



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikipedia store

Interaction  
Help  
About Wikipedia  
Community portal  
Recent changes  
Contact page

Tools  
What links here  
Related changes  
Upload file  
Special pages  
Permanent link  
Page information  
Wikidata item  
Cite this page


Print/export  
Create a book  
Download as PDF  
Printable version

Languages   
فارسی  Edit links

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [More](#) ▾



# Rader's FFT algorithm

From Wikipedia, the free encyclopedia

**Rader's algorithm** (1968) is a [fast Fourier transform](#) (FFT) algorithm that computes the [discrete Fourier transform](#) (DFT) of [prime](#) sizes by re-expressing the DFT as a cyclic [convolution](#) (the other algorithm for FFTs of prime sizes, [Bluestein's algorithm](#), also works by rewriting the DFT as a convolution).

Since Rader's algorithm only depends upon the periodicity of the DFT kernel, it is directly applicable to any other transform (of prime order) with a similar property, such as a [number-theoretic transform](#) or the [discrete Hartley transform](#).

The algorithm can be modified to gain a factor of two savings for the case of DFTs of real data, using a slightly modified re-indexing/permutation to obtain two half-size cyclic convolutions of real data (Chu & Burrus, 1982); an alternative adaptation for DFTs of real data, using the discrete Hartley transform, was described by Johnson & Frigo (2007).

Winograd extended Rader's algorithm to include prime-power DFT sizes *p<sup>m</sup>* (Winograd 1976; Winograd 1978), and today Rader's algorithm is sometimes described as a special case of [Winograd's FFT algorithm](#), also called the *multiplicative Fourier transform algorithm* (Tolimieri et al., 1997), which applies to an even larger class of sizes. However, for [composite](#) sizes such as prime powers, the [Cooley–Tukey FFT algorithm](#) is much simpler and more practical to implement, so Rader's algorithm is typically only used for large-prime [base cases](#) of Cooley–Tukey's [recursive](#) decomposition of the DFT (Frigo and Johnson, 2005).

## Algorithm [\[edit\]](#)

Recall that the DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk} \quad k = 0, \dots, N-1.$$

If *N* is a prime number, then the set of non-zero indices *n* = 1,...,*N*−1 forms a [group](#) under multiplication [modulo](#) *N*. One consequence of the [number theory](#) of such groups is that there exists a [generator](#) of the group (sometimes called a [primitive root](#)), an integer *g* such that *n* = *g<sup>q</sup>* (mod *N*) for any non-zero index *n* and for a unique *q* in 0,...,*N*−2 (forming a [bijection](#) from *q* to non-zero *n*). Similarly *k* = *g<sup>p</sup>* (mod *N*) for any non-zero index *k* and for a unique *p* in 0,...,*N*−2, where the negative exponent denotes the [multiplicative inverse](#) of *g<sup>p</sup>* modulo *N*. That means that we can rewrite the DFT using these new indices *p* and *q* as:

$$\begin{aligned} X_0 &= \sum_{n=0}^{N-1} x_n, \\ X_{g^{-p}} &= x_0 + \sum_{q=0}^{N-2} x_{g^q} e^{-\frac{2\pi i}{N}g^{-(p-q)}} \quad p = 0, \dots, N-2. \end{aligned}$$

(Recall that *x<sub>n</sub>* and *X<sub>k</sub>* are implicitly periodic in *N*, and also that *e<sup>2πi</sup>*=1. Thus, all indices and exponents are taken modulo *N* as required by the group arithmetic.)

The final summation, above, is precisely a cyclic convolution of the two sequences *a<sub>q</sub>* and *b<sub>q</sub>* of length *N*−1 (*q* = 0,...,*N*−2) defined by:

$$\begin{aligned} a_q &= x_{g^q} \\ b_q &= e^{-\frac{2\pi i}{N}g^{-q}}. \end{aligned}$$

## Evaluating the convolution [\[edit\]](#)


Since *N*−1 is composite, this convolution can be performed directly via the [convolution theorem](#) and more conventional FFT algorithms. However, that may not be efficient if *N*−1 itself has large prime factors, requiring recursive use of Rader's algorithm. Instead, one can compute a length-(*N*−1) cyclic convolution exactly by zero-padding it to a length of at least 2(*N*−1)−1, say to a [power of two](#), which can then be evaluated in *O*(*N* log *N*) time without the recursive application of Rader's algorithm.

This algorithm, then, requires *O*(*N*) additions plus *O*(*N* log *N*) time for the convolution. In practice, the *O*(*N*)

additions can often be performed by absorbing the additions into the convolution: if the convolution is performed by a pair of FFTs, then the sum of  $x_n$  is given by the DC (0th) output of the FFT of  $a_q$  plus  $x_0$ , and  $x_0$  can be added to all the outputs by adding it to the DC term of the convolution prior to the inverse FFT. Still, this algorithm requires intrinsically more operations than FFTs of nearby composite sizes, and typically takes 3–10 times as long in practice.

If Rader's algorithm is performed by using FFTs of size  $N-1$  to compute the convolution, rather than by zero padding as mentioned above, the efficiency depends strongly upon  $N$  and the number of times that Rader's algorithm must be applied recursively. The worst case would be if  $N-1$  were  $2N_2$  where  $N_2$  is prime, with  $N_2-1 = 2N_3$  where  $N_3$  is prime, and so on. In such cases, supposing that the chain of primes extended all the way down to some bounded value, the recursive application of Rader's algorithm would actually require  $O(N^2)$  time. Such  $N_j$  are called [Sophie Germain primes](#), and such a sequence of them is called a [Cunningham chain](#) of the first kind. The lengths of Cunningham chains, however, are observed to grow more slowly than  $\log_2(N)$ , so Rader's algorithm applied in this way is probably not  $\Omega(N^2)$ , though it is possibly worse than  $O(N \log N)$  for the worst cases. Fortunately, a guarantee of  $O(N \log N)$  complexity can be achieved by zero padding.

## References [\[edit\]](#)

- C. M. Rader, "Discrete Fourier transforms when the number of data samples is prime," *Proc. IEEE* **56**, 1107–1108 (1968).
- S. Chu and C. Burrus, "A prime factor FTT [sic] algorithm using distributed arithmetic," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **30** (2), 217–227 (1982).
- Matteo Frigo and Steven G. Johnson, "[The Design and Implementation of FFTW3](#) , " *Proceedings of the IEEE* **93** (2), 216–231 (2005).
- S. Winograd, "On Computing the Discrete Fourier Transform", *Proc. National Academy of Sciences USA*, **73**(4), 1005–1006 (1976).
- S. Winograd, "On Computing the Discrete Fourier Transform", *Mathematics of Computation*, **32**(141), 175–199 (1978).
- R. Tolimieri, M. An, and C.Lu, "Algorithms for Discrete Fourier Transform and Convolution," Springer-Verlag, 2nd ed., 1997.

Categories: [FFT algorithms](#)

This page was last modified on 21 June 2015, at 18:36.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

