Article

Historical Journey of the Fast Fourier Transform –From Enlightenment Europe via Bletchley Park to the Modern World

Keith Jones *

Article

Historical Journey of the Fast Fourier Transform – From Enlightenment Europe via Bletchley Park to the Modern World

Keith John Jones

Consultant Mathematician (Retired), Weymouth, Dorset, UK; keith.jones8@talktalk.net

Abstract: *The emergence of the semiconductor industry in the early 1960s resulted in a significant stage in the evolution of computing when large computational problems, as typified by the application of the discrete Fourier transform (DFT) to the task of spectrum estimation, could suddenly, with the availability of suitable algorithms, be solved in close to real time fashion. This paper provides a brief and meandering account of the history of the DFT's various solutions, referred to generically as the fast Fourier transform (FFT), the algorithm being chosen for its mathematical elegance, practical significance and ever-increasing range of applications. We touch upon a few of the most striking personalities, places and events that were encountered along the way and look, in particular, at the recent British contribution to the journey.*

Keywords: DFT; FFT; history

The subject of spectrum analysis started in earnest with the work of Joseph Fourier (1768-1830), who asserted and proved that an arbitrary function could be transformed into a sum of trigonometric functions. It seems likely, however, that such ideas were already common knowledge amongst European mathematicians by the time Fourier appeared on the scene, mainly through the earlier work of Leonard Euler (1707-1783) and Joseph Louis Lagrange (1736-1813), with the first appearance of the discrete version of this transformation, the discrete Fourier transform (DFT), dating back to Euler's investigations of sound propagation in elastic media in 1750 and to the astronomical work of Alexis Claude Clairut (1713-1765) in 1754 [1]. Today, due to the suitability of the frequency domain for convolution-based filtering operations, the DFT plays a central role in many disciplines, including signal/image processing and computer vision.

Turning to its definition, the DFT is a unitary transform [2], which for the case of N input/output samples, may be expressed in normalized form via the equation

$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] . W_N^{nk}$$

for k = 0, 1, … , N-1 where the transform kernel – also known as the *Fourier matrix* – derives from the term

$$W_N = \exp(-i2\pi/N), \qquad i = \sqrt{-1},$$

the primitive N^{th} complex root of unity [2]. The direct computation of the DFT involves $O(N^2)$ arithmetic operations, so that many of the early scientific problems involving it's use could not be seriously attacked without access to fast algorithms for its efficient solution. The key to designing such algorithms is the identification and exploitation of *symmetries* – through the application of suitable *number theoretic* and/or *algebraic* techniques – whether the symmetries exist in just the transform kernel or the input/output data as well.

One early area of activity for the DFT involved astronomical calculations and in the early part of the 19th century Carl Friedrich Gauss (1777-1855) used it for the interpolation of asteroidal orbits from equally-spaced sets of observations [1]. He developed a fast two-factor algorithm for its

computation that was identical to that described in 1965 by James Cooley and John Tukey [3] – as with many of Gauss's greatest ideas, however, the algorithm wasn't published outside of his collected works and only then in an obscure Latin form. This algorithm, which for a transform of length $N=N_1 \times N_2$ involves just $O((N_1+N_2) \times N)$ arithmetic operations, was probably the first member of the class of algorithms now commonly referred to as the fast Fourier transform (FFT) [4], which is unquestionably the most ubiquitous algorithm in use today for the analysis of digital data. In fact, Gauss is known to have first used the above- mentioned algorithm as far back as 1805, the same year that Admiral Nelson routed the French fleet at the Battle of Trafalgar – interestingly, Fourier served in Napoleon Bonaparte's army from 1798 to 1801, during its invasion of Egypt, acting as scientific advisor.

Pioneering work was later carried out in the early 20th century by the German mathematician Carl Runge [5], who in 1903 recognized that the periodicity of the DFT kernel could be exploited to enable the computation of a 2N-point DFT to be reduced to that of two N-point DFTs, this factorization technique being subsequently referred to as the *doubling algorithm*. The Cooley-Tukey algorithm, which does not rely upon any specific factorization of the transform length, may thus be viewed as a simple generalization of this algorithm, as successive application of the doubling algorithm leads directly to the radix-2 version of the Cooley-Tukey algorithm which involves just $O(N \times \log_2 N)$ arithmetic operations. Runge's influential work was subsequently picked up and popularized in publications by Karl Stumpff [6] in 1939 and Gordon Danielson and Cornelius Lanczos [7] in 1942, each in turn making contributions of their own to the subject. Danielson and Lanczos, for example, produced reduced-complexity solutions by exploiting symmetries in the transform kernel, whilst Stumpff discussed versions of both the doubling and the related *tripling algorithm*.

All the techniques developed, including those of more recent origin such as the *nesting algorithm* of Schmuel Winograd [8] from 1975 (which exploits his own minimum-complexity small-DFT routines) and the *split-radix algorithm* of Pierre Duhamel [9] from 1984, rely upon the familiar *divide-and-conquer* principle, whereby the computation of a composite-length DFT is broken down into that of a number of smaller DFTs where the small-DFT lengths correspond to the factors of the original transform length. Depending upon the particular factorization of the transform length, this process may be repeated in a recursive fashion on the increasingly smaller DFTs.

The class of fast algorithms based upon the decomposition of a composite-length DFT into smaller DFTs whose lengths have common factors – such as the Cooley-Tukey algorithm – is known as the *common factor algorithm* (CFA) [4]. When the small-DFT lengths are relatively prime [2], however, it becomes known as the *prime factor algorithm* (PFA) [4]. The CFA requires that the intermediate results obtained between successive stages of small DFTs be multiplied by elements of the Fourier matrix, these terms commonly referred to as *twiddle factors*. For the case of the PFA, however, there is no such requirement as the resulting twiddle factors each equate to one. This attractive algorithm, which greatly extends the range of admissible transform lengths, arose through the development of a new data re-ordering scheme in 1958 by statistician Jack Good [10]. The scheme is based upon the ubiquitous *Chinese Remainder Theorem* [11] – which provides a means of obtaining a unique solution to a set of simultaneous linear congruences – whose origins supposedly date back to the mathematician Sun Zi from the 1st century A.D. [11].

With regard to the various developments in FFT design that have taken place, it is the names of Cooley and Tukey that are invariably mentioned in any historical account. This does not really do justice, however, to the many previous contributions whose work was simply not picked up on, or appreciated, at the time of publication. The prime reason for such a situation was the lack of a suitable technology for their efficient implementation, this remaining the case until the arrival of the semiconductor revolution.

We return now to Good as his background is a fascinating one for anyone interested in the history of computing. During World War II he served at Bletchley Park [12], in Buckinghamshire, working alongside Alan Turing on, amongst other things, the decryption of messages produced by the *Enigma* machine [12] – as used by the German armed forces. At the same time, on the same site, a

team of engineers under the leadership of Tom Flowers – all seconded from the Post Office Research Establishment at Dollis Hill in North London – were, unbeknown to the outside world, developing the world's first electronic computer, the *Colossus* [13], under the supervision of Turing and Cambridge mathematician Max Newman. The Colossus was built primarily to automate various essential code breaking tasks such as the cracking of the Lorenz code used by Adolf Hitler to communicate with his generals.

Jack Good – later to work with Turing and Newman at Manchester University on a successor to Colossus, the Manchester Mark I [14] – therefore had direct involvement not only with the development of the FFT but also with that of the very technology that makes it such a crucially important algorithm today. The Colossus was the first serious device – albeit a very large and specialized one – on the path towards our current state of technology whereby the FFT may be mapped onto ever-smaller silicon devices for use in an ever-increasing number of applications ranging from astronomy and optics to mobile communications and cryptography. However, the parallel computing capabilities resulting from the recent technological innovations has necessitated a fundamental change in how we assess the *complexity* of the FFT, with straightforward arithmetic operation counts being replaced by new metrics such as those discussed by Jones [15] relating to power consumption or computational density.

Thus, the journey of the FFT has been traced from its origins in Enlightenment Europe, via an unusual connection at Bletchley Park, to its unique place in the technological world of today. Moving into the future, *intelligent* FFT designs able to exploit the increasing levels of parallelism [16] made available by modern computing technology – such as that provided by the field-programmable gate array (FPGA) [17] – might enable one, for example, to: a) handle large *multi-dimensional transforms*, as typically required for the continuous real-time processing of multi-dimensional images; or, with the potential opportunities offered by those quantum computing [18] technologies currently being developed, to b) assist with implementing *Shor's algorithm* [19] (concerning the factorization of 'extremely' large integers) on a quantum computer, thus offering the potential for breaking existing *public-key cryptography* schemes [20].

With problems such as these to address, both researchers and practitioners alike should find much to delight. The key, underlying requirement, in achieving success in these endeavors will be the effective application of mathematic techniques, both old and new:

> **"For the things of this world cannot be made known without a knowledge of mathematics"**

Roger Bacon (1214-1294), English philosopher & scientist [21].

Conflicts of Interest: The author states that there are no conflicts of interest.

References

1. Heideman, M., Johnson, D.H. & Burrus, C.S. (1984) *Gauss and the History of the Fast Fourier Transform*, IEEE ASSP Magazine, Vol. 1, pp. 14-21.
2. Birkhoff, G. & MacLane, S. (1977) *A Survey of Modern Algebra*, MacMillan.
3. Cooley, J.W. & Tukey, J.W. (1965) *An Algorithm for the Machine Calculation of Complex Fourier Series*, Mathematics of Computation, Vol. 19, No. 4, pp. 297-301.
4. Brigham, E.O. (1988) *The Fast Fourier Transform and its Applications*, Englewood Cliffs, Prentice Hall.
5. Runge, C. (1903) *Uber die Zerlegung Empirisch Periodischer Funktioner in Sinnus-Wellen*, Zeit. Fur Math. und Physik, Vol. 48, pp. 443-456.
6. Stumpff, K. (1939) *Tafeln und Aufgaben zur Harmonischer Analyse und Periodogramm- rechnung*, Berlin, Julius Springer.
7. Danielson, G.C. & Lanczos, C. (1942) *Some Improvements in Practical Fourier Series and their Application to X-Ray Scattering from Liquids*, J. Franklin Inst., Vol. 233, pp. 365-380 & 435-452.
8. Winograd, S. (1980) *Arithmetic Complexity of Computations*, SIAM.
9. Duhamel, P. (1986) *Implementations of Split-Radix FFT Algorithms for Complex, Real and Real-Symmetric Data*, IEEE Trans. ASSP, Vol. 34, No. 2, pp. 285-295.
10. Good, I.J. (1958) *The Interaction Algorithm and Practical Fourier Series*, J. Royal Statistical Society, Ser. B, Vol. 20, pp. 361-372.

11. Ding, C., Pei, D. & Salomaa, A. (1996) *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*, World Scientific.
12. McKay, S. (2010) *The Secret Life of Bletchley Park*, Aurum.
13. Gannon, P. (2006) *Colossus: Bletchley Park's Greatest Secret*, Atlantic Books, London.
14. Lavington, S. (Ed.) (2012) *Alan Turing and His Contemporaries: Building the World's First Computers*, BCS (Chartered Institute for IT).
15. Jones, K.J. (2022) *The Regularized Fast Hartley Transform: Low-Complexity Parallel Computation of the FHT in One and Multiple Dimensions*, Springer.
16. Akl, G. (1989) *The Design and Analysis of Parallel Algorithms*, Prentice-Hall.
17. Maxfiled, C. (2004) *The Design Warrior's Guide to FPGAs*, Elsevier.
18. Bernhardt, C. (2019) *Quantum Computing for Everyone*, MIT Press.
19. Shor, P. (1994) *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, Proc. 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Science Press.
20. Hankerson, D, Menezes, A & Vanstone, S. (2004) *Guide to Elliptic Curve Cryptography*, Springer.
21. Zagorin, P. (1999) *Francis Bacon*, Princeton.