LINEAR DISCRETE CONVOLUTION AND ITS INVERSE. PART 1. CONVOLUTION

Mircea Cirnu[1]

Abstract

We present here several ways of calculating the linear discrete convolution, its inverse - the deconvolution, by direct methods, generator functions, Z-transform, using matrices and MATLAB. These notions was used by author in a series of papers, especially for solve several types of equations.

Keywords: **complete and truncated linear discrete convolution and deconvolution.**
2000 Mathematics Subject Classification: **40-04, 40A05**

Introduction

The convolution is a fundamental concept in mathematics and applications. The use of the convolution and other related notions, as its inverse - the deconvolution, to solve several kinds of equations is particularly of a great importance and was considered by the author in a series of papers, [1] – [6]. They also play a fundamental role in the study of linear physical discrete systems, causal, time-invariant.

In this part of the paper, we present the notion of linear discrete convolution in complete and also in truncated form. We give here the algorithms for calculus of these notions. Are considered both direct algorithms and those based on Matlab, generator functions, Z-transform and matrices. Examples are included. Other algorithms for calculating discrete convolution are given in the book [8].

I. Complete convolution

1. Linear discrete convolution in complete form

We call *discrete convolution* (or *Cauchy product*) of two finite sequences of real or complex numbers, $a=(a_0,a_1,\ldots,a_m)$ and $b=(b_0,b_1,\ldots,b_n)$, of the lengths $m+1$ and $n+1$, the finite sequence

$$c=a*b=(c_0,c_1,\ldots,c_{m+n}) \tag{1}$$

of the length $m+n+1=(m+1)+(n+1)-1$, with the terms given by the relations

---

[1] Mircea I Cirnu, Dept. of Mathematics III, University "Politehnica" of Bucharest, Romania, email: cirnumircea@yahoo.com

$$c_0 = a_0 b_0,\, c_1 = a_1 b_0 + a_0 b_1, \ldots, c_k = \sum_{j=0}^{k} a_{k-j} b_j \,, \ldots, \qquad c_{m+n} = a_m b_n \,. \tag{2}$$

The convolution is commutative, associative, distributive with respect to the addition of the sequences and has the unit $\delta = (1,0,0,...,0)$. The addition and the multiplication with scalars of the sequences are the usual ones.

A simple direct algorithm for calculating the convolution product consists in writing the two factors on rows, one under the other, and in multiplying each terms of the first factor by every term of the second and writing these partial rows on the right, starting with position of the considered term of second factor. The convolution product will result by adding these partial rows.

Example. In the examples throughout this first part of paper, we consider sequences $a = (1,2,0,-1,1)$, $b = (1,3,-1,-2)$ and their convolution product $c = a * b$. The convolution can be calculated by the algorithm

$$\begin{array}{cccccccc}
1 & 2 & 0 & -1 & 1 & & & \\
1 & 3 & -1 & -2 & & & & \\
\hline
1 & 2 & 0 & -1 & 1 & & & \\
 & 3 & 6 & 0 & -3 & 3 & & \\
 & & -1 & -2 & 0 & 1 & -1 & \\
 & & & -2 & -4 & 0 & 2 & -2 \\
\hline
1 & 5 & 5 & -5 & -6 & 4 & 1 & -2
\end{array}$$

therefore is $c = a * b = (1,5,5,-5,-6,4,1,-2)$.

2. Computing complete convolution by generating function

It is called *generating function* (see [10]) of a finite sequence $a = (a_0, a_1, \ldots, a_m)$ the polynomial $(G(a))(z) = \sum_{k=0}^{m} a_k z^k$, $\forall z \in \mathbf{C}$. If $(G(b))(z) = \sum_{k=0}^{m} b_k z^k$ is the generating function of b , then

$$(G(a * b))(z) = (G(a))(z)\,(G(b))(z), \qquad \forall z \in \mathbf{C} \,. \tag{3}$$

The convolution product of the sequences a and b can be calculated using the formula (3).

Example. $(G(c))(z)=(G(a*b))(z)=(G(a))(z)\,(G(b))(z)=$
$=(1+2z-z^3+z^4)(1+3z-z^2-2z^3)=1+5z+5z^2-5z^3-6z^4+z^6-2z^7$
,
and is obtained the same convolution product c as above.

3. Computing complete convolution by Z-transform

It is called *Z-transform* of a finite sequence $a=(a_0,a_1,\dots,a_m)$ the rational function $(Z(a))(z)=\sum_{k=0}^{m}a_k z^{-k}$, $\forall z\in\mathbf{C}$. If $(Z(b))(z)=\sum_{k=0}^{m}b_k z^{-k}$ is Z-transform of b, then

$$(Z(a*b))(z)=(Z(a))(z)\,(Z(b))(z),\quad \forall z\in\mathbf{C}. \tag{4}$$

The convolution product of the sequences a and b can also be calculated using the formula (4).

Example. $(Z(c))(z)=(Z(a*b))(z)=(Z(a))(z)\,(Z(b))(z)=$
$=(1+2z^{-1}-z^{-3}+z^{-4})(1+3z^{-1}-z^{-2}-2z^{-3})=$
$=1+5z^{-1}+5z^{-2}-5z^{-3}-6z^{-4}+z^{-6}-2z^{-7}$,
and is obtained the same convolution product c as above.

4. Matrix calculation of linear discrete convolution in complete form

We associate with the sequence a the matrices $M(a)\in M_{m+n+1,m+n+1}$ and $C(a)\in M_{m+n+1,1}$, given by relations

$$M(a)=\begin{bmatrix} a_0 & & & & & & & \\ a_1 & a_0 & & & & & & \\ \vdots & a_1 & \ddots & & & & & \\ a_{m-1} & \vdots & \ddots & a_0 & & & & \\ a_m & a_{m-1} & \vdots & a_1 & a_0 & & & \\ & a_m & \ddots & \vdots & \ddots & \ddots & & \\ & & \ddots & a_{m-1} & a_{m-2} & \cdots & a_0 & \\ & & & a_m & a_{m-1} & \cdots & a_1 & a_0 \end{bmatrix},\ C(a)=\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The convolution product can then be calculated by relations

$$M(a*b)=M(a)M(b),\ C(a*b)=M(a)C(b). \tag{5}$$

Examples. We have

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & -1 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & -2 & -1 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & -1 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 2 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\ -5 & 5 & 5 & 1 & 0 & 0 & 0 & 0 \\ -6 & -5 & 5 & 5 & 1 & 0 & 0 & 0 \\ 4 & -6 & -5 & 5 & 5 & 1 & 0 & 0 \\ 1 & 4 & -6 & -5 & 5 & 5 & 1 & 0 \\ -2 & 1 & 4 & -6 & -5 & 5 & 5 & 1 \end{bmatrix}, \text{ or}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & -1 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & -2 & -1 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & -1 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 5 \\ -5 \\ -6 \\ 4 \\ 1 \\ -2 \end{bmatrix},$$

and is obtained the same convolution product $c = a * b$ as above.

5. MATLAB calculation of complete convolution

The convolution product of the sequences a and b can be computed in MATLAB by instruction conv(a,b), that calculate the product of the polynomials having terms of sequences a and b as coefficients, hence the product of the generating functions of these sequences.

Example. $a=[1\,2\,0\,-1 1]$, $b=[1\,3\,-1\,-2]$, $c=conv(a,b)$

II. Truncated convolution

6. Linear discrete convolution in truncated form

Very useful is also the truncated form of linear discrete convolution (see [7]), which is calculated with a smaller number of arithmetic operations. Truncated convolution is calculated by the same formula used above, but only for sequences that have the same length, the result also having the same length as the factors. More exactly, let $a=(a_0,a_1,\ldots,a_n)$ and $b=(b_0,b_1,\ldots,b_n)$ be two sequences of the same length $n+1$. Truncated convolution of a and b is the sequence $c=a*b=(c_0,c_1,\ldots,c_n)$ of the same length $n+1$, whose coefficients are given by the formula

$$c_k=\sum_{j=0}^{k} a_{k-j}\, b_j\,,\ k=0,1,\ldots,n\,. \tag{6}$$

This convolution is also commutative, associative, distributive with respect to the addition of the sequences and has the unit $\delta=(1,0,0,\ldots,0)$, sequence of length $n+1$.

Truncated convolution can be calculated by each method given above, but truncated, with fewer calculations. We present below how the truncated convolution can be computed for the sequences from the previous examples, considered now of the same length, by adding a zero to the right of the shortest of them, namely $\tilde{c}=a*\tilde{b}$ for $a=(1,2,0,-1,1)$ and $\tilde{b}=(1,3,-1,-2,0)$.

Example. We have the truncated algorithm

$$\begin{array}{rrrrr}
1 & 2 & 0 & -1 & 1 \\
1 & 3 & -1 & -2 & 0 \\
\hline
1 & 2 & 0 & -1 & 1 \\
 & 3 & 6 & 0 & -3 \\
 & & -1 & -2 & 0 \\
 & & & -2 & -4 \\
\hline
1 & 5 & 5 & -5 & -6
\end{array}$$

hence $\tilde{c}=a*\tilde{b}=(1,5,5,-5,-6)$.

7. Computing truncated convolution by generator function

Example. We consider only polynomials up to degree 4, so

$$\begin{aligned}&(G(\tilde{c}))(z)=\left(G\left(a*\tilde{b}\right)\right)(z)=(G(a))(z)\left(G\left(\tilde{b}\right)\right)(z)=\\&=\left(1+2z-z^3+z^4\right)\left(1+3z-z^2-2z^3\right)=1+5z+5z^2-5z^3-6z^4,\end{aligned}$$

and is obtained the same convolution product $\tilde{c}$ as above.

8. Computing truncated convolution by Z-transform

Example. We consider only Z-transform up to power -4, so

$$\begin{aligned}&(Z(\tilde{c}))(z)=\left(Z\left(a*\tilde{b}\right)\right)(z)=(Z(a))(z)\left(Z\left(\tilde{b}\right)\right)(z)=\\&=\left(1+2z^{-1}-z^{-3}+z^{-4}\right)\left(1+3z^{-1}-z^{-2}-2z^{-3}\right)=\\&=1+5z^{-1}+5z^{-2}-5z^{-3}-6z^{-4},\end{aligned}$$

and is obtained the same convolution product $\tilde{c}$ as above.

9. Matrix calculation of linear discrete convolution in truncated form

By matrix method, the truncated convolution $c=(c_1,\ldots,c_n)$ of the sequences $a=(a_0,a_1,\ldots,a_n)$ and $b=(b_0,b_1,...,b_n)$ results by one of formulas (5), in which matrices $M(a)\in M_{n+1,n+1}$ and $C(a)\in M_{n+1,1}$ are given by relations

$$M(a)=\begin{bmatrix}a_0 & 0 & \cdots & 0 & 0\\ a_1 & a_0 & \cdots & 0 & 0\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ a_{n-1} & a_{n-2} & \cdots & a_0 & 0\\ a_n & a_{n-1} & \cdots & a_1 & a_0\end{bmatrix},\ C(a)=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_{n-1}\\ a_n\end{bmatrix}. \tag{7}$$

Example. We have

$$M(a)M\left(\tilde{b}\right)=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 2 & 1 & 0 & 0 & 0\\ 0 & 2 & 1 & 0 & 0\\ -1 & 0 & 2 & 1 & 0\\ 1 & -1 & 0 & 2 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 3 & 1 & 0 & 0 & 0\\ -1 & 3 & 1 & 0 & 0\\ -2 & -1 & 3 & 1 & 0\\ 0 & -2 & -1 & 3 & 1\end{bmatrix}=$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 \\ 5 & 5 & 1 & 0 & 0 \\ -5 & 5 & 5 & 1 & 0 \\ -6 & -5 & 5 & 5 & 1 \end{bmatrix} = M\left(a * \tilde{b}\right) = M(\tilde{c}), \text{ or}$$

$$M(a)C\left(\tilde{b}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ -1 & 0 & 2 & 1 & 0 \\ 1 & -1 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ -1 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 5 \\ -5 \\ -6 \end{bmatrix} = C\left(a * \tilde{b}\right) = M(\tilde{c}),$$

and is obtained the same truncated convolution $\tilde{c} = a * \tilde{b}$ as above.

10. MATLAB calculation of truncated convolution

We can easily create in MATLAB a new instruction, named $c = tconv(a,b)$, which unlike $conv(a,b)$, to compute the truncated convolution $c = a * b$, with a,b,c of same length.

Example. $a = [1\,2\,0\,-1\,1]$, $\tilde{b} = [1\,3\,-1\,-2\,0]$, $\tilde{c} = tconv\left(a, \tilde{b}\right)$.

11. Linear discrete convolution of infinite (unilateral) sequences

For two infinite (unilateral) sequences $a = (a_k : k \in \mathbf{N})$ and $b = (b_k : k \in \mathbf{N})$, the linear convolution $c = a * b = (c_k : k \in \mathbf{N})$ is defined by formula (6) for every $k \in \mathbf{N}$. The methods for calculating the truncated convolution apply in this case, with remarks that the index k is now a natural number, the matrices are infinite, and the finite sums must to be substitute with series, in case the latter are converging.

If a, b and c are above sequences of numbers, then the product of series $\sum_{n=0}^{\infty} a_n$ and $\sum_{n=0}^{\infty} b_n$ is given by formula

$$\sum_{n=0}^{\infty} a_n \sum_{n=0}^{\infty} b_n = \sum_{n=0}^{\infty} c_n .$$

(8)

About the convergence of these series, we know two main results (see [9]) :

Cauchy theorem. *If series* $\sum_{n=0}^{\infty} a_n$ *and* $\sum_{n=0}^{\infty} b_n$ *are absolutely convergent, then series* $\sum_{n=0}^{\infty} c_n$ *is absolutely convergent.*

Mertens theorem. *If one of series* $\sum_{n=0}^{\infty} a_n$ *and* $\sum_{n=0}^{\infty} b_n$ *is convergent, and the other is absolutely convergent, then the series* $\sum_{n=0}^{\infty} c_n$ *is convergent.*

For power series, there is a similar result with (8), namely

$$\sum_{n=0}^{\infty} a_n x^n \quad \sum_{n=0}^{\infty} b_n x^n = \sum_{n=0}^{\infty} c_n x^n . \tag{9}$$

References

[1] M. Cîrnu, Solving difference and differential equations by discrete deconvolution, *UPB Scientific Bulletin*, A, **69** (2007) no. 1, 13-26.

[2] M. Cîrnu, Determination of particular solutions of nonhomogeneous linear differential equations by discrete deconvolution, *UPB Scientific Bulletin*, A, **69** (2007) No. 2, 3-16.

[3]. M. Cîrnu, Approximate calculus by deconvolution of the polynomial roots, *UPB Scientific Bulletin*, A **69** (2007) No. 4, 9-22.

[4] M. Cîrnu, “First order differential recurrence equations with discrete auto-convolution”, *International Journal of Mathematics and Computation*, **4** (2009) S09, 124-128.

[5]. M. I. Cîrnu, F. D. Frumosu, Initial value problems for nonlinear differential equations solved by differential transform method, I, Journal of Information Systems and Operations Management, Vol. 3, No. 1, July 2009, 102-107.

[6]. M. I. Cîrnu, F. D. Frumosu, Initial value problems for nonlinear differential equations solved by differential transform method, II, Journal of Information Systems and Operations Management, Vol. 3, No. 2, December 2009, 381-387.

[7] M. A. Cuenod, Introduction a l’analyse impulsionelle. Principe et application, *Dunod, Paris*, 1970.

[8] H. J. Nussbaumer, Fast Fourier transform and convolution algorithms, *Springer Verlag*, 1982.

[9] W. Rudin, Principles of mathematical analysis, Mac Graw Hill, 1976.

[10] H. S. Wilf, Generating functionology, *Academic Press*, 1990.