FFTs of prime length and Rader's conversion*

Douglas L. Jones

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

FFTs of prime length can be computed efficiently via Rader's conversion, via the chirp z-transform, or by use of Winograd methods.

The power-of-two FFT algorithms[1], such as the radix-2[2] and radix-4[3] FFTs, and the common-factor[4] and prime-factor[5] FFTs, achieve great reductions in computational complexity of the DFT[6] when the length, N, is a composite number. DFTs of prime length are sometimes needed, however, particularly for the short-length DFTs in common-factor or prime-factor algorithms. The methods described here, along with the composite-length algorithms, allow fast computation of DFTs of **any** length.

There are two main ways of performing DFTs of prime length:

1. Rader's conversion, which is most efficient, and the
2. Chirp-z transform[7], which is simpler and more general.

Oddly enough, both work by turning prime-length DFTs into convolution! The resulting convolutions can then be computed efficiently by either

1. fast convolution[8] via composite-length FFTs (simpler) or by
2. Winograd techniques (more efficient)

1 Rader's Conversion

Rader's conversion is a one-dimensional index-mapping[9] scheme that turns a length-NDFT[10] (N prime) into a length-($N-1$) convolution and a few additions. Rader's conversion works **only** for prime-length N.

*Version 1.3: Sep 24, 2006 8:38 am -0500

†http://creativecommons.org/licenses/by/1.0

[1]"Power-of-two FFTs" <http://cnx.org/content/m12059/latest/>

[2]"Decimation-in-time (DIT) Radix-2 FFT" <http://cnx.org/content/m12016/latest/>

[3]"Radix-4 FFT Algorithms" <http://cnx.org/content/m12027/latest/>

[4]"Multidimensional Index Maps" <http://cnx.org/content/m12025/latest/>

[5]"The Prime Factor Algorithm" <http://cnx.org/content/m12033/latest/>

[6]"DFT Definition and Properties" <http://cnx.org/content/m12019/latest/>

[7]"Chirp-z Transform" <http://cnx.org/content/m12013/latest/>

[8]"Fast Convolution" <http://cnx.org/content/m12022/latest/>

[9]"Multidimensional Index Maps" <http://cnx.org/content/m12025/latest/>

[10]"Spectrum Analysis Using the Discrete Fourier Transform", (3) <http://cnx.org/content/m12032/latest/#DFTequation>

An **index map** simply rearranges the order of the sum operation in the DFT definition[11]. Because addition is a commutative operation, the same mathematical result is produced from any order, as long as all of the same terms are added once and only once. (This is the condition that defines an index map.) Unlike the multi-dimensional index maps[12] used in deriving common factor[13] and prime-factor FFTs[14], Rader's conversion uses a one-dimensional index map in a finite group of N integers: $k = r^m modN$

1.1 Fact from number theory

If N is prime, there exists an integer "r" called a **primitive root**, such that the index map $k = r^m modN$, $m = [0, 1, 2, \ldots, N-2]$, uniquely generates all elements $k = [1, 2, 3, \ldots, N-1]$

Example 1
$N = 5$, $r = 2$

$$2^0 mod5 = 1$$

$$2^1 mod5 = 2$$

$$2^2 mod5 = 4$$

$$2^3 mod5 = 3$$

1.2 Another fact from number theory

For N prime, the inverse of r (i.e. $r^{-1} rmodN = 1$ is also a primitive root (call it r^{-1}).

Example 2
$N = 5$, $r = 2 r^{-1} = 3$

$$2 \times 3 mod5 = 1$$

$$3^0 mod5 = 1$$

$$3^1 mod5 = 3$$

$$3^2 mod5 = 4$$

$$3^3 mod5 = 2$$

So why do we care? Because we can use these facts to turn a DFT[15] into a convolution!

[11] "DFT Definition and Properties" <http://cnx.org/content/m12019/latest/>
[12] "Multidimensional Index Maps" <http://cnx.org/content/m12025/latest/>
[13] "Multidimensional Index Maps" <http://cnx.org/content/m12025/latest/>
[14] "The Prime Factor Algorithm" <http://cnx.org/content/m12033/latest/>
[15] "Spectrum Analysis Using the Discrete Fourier Transform", (3) <http://cnx.org/content/m12032/latest/#DFTequation>

1.3 Rader's Conversion

Let $\forall \mathrm{mn}, (m = [0, 1, \ldots, N-2]) \ \wedge \ n \in [1, 2, \ldots, N-1] : (n = r^{-m} mod N)$, $\forall \mathrm{pk}, (p = [0, 1, \ldots, N-2]) \ \wedge \ k \in [1, 2, \ldots, N-1] : (k = r^{p} mod N)$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \begin{cases} x(0) + \sum_{n=1}^{N-1} x(n) W_N^{nk} & \text{if} \quad k \neq 0 \\ \sum_{n=0}^{N-1} x(n) & \text{if} \quad k = 0 \end{cases}$$

where for convenience $W_N^{nk} = e^{-\left(i \frac{2\pi nk}{N}\right)}$ in the DFT equation. For $k \neq 0$

$$\begin{aligned} X(r^p mod N) &= \sum_{m=0}^{N-2} x(r^{-m} mod N) W^{r^p r^{-m}} + x(0) \\ &= \sum_{m=0}^{N-2} x(r^{-m} mod N) W^{r^{p-m}} + x(0) \\ &= x(0) + x\left(r^{-l} mod N\right) * W^{r^l} \end{aligned} \tag{1}$$

where $l = [0, 1, \ldots, N-2]$

Example 3
$N = 5$, $r = 2$, $r^{-1} = 3$

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 1 & 3 \\ 0 & 3 & 1 & 4 & 2 \\ 0 & 4 & 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \end{pmatrix}$$

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(4) \\ X(3) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 4 & 2 \\ 0 & 2 & 1 & 3 & 4 \\ 0 & 4 & 2 & 1 & 1 \\ 0 & 3 & 4 & 2 & 3 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(3) \\ x(4) \\ x(2) \end{pmatrix}$$

where for visibility the matrix entries represent only the **power**, m of the corresponding DFT term W_N^m Note that the 4-by-4 circulant matrix[16]

$$\begin{pmatrix} 1 & 3 & 4 & 2 \\ 2 & 1 & 3 & 4 \\ 4 & 2 & 1 & 1 \\ 3 & 4 & 2 & 3 \end{pmatrix}$$

corresponds to a length-4 circular convolution.

Rader's conversion turns a prime-length DFT[17] into a few adds and a **composite-length** $(N-1)$ circular convolution, which can be computed efficiently using either

1. fast convolution[18] via FFT and IFFT

[16] http://en.wikipedia.org/wiki/Circulant_matrix
[17] "Spectrum Analysis Using the Discrete Fourier Transform", (3) <http://cnx.org/content/m12032/latest/#DFTequation>
[18] "Fast Convolution" <http://cnx.org/content/m12022/latest/>

2. index-mapped convolution algorithms and short Winograd convolution alogrithms. (Rather complicated, and trades fewer multiplies for many more adds, which may not be worthwile on most modern processors.) See *R.C. Agarwal and J.W. Cooley*[1]

2 Winograd minimum-multiply convolution and DFT algorithms

S. Winograd has proved that a length-N circular or linear convolution or DFT[19] requires less than $2N$ multiplies (for real data), or $4N$ real multiplies for complex data. (This doesn't count multiplies by rational fractions, like 3 or $\frac{1}{N}$ or $\frac{5}{17}$, which can be computed with additions and one overall scaling factor.) Furthermore, Winograd showed how to construct algorithms achieving these counts. Winograd prime-length DFTs and convolutions have the following characteristics:

1. Extremely efficient for small N ($N < 20$)
2. The number of adds becomes **huge** for large N.

Thus Winograd's minimum-multiply FFT's are useful only for small N. They are very important for Prime-Factor Algorithms[20], which generally use Winograd modules to implement the short-length DFTs. Tables giving the multiplies and adds necessary to compute Winograd FFTs for various lengths can be found in *C.S. Burrus (1988)*[2]. Tables and FORTRAN and TMS32010 programs for these short-length transforms can be found in *C.S. Burrus and T.W. Parks (1985)*[3]. The theory and derivation of these algorithms is quite elegant but requires substantial background in number theory and abstract algebra. Fortunately for the practitioner, all of the short algorithms one is likely to need have already been derived and can simply be looked up without mastering the details of their derivation.

3 Winograd Fourier Transform Algorithm (WFTA)

The Winograd Fourier Transform Algorithm (WFTA) is a technique that recombines the short Winograd modules in a prime-factor FFT[21] into a composite-N structure with fewer multiplies but more adds. While theoretically interesting, WFTAs are complicated and different for every length, and on modern processors with hardware multipliers the trade of multiplies for many more adds is very rarely useful in practice today.

References

[1] R.C. Agarwal and J.W. Cooley. New algorithms for digital convolution. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 25:392–410, Oct 1977.

[2] C.S. Burrus. Efficient fourier transform and convolution algorithms. In J.S. Lin and A.V. Oppenheim, editors, *Advanced Topics in Signal Processing*, chapter Chapter 4. Prentice-Hall, 1988.

[3] C.S. Burrus and T.W. Parks. *DFT/FFT and Convolution Algorithms*. Wiley-Interscience, 1985.

[19] "Spectrum Analysis Using the Discrete Fourier Transform", (3) <http://cnx.org/content/m12032/latest/#DFTequation>
[20] "The Prime Factor Algorithm" <http://cnx.org/content/m12033/latest/>
[21] "The Prime Factor Algorithm" <http://cnx.org/content/m12033/latest/>