# Fast Fourier Transform (FFT)

*Wikipedia*

A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the complexity of computing the DFT from $O(N^2)$, which arises if one simply applies the definition of DFT, to $O(N \log N)$, where $N$ is the data size. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In the presence of round-off error, many FFT algorithms are much more accurate than evaluating the DFT definition directly. There are many different FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory.

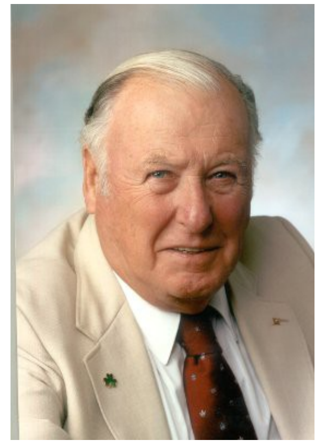
The development of fast algorithms for DFT can be traced to Gauss's unpublished work in 1805 when he needed it to interpolate the orbit of asteroids Pallas and Juno from sample observations. His method was very similar to the one published in 1965 by Cooley and Tukey, who are generally credited for the invention of the modern generic FFT algorithm. While Gauss's work predated even Fourier's results in 1822, he did not analyze the computation time and eventually used other methods to achieve his goal.

Between 1805 and 1965, some versions of FFT were published by other authors. Frank Yates in 1932 published his version called interaction algorithm, which provided efficient computation of Hadamard and Walsh transforms. Yates' algorithm is still used in the field of statistical design and analysis of experiments. In 1942, G. C. Danielson and Cornelius Lanczos published their version to compute DFT for x-ray crystallography, a field where calculation of Fourier transforms presented a formidable bottleneck. While many methods in the past had focused on reducing the constant factor for $O(N^2)$ computation by taking advantage of "symmetries", Danielson and Lanczos realized that one could use the "periodicity" and apply a "doubling trick" to get $O(N \log N)$ runtime.

James Cooley and John Tukey published a more general version of FFT in 1965 that is applicable when N is composite and not necessarily a power of 2. Tukey came up with the idea during a meeting of President Kennedy's Science Advisory Committee where a discussion topic involved detecting nuclear tests by the Soviet Union by setting up sensors to surround the country from outside. To analyze the output of these sensors, a fast Fourier transform algorithm would be needed. In discussion with Tukey, Richard Garwin recognized the general applicability of the algorithm not just to national security problems, but also to a wide range of problems including one of immediate interest to him, determining the periodicities of the spin orientations in a 3-D crystal of Helium-3. Garwin gave Tukey's idea to Cooley (both worked at IBM's Watson labs) for implementation. Cooley and Tukey published the paper in a relatively short time of six months. As Tukey did not work at IBM, the patentability of the idea was doubted and the algorithm went into the public domain, which, through the computing revolution of the next decade, made FFT one of the indispensable algorithms in digital signal processing.

Fast Fourier transforms are widely used for applications in engineering, science, and mathematics. The basic ideas were popularized in 1965, but some algorithms had been derived as early as 1805. In 1994, Gilbert Strang described the FFT as "the most important numerical algorithm of our lifetime", and it was included in Top 10 Algorithms of 20th Century by the IEEE magazine Computing in Science & Engineering.

The best-known FFT algorithms depend upon the factorization of N, but there are FFTs with $O(N \log N)$ complexity for all *N*, even for prime *N*. Many FFT algorithms only depend on the fact that $e^{-2\pi j/N}$ is an *N*-th primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it.



James William Cooley (born 1926, died June 29, 2016) was an American mathematician. Cooley received a B.A. degree in 1949 from Manhattan College, Bronx, NY, an M.A. degree in 1951 from Columbia University, New York, NY, and a Ph.D. degree in 1961 in applied mathematics from Columbia University. He was a programmer on John von Neumann's computer at the Institute for Advanced Study, Princeton, NJ, from 1953 to 1956. He worked on quantum mechanical computations at the Courant Institute, New York University, from 1956 to 1962, when he joined the Research Staff at the IBM Watson Research Center, Yorktown Heights, NY. Upon retirement from IBM in 1991, he joined the Department of Electrical Engineering, University of Rhode Island, Kingston, where he served on the faculty of the computer engineering program.

His most significant contribution to the world of mathematics and digital signal processing is the Fast Fourier transform, which he co-developed with John Tukey (see Cooley–Tukey FFT algorithm) while working for the research division of IBM in 1965.

The motivation for it was provided by Dr. Richard L. Garwin at IBM Watson Research who was concerned about verifying a Nuclear arms treaty with the Soviet Union for the SALT talks. Garwin thought that if he had a very much faster Fourier Transform he could plant sensors in the ground in countries surrounding the Soviet Union. He suggested the idea of how Fourier transforms could be programmed to be much faster to both Cooley and Tukey. They did the work, the sensors were planted, and he was able to locate nuclear explosions to within 15 kilometers of where they were occurring.

J. W. Cooley was a member of the Digital Signal Processing Committee of the IEEE, was elected a Fellow of IEEE for his work on FFT, and received the IEEE Centennial Medal. In 2002 he received the IEEE Jack S. Kilby Signal Processing Medal. He considerably contributed to the establishing of terminology in digital signal processing.

Publications


1. James W. Cooley (1961): "An improved eigenvalue corrector formula for solving the Schrödinger equation for central fields", Math. Comput. 15, 363–374. DOI: 10.1090/S0025-5718-1961-0129566-X This describes the so-called Numerov-Cooley method for numerically solving one-dimensional Schrödinger equations.
2. James W. Cooley & John W. Tukey (1965): "An algorithm for the machine calculation of complex Fourier series", Math. Comput. 19, 297–301.
3. Real, Edward C., Donald W. Tufts and James W. Cooley. "Two Algorithms for Fast Approximate Subspace Tracking." IEEE Transactions on Signal Processing. 47(7):1936–1945. July 1999.