



THE FAST FOURIER TRANSFORM FOR EXPERIMENTALISTS, PART II: CONVOLUTIONS

By Denis Donnelly and Bert Rust

WHEN UNDERGRADUATE STUDENTS FIRST COMPUTE A FAST FOURIER TRANSFORM (FFT), THEIR INITIAL IMPRESSION IS OFTEN A BIT MISLEADING. THE PROCESS ALL

seems so simple and transparent: the software takes care of the computations, and it's easy to create the plots. But once they start probing, students quickly learn that like any rich scientific expression, the implications, the range of applicability, and the associated multilevel understandings needed to fully appreciate the subtleties involved take them far beyond the basics. Even professionals find surprises when performing such computations, becoming aware of details that they might not have fully appreciated until they asked more sophisticated questions.

In the first of this five-part series,¹ we discussed several basic properties of the FFT. In addition to some fundamental elements, we treated zero-padding, aliasing, and the relationship to a Fourier series, and ended with an introduction to windowing. In this article, we'll briefly look at the convolution process.

Convolution

Convolution, a process some would say lies at the heart of digital signal processing, involves two functions, which we'll call $x(t)$ and $b(t)$, where $x(t)$, for example, could be an input signal and $b(t)$ some linear system's impulse response. When convolved, \otimes , they yield an output function $y(t)$. The process expresses the amount of one function's overlap as it is shifted over the other, providing a kind of blending of the two functions:

$$y(t) = x(t) \otimes b(t). \quad (1)$$

This process has many applications. Filtering is one example: given the appropriate impulse response, we can create any one of a number of filters. We'll give some examples in the next section, but we'll postpone further information

about filtering and detrending until the next installment. Correlation is another closely related process and can help determine if a particular signal occurs in another datastream.

Deconvolution is the reverse: in effect, it uses the process itself to remove the effects of an undesired convolution or data distortion. When taking data, a convolution can obscure the desired information, perhaps due to interfering physical interactions or by the detection system itself (which has its own response). A gamma ray arriving at a detector, for example, has a well-defined energy, yet the detector output shows several associated effects related to the interaction of the gamma ray with a crystal. If a nuclear physicist is interested in the gamma ray's energy or intensity instead of the detector's response, then he or she needs to know how to extract the appropriate information from this much larger signal set. Deconvolving can remove the detector response, restoring the data to a form closer to the original.

When noise accompanies a signal, as it always does to some extent, a direct deconvolution can generate unstable results, which renders the process unusable. One way to reduce the noise's influence is to assume that analytic functions can represent either (or both) the original signal and the convoluted signal. When such a representation is possible, the chances of success with the deconvolution process greatly improve. Still, deconvolution is beyond the scope of this series, so we won't discuss it here.

The continuous convolution is defined as

$$y(t) = x(t) \otimes b(t) = \int_{-\infty}^{\infty} x(\tau)b(t-\tau)d\tau = \int_{-\infty}^{\infty} x(t-\tau)b(\tau)d\tau. \quad (2)$$

In his book on the FFT, E. Oran Brigham states that "Possibly the most important and powerful tool in modern scientific analysis is the relationship between [Equation 2] and its Fourier transform."² The relationship referred to is the time-convolution theorem:

$$\mathcal{F}\{x(t) \otimes b(t)\} = \mathcal{F}\{x(t)\} \times \mathcal{F}\{b(t)\} = X(f)H(f), \quad (3)$$

where \times denotes ordinary multiplication, and $X(f)$ and $H(f)$ are the continuous Fourier transforms of $x(t)$ and $b(t)$.

In real life, we seldom have access to the functions $x(t)$ and $b(t)$; instead, we have only finite time-series representations, such as

$$x_k = x(k \cdot \Delta t)$$

and

$$b_k = b(k \cdot \Delta t), k = 0, 1, 2, \dots, N-1. \quad (4)$$

Given this discrete representation, we can't compute $y(t)$ exactly, but we can compute a time-series approximation to it. Specifically, we can write an expression for the discrete convolution as

$$y_n = \Delta t \sum_{k=1}^{N-1} x_k \cdot b_{n-k} = \Delta t \sum_{k=1}^{N-1} x_{n-k} \cdot b_k$$

$$n = 0, 1, 2, \dots, N-1. \quad (5)$$

If the response function were the trivial example in which b_0 has the value 1 and all other b values are 0, then the convolution process would just reproduce the input signal (if b_0 differed from 1, it would scale the input signal proportionally to b_0). If all b 's were 0 except for b_m , then we would scale the input signal by the magnitude of b_m and delay it by m sample intervals. The convolution process is the summation of such elements.

It's important to keep two details in mind when performing a convolution process: one, the two signals must have the same number of elements (zero-padding easily solves this problem), and two, the discrete convolution theorem treats the data as if it were periodic. We can express the summation associated with this *circular convolution* as

$$\sum_{k=0}^{N-1} x[(n-k) \bmod N] b(k). \quad (6)$$

This cyclic effect causes a wraparound problem that we'll explain in more detail later.

The FFT form of the convolution of two time series is given by

$$x \otimes b = \text{ifft}(\text{fft}(x) \times \text{fft}(b)), \quad (7)$$

where the product of the two transforms is element by element and *ifft* stands for inverse FFT. (While we're discussing convolution in the time domain and multiplication in the frequency domain, we should mention that an interchange of roles is also possible. Multiplication in the time domain corresponds to convolution in the frequency domain.)

We can readily program the summation required to compute a convolution: as the number of data points increases, the computational advantage goes to the convolution's implementation with FFT, even though it requires several steps. The reason is that a convolution in the time domain requires N^2 multiplications whereas the computational cost of taking the FFT route is on the order of $3N \log_2(N)$ multiplications. Despite the fact that three steps are involved, for large N , the advantages of the FFT approach are unmistakable. Even for the very modest case of $N = 250$, using FFTs to compute a convolution is already more than 10 times faster than the time-domain computation.

One way to implement the summation shown in Equation 6 is by expressing the equation itself in matrix form. Create an $N \times N$ matrix in which the first column takes on the x -values from x_0 to x_{N-1} . Let the next column take on the same x -values but shifted down one row, with the last value becoming the first, and repeat this rolling procedure for each successive column. Multiplying this x -matrix by the b -vector yields a circular convolution. We get a *linear convolution* from this same multiplication if we set all the terms in the x -matrix above the diagonal to zero.

To avoid the wraparound pitfall, we could do one of two things: compute the linear convolution (setting all elements above the x -matrix's diagonal to zero) or zero-pad the functions so that the total number of data points is at least $N_0 + K_0 - 1$, where N_0 and K_0 are the original numbers of data points in the functions x and b . With this number of elements, we avoid any distortion due to wraparound:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{N-1} \end{bmatrix}. \quad (8)$$

Examples

As an example of a linear convolution calculation, consider the signals

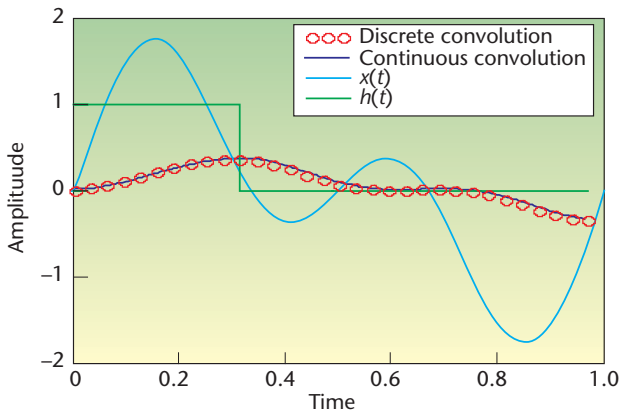


Figure 1. Comparison of continuous and discrete convolution calculations. We calculated the convolution of $x(t)$ and $h(t)$ in three ways: continuous and discrete in the time and frequency domains. The discrete convolution calculations approach the continuous form.

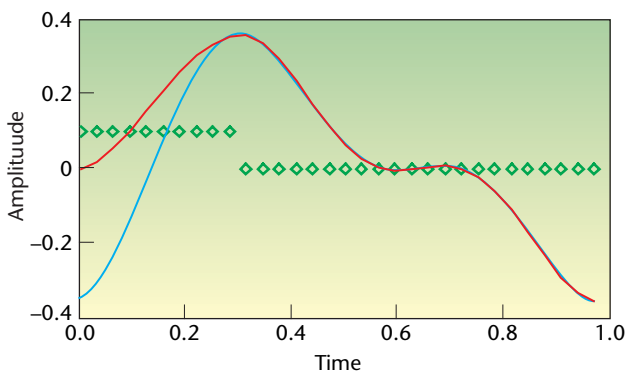


Figure 2. Convolution with and without wraparound distortions. The blue curve shows the circular form of the convolution without zero-padding. The red curve is based on a zero-padded calculation that avoids the distortion associated with circularity. The diamonds show the h response curve (scaled at 10 percent of true height); the width of the response function is associated with the region in which the circular convolution is spoiled.

$$x(t) = \begin{cases} \sin(2\pi t) + \sin(4\pi t), & 0 \leq t \leq 1, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

and

$$h(t) = \begin{cases} 1, & 0 \leq t \leq 0.3125, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

which we discretize to have 32 equally spaced points on the interval $[0,1]$.

Figure 1 shows the signal, the impulse response, and the

associated continuous and discrete convolutions. The discrete convolution as computed by taking the IFFT of the product of the FFTs of x and h is identical to that obtained via matrix multiplication.

Figure 2 shows the wraparound associated with the circular convolution example. The convolution is altered for the number of nonzero data points in h .

In Figure 3, we show the FFTs of the linear and circular convolutions. The FFT of the convolution resulting from the matrix multiplication is the same as the product of x and h 's FFTs. In the figure, we can see some frequency dependence associated with the convolution process. Figure 4 gives an overall summary of the operations and their interrelation.

For a more realistic example of convolution, let's look at the propagation of an acoustic pressure wave through a rectangular waveguide. The waveguide's resonant conditions restrict the wave numbers of the transverse wave components to discrete values, and the wave propagates only in certain modes. If we treat the waveguide as a linear device with an impulse response h , then we can predict the form of the transmitted signal by taking the convolution of our input signal x and the impulse response of the waveguide. Kristien Meykens and colleagues³ show that for modes other than $(0, 0)$, the impulse response departs from a δ -function in which the lower frequencies resemble a reversed chirp.

Figure 5 shows the convolution of an input signal consisting of a brief acoustic burst with the impulse response of a rectangular waveguide (which we represent as a chirp function). We form this input signal by multiplying an 8-kHz sine wave by a Bartlett (tent-shaped) window. The chirp function represents the impulse response for the waveguide's $(1, 0)$ mode, and $f(t) = 10^3 + 2 \cdot 10^6 t$ represents the chirp function's frequency dependence. The chirp expression is simply $\sin(\theta(t))$, where

$$\theta(t) = \int_0^t 2\pi f(t') dt'. \quad (11)$$

In general, a convolution shows the two functions' entanglement. The examples we've discussed here provide a clear instance in which we can see where the similarity between the input signal and the impulse response is the greatest. Such computations are in reasonable agreement with experimental results.³

In the next installment of this series, we'll continue to examine the problem of spectrum estimation with a discussion of the autocorrelation function and the correlogram estimates, which are based upon it.

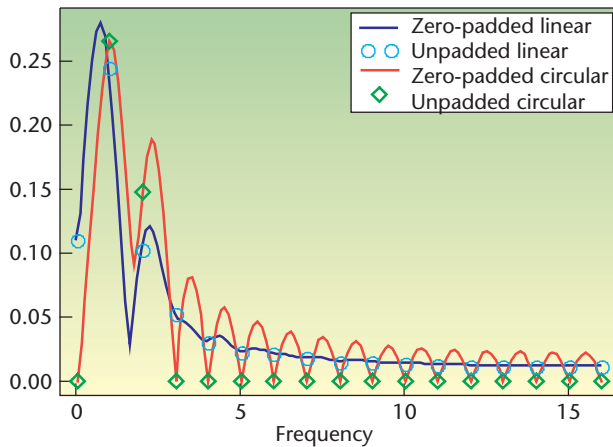


Figure 3. The FFTs of the linear and circular convolutions. The two curves are shown with (solid curves) and without (circles and diamonds) zero padding. We computed these FFTs from the convolution data for Figure 1's discrete transform. The results are the same as those obtained by taking the product of x and h 's FFTs.

References

1. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists Part I: Concepts," *Computing in Science & Eng.*, vol. 7, no. 2, 2005, pp. 80–88.
2. E. Oran Brigham, *The Fast Fourier Transform and Its Applications*, Prentice-Hall, 1988.
3. K. Meykens, B. Van Rompaey, and J. Janssen, "Dispersion in Acoustic Waveguide: A Teaching Laboratory Experiment," *Am. J. Physics*, vol. 67, no. 5, 1999, pp. 400–406.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly received a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Bert Rust is a mathematician at the US National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust received a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

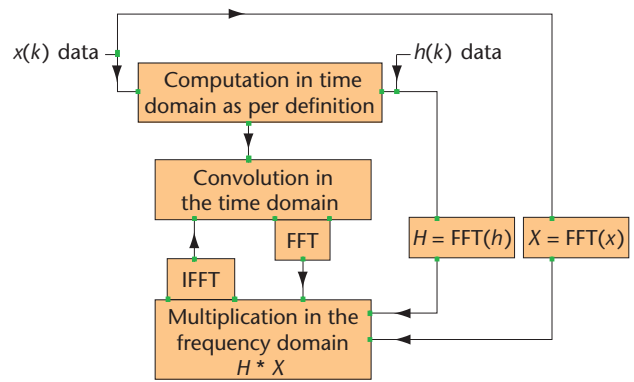


Figure 4. The interrelation between time and frequency domain operations that lead to convolution. Multiplying the FFT's of x and h followed by an IFFT also lead to the convolution. An FFT of the convolution would yield the same result as the product of the FFTs.

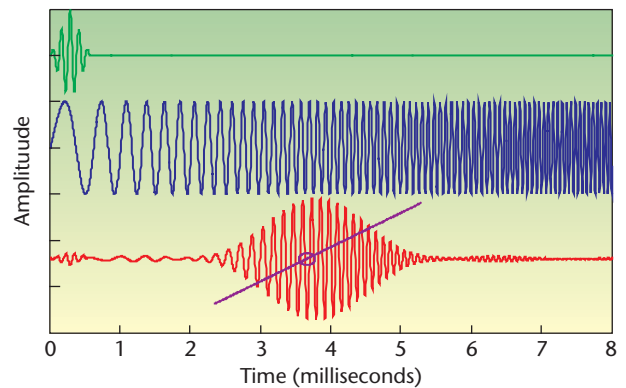


Figure 5. The convolution of a windowed sine wave burst and a chirp function. The top curve shows the input signal, and the middle curves show the impulse response of the waveguide (a chirp function). The chirp frequency increases linearly with time, ranging from roughly 1 kHz at $t = 0$ to roughly 17 kHz at $t = 8$ ms; the frequency increases at a rate of approximately 2 kHz/ms. The bottom curves show the convolution and the approximate frequencies associated with the most significant section of the convolution over time. The one marked point represents the frequency of the windowed sine curve which is 8 kHz. The slope of the line representing frequency is about 1.9 kHz/ms.

Submissions: Send one PDF copy of articles and/or proposals to Norman Chonacky, Editor in Chief, cise-editor@aip.org. Submissions should not exceed 6,000 words and 15 references. All submissions are subject to editing for clarity, style, and space.

Editorial: Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *CISE* does not necessarily constitute endorsement by the IEEE, the AIP, or the IEEE Computer Society.

Circulation: *Computing in Science & Engineering* (ISSN 1521-9615) is published bimonthly by the AIP and the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314, phone +1 714 821 8380; IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; AIP Circulation and Fulfillment Department, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Annual subscription rates for 2005: \$42 for Computer Society members (print only) and \$42 for AIP society members (print plus online). For more information on other subscription prices, see www.computer.org/subscribe/ or https://www.aip.org/forms/journal_catalog/order_form_fs.html. Computer Society back issues cost \$20 for members, \$96 for nonmembers; AIP back issues cost \$22 for members.

Postmaster: Send undelivered copies and address changes to *Computing in Science & Engineering*, 445 Hoes Ln., Piscataway, NJ 08855. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in the USA.

Copyright & reprint permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons those articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Dr., Danvers, MA 01923. For other copying, reprint, or republication permission, write to Copyright and Permissions Dept., IEEE Publications Administration, 445 Hoes Ln., PO Box 1331, Piscataway, NJ 08855-1331. Copyright © 2005 by the Institute of Electrical and Electronics Engineers Inc. All rights reserved.