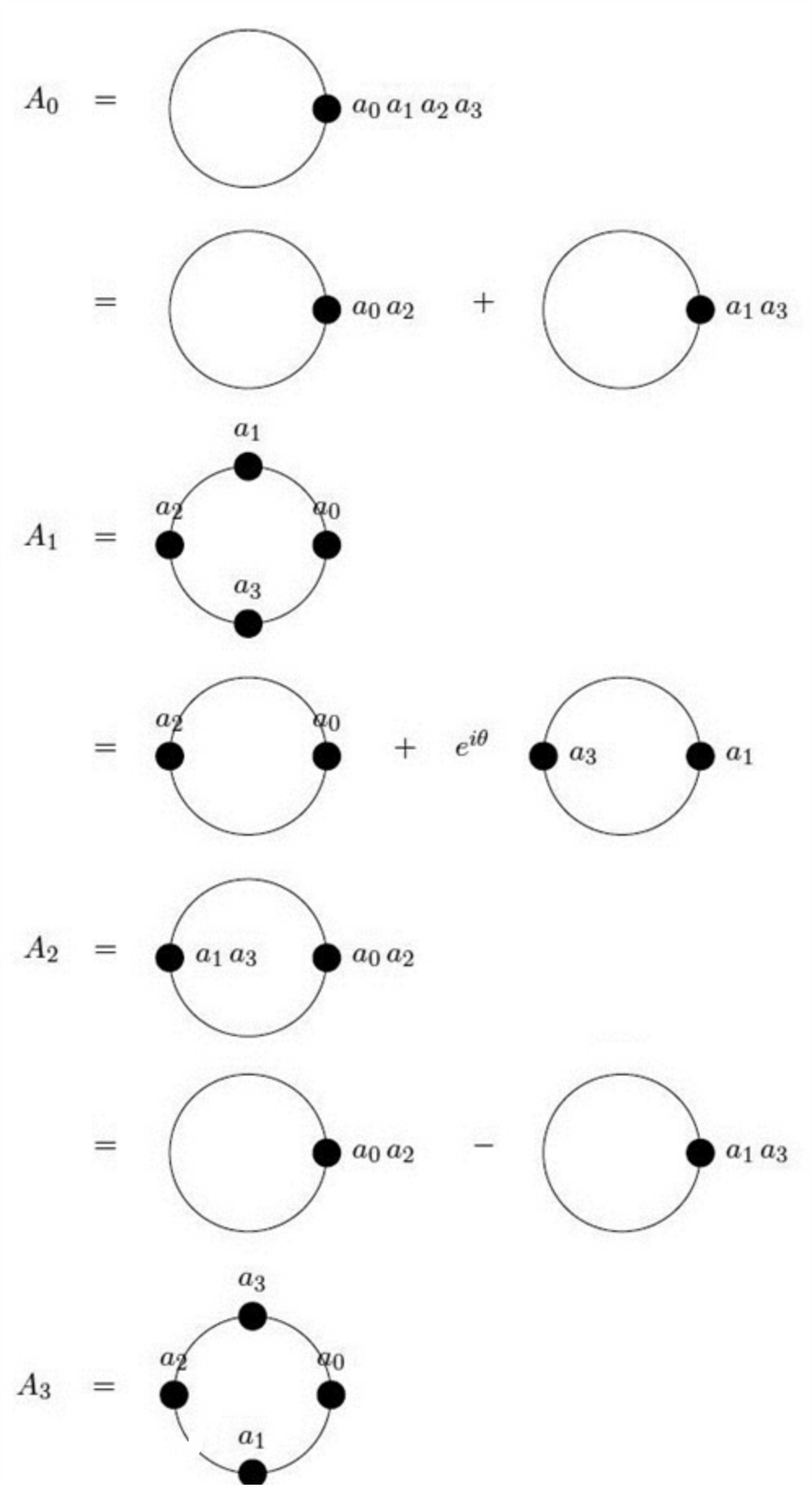
$A_0 = a_0\, a_1\, a_2\, a_3$
$= a_0\, a_2 + a_1\, a_3$
$A_1 =$
a_1
a_2
a_0
a_3
$= a_2 \quad a_0 + e^{i\theta} \quad a_3 \quad a_1$
$A_2 = a_1\, a_3 \quad a_0\, a_2$
$= a_0\, a_2 - a_1\, a_3$
$A_3 =$
a_3
a_2
a_0
a_1

Figure 4. All of the terms in the discrete Fourier transform of {a_0, a_1, a_2, a_3}.

When expanding the FFTs recursively, FFT$\{a_1, a_2, a_3, a_4\}$ has the combined computational cost of FFT$\{a_1\}$, FFT$\{a_3\}$, FFT$\{a_2\}$, FFT$\{a_4\}$, and $8c_2$. The cost $8c_2$ comes from the $4c_2$ cost of operations required to combine the one-point DFTs to form each of the four terms in Figure 5, plus the $4c_2$ cost from the previous step.

Naively computing the DFT with N points requires $c_1 N^2$ work, while computing two DFTs of size $\frac{N}{2}$ requires only half as much work: $2c_1(\frac{N}{2})^2 = \frac{1}{2}c_1 N^2$, plus $c_2 N$ work to combine the two results.

We can apply this recursively $\log_2 N$ times to completely eliminate the quadratic cost, leaving only the cost of N one-point DFTs plus $\log_2 N$ combinations—each requiring $\mathcal{O}(N)$ work—for a total of $\mathcal{O}(N)\log_2 N$ work.

Thus, the total cost in general is $\mathcal{O}(N) + cN\log_2(N) \approx \mathcal{O}(N\log_2(N))$.

An earlier version of this description is available in [2].

Figure 5. The first two terms are the discrete Fourier transform (DFT) of {a_0, a_2} and the last two are the DFT of {a_1, a_3}. Together they give the DFT of {a_0, a_1, a_2, a_3}, as shown in Figure 4.

The figures in this article were provided by the author.

Are you a graduate student? Have you discovered a unique way to learn, teach, or understand a complicated mathematical concept? Write to us at sinews@siam.org! We may publish your insights in an upcoming issue of SIAM News.

References

[1] Cooley, J.W., & Tukey, J.W. (1965). An algorithm for the machine calculation of complex Fourier series. *Math. Comp., 19*(90), 297-301.

[2] George, J.D. (2018). The right way to teach the FFT. Preprint, *arXiv:1805.08633*.

Jithin George is a Ph.D. student in the Department of Engineering Sciences and Applied Mathematics at Northwestern University. He completed this work as a master's student in the Department of Applied Mathematics at the University of Washington.