计算数学丛书

数论变换

蒋增荣

计算数学丛书

数 论 变 换

蒋 增 荣

上海科学技术出版社

内 容 提 要

七十年代以来出现的数论变换是一种以数论为基础的计算循环卷积的方法．它是在以正整数 M 为模的整数环（域）Z_M 上定义的线性正交变换，所用的计算方法是数论中的同余运算．它在 Z_M 上具有循环卷积特性，基本函数又是由整数的方幂构成．本书从数学的角度介绍数论变换的原理、性质、快速算法，Mersenne 数变换，Fermat 数变换，伪 Fermat 数变换，复数数论变换，二维数论变换和减少字长的方法，数论变换的应用等．

本书可供高等学校计算数学专业的学生及研究生参考，也可供高等学校数学系和计算机科学系的师生以及计算数学工作者参考．

计算数学丛书

数 论 变 换

蒋 增 荣

上海科学技术出版社出版

（上海瑞金二路 450 号）

新华书店上海发行所发行　上海市印刷三厂印刷

开本 787×1092 1/32　印张 6.25　字数 137,000

1980 年 8 月第 1 版　1980 年 8 月第 1 次印刷

印数 1—12,000

书号：13119·841　定价：(科四) 0.60 元

出 版 说 明

《计算数学丛书》是为了适应计算数学和计算机科学的发展，配合高等院校计算数学教学的需要而组织的一套参考读物。读者对象主要是高等院校数学系和计算机科学系的学生、研究生，亦可供高等院校数学系和计算机科学系的教师以及工矿企业、科研单位从事计算工作的技术人员参考。

本丛书向读者介绍近代计算方法的一些主要进展及其适用范围和实用效果。每种书集中介绍一个专题，针对本专题的近代发展作综合性的介绍，内容简明扼要，重点突出，有分析，有评价，力图使读者对该专题的动向和发展趋势得到一个完整的了解。

本丛书已拟定的选题计有：《线性代数与多项式的快速算法》、《数论变换》、《数值有理逼近》、《矩阵特征值问题》、《Sobolev 空间引论》、《计算组合数学》、《样条与逼近》、《有限条形法》、《广义逆矩阵及其计算方法》、《非线性方程迭代解法》、《奇异摄动》、《Walsh 函数及其应用》、《多项式最佳逼近》、《坏条件常微分方程数值解》、《最优控制问题的计算方法》、《误差分析》、《最小二乘问题的数值解法》、《快速傅里叶变换》、《板壳问题非协调方法》、《外推法》、《并行算法》、《Padé 逼近》、《Monte Carlo 方法》、《差分格式理论》、《高维偏微分方程数值解》、《初值问题差分方法》等二十余种，将于一九八〇年初起陆续出版。

前　言

近年来，数字式信号处理在无线电电子学领域如遥感、自动控制等方面，甚至在无线电电子学以外的地震、石油勘探、医学技术等部门都获得了广泛的应用．采用线性变换的方法是数学上分析线性非时变系统的一种手段．其中最常用的就是离散傅里叶变换 (Discrete Fourier Transform，简写为 DFT)．1965 年 Cooley 和 Tukey 提出了离散傅里叶变换的快速算法 (Fast Fourier Transform，简记为 FFT)，大大节省了计算工作量，从而缩短了处理时间．因此，七十年代初期，许多部门中都已使用了 FFT．

但是，在数字信号序列的长度 N 很大的情况下，利用 FFT，计算量仍然很大．用 FFT 做一个 N 点的复信号变换，大约需要 $N\log_2 N$ 次复数乘法及 $N\log_2 N$ 次复数加法．利用 FFT 的循环卷积特性计算 N 点的循环卷积，需要三次变换及 N 次复乘，同时还存在舍入误差，从而不能得到高精度的卷积．此外，由于 DFT 的基本函数是三角函数，所以必需预先贮存基本函数．

能不能在保持循环卷积特性的前提下，采用比三角函数更为简单的基本函数呢？正如本书 **2** 中所证明的，在复数域内，具有循环卷积特性的唯一变换是 DFT．因此，在复数域内，不存在既具有循环卷积特性、基本函数又比三角函数更简单的线性正交变换．但是，事物是发展的，最近提出的一种以数论为基础的计算循环卷积的方法，已引起了人们的重视，这

种方法就叫做数论变换.

数论变换是在以正整数 M 为模的整数环(域)Z_M 上定义的线性正交变换,所用的运算法则是数论中的同余运算,它在 Z_M 上既具有循环卷积特性,基本函数又是由整数的方幂构成. 特别是 Fermat 数变换(Fermat Number Transform,简记为 FNT),其基本函数由 2 的方幂构成,变换长度 $N=2^m$. 因此,在二进制计算机上,FNT 根本不用乘法,仅为移位操作,且具有 FFT 类型的快速算法. 又由于是模运算,所以不存在舍入误差,从而能得到高精度的卷积. 此外,也不需要基本函数的贮存. 但是,FNT 也有缺点. 主要是它没有物理意义,在信号处理中不能运用中间过程;其次是估计误差有困难;再次就是字长比较受限制,不够灵活,并且所需字长与变换点数之间存在着严格的关系. 尽管如此,数论变换在发展中将会不断地完善起来.

本书从数学的角度较系统地介绍了数论变换的原理,性质,快速算法,Mersenne 数变换,Fermat 数变换,伪 Fermat 数变换,复数数论变换,二维数论变换和减少字长的方法,最后介绍了它在其它方面的应用以及它所应用的代码. 其中的定理和公式,都严格地做了证明. 为便于不熟悉数论基本知识的读者阅读,在第三章还介绍了一点必要的数论基本知识.

本书在写作过程中,始终得到孙本旺教授、汪浩教授等许多同志的热心支持和帮助,特表示感谢.

由于作者水平所限,书中难免有错误,恳切希望同志们批评指正.

作 者 1979.3.

目　录

—1—

卷积与循环卷积

设两个长为 N 的序列 x_n 和 $h_n(n=0, 1, \cdots, N-1)$，其卷积是指*

$$y_n=\sum_{k=0}^{N-1} x_k h_{n-k}=\sum_{k=0}^{N-1} x_{n-k}h_k \quad (n=0, 1, \cdots, N-1). \quad (1)$$

其中假定 $x_n=h_n=0(n<0)$．这种卷积在用电子计算机进行信息处理时是经常用到的．(1)式的矩阵表示是

* 通常在数字滤波等应用中，将遇到两个长度不同的序列 $x_n(n=0, 1, \cdots, M_1-1)$，$h_n(n=0, 1, \cdots, M_2-1)$，其卷积

$$y_n=\sum_{k=0}^{M_1-1} x_k h_{n-k}=\sum_{k=0}^{M_2-1} x_{n-k}h_k$$

$$(n=0, 1, \cdots, M_1+M_2-2;\ x_n=h_n=0,\ n<0).$$

但这种卷积可以通过补零变成长度相同(长度均为 M_1+M_2-2)的卷积(1)，输出序列 y_n 的长度亦为 M_1+M_2-2．例如，$x_n(n=0, 1)$，$h_n(n=0, 1, 2, 3)$，其卷积 y_n 的长度应为 4，即

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}=\begin{bmatrix} h_0 & 0 \\ h_1 & h_0 \\ h_2 & h_1 \\ h_3 & h_2 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix}. \quad (a)$$

在所给序列 x_n 和 h_n 的后面补零而成为长度为 4 的序列 $(x_0, x_1, 0, 0)$，(h_0, h_1, h_2, h_3)，作它们的如下卷积

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}=\begin{bmatrix} h_0 & & & 0 \\ h_1 & h_0 & & \\ h_2 & h_1 & h_0 & \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ 0 \\ 0 \end{bmatrix}. \quad (b)$$

由计算知，(a)式和(b)式所得结果相同．

由于上述原因，只需研究如正文(1)式那样序列长度相同、卷积序列长度亦相同的卷积．

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_0 & & & 0 & \\ h_1 & h_0 & & & \\ h_2 & h_1 & h_0 & & \\ \vdots & \vdots & \vdots & & \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}. \quad (1')$$

通常用列矢量来表示序列 x_n 和 $y_n(n=0, 1, \cdots, N-1)$. 比如要求下列两序列的卷积:

$$(x) = \begin{bmatrix} 1 \\ 2 \\ 0 \\ -2 \\ 1 \end{bmatrix}, \qquad (h) = \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \\ 1 \end{bmatrix},$$

按照(1)式或(1')式, 我们有

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & & & 0 & \\ -2 & 1 & & & \\ 1 & -2 & 1 & & \\ 0 & 1 & -2 & 1 & \\ 1 & 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -3 \\ 0 \\ 6 \end{bmatrix}.$$

直接计算(1)式, 大约需要 N^2 次乘法和 N^2 次加法, 当 N 很大时, 其计算量是超量的, 实际上难以完成且很费时间. 因此, 寻求快速算法以节省时间就是一件有意义的工作.

通常, 通过循环卷积来计算(1). 所谓两个序列 $x_n(n=0, 1, \cdots, N-1)$ 和 $h_n(n=0, 1, \cdots, N-1)$ 的循环卷积是指:

$$y_n = \sum_{k=0}^{N-1} x_k h_{\langle n-k \rangle_N} = \sum_{k=0}^{N-1} x_{\langle n-k \rangle_N} h_k$$
$$(n=0, 1, \cdots, N-1). \quad (2)$$

即

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_0 & h_{N-1} & h_{N-2} & \cdots & h_1 \\ h_1 & h_0 & h_{N-1} & \cdots & h_2 \\ h_2 & h_1 & h_0 & \cdots & h_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}. \tag{2'}$$

(2)式中的符号 $\langle k\rangle_N$ 表示整数 k 模 N 的最小非负剩余，也就是整数 k 被正整数 N 除所余的非负整数，例如

$$\langle 7\rangle_4=3, \qquad \langle -7\rangle_4=1.$$

七

从卷积与循环卷积的定义(式(1)和式(2))可知，它们是不同的. 比如上例中两个序列的循环卷积就是：

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & -2 \\ -2 & 1 & 1 & 0 & 1 \\ 1 & -2 & 1 & 1 & 0 \\ 0 & 1 & -2 & 1 & 1 \\ 1 & 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -5 \\ 1 \\ 6 \end{bmatrix}.$$

下面的引理说明如何用循环卷积来计算卷积.

引理 1 两个长为 N 的序列 x_n 和 h_n，其卷积(1)可通过如下的两个长为 $2N$ 的序列 $\hat{x}_n$ ($n=0, 1, \cdots, 2N-1$) 和 $\hat{h}_n$ ($n=0, 1, \cdots, 2N-1$)的循环卷积来计算.

设

$$\hat{x}_n=\begin{cases} x_n, & n=0, 1, \cdots, N-1, \\ 0, & \text{其它}; \end{cases} \tag{3}$$

$$\hat{h}_n=\begin{cases} h_n, & n=0, 1, \cdots, N-1, \\ 0, & \text{其它}; \end{cases} \tag{4}$$

$\hat{x}_n$ 和 $\hat{h}_n$ 的循环卷积记为 $\hat{y}_n$，即

$$\hat{y}_n=\sum_{k=0}^{2N-1}\hat{x}_k\hat{h}_{\langle n-k\rangle_{2N}}\quad(n=0,\ 1,\ \cdots,\ 2N-1),$$

则

$$y_n=\hat{y}_n\quad(n=0,\ 1,\ \cdots,\ N-1).\tag{5}$$

证明 由 $\hat{x}_n$ 的定义(3)知,当 $n=0,\ 1,\ \cdots,\ N-1$ 时,有

$$\hat{y}_n=\sum_{k=0}^{2N-1}\hat{x}_k\hat{h}_{\langle n-k\rangle_{2N}}=\sum_{k=0}^{N-1}x_k\hat{h}_{\langle n-k\rangle_{2N}}.$$

当 $n,\ k=0,\ 1,\ \cdots,\ N-1$ 时,有

$$-(N-1)\leqslant n-k\leqslant N-1.$$

由定义(4)知,

$$\hat{h}_{\langle n-k\rangle_{2N}}=\begin{cases}h_{n-k}, & 0\leqslant n-k\leqslant N-1,\\ 0, & -(N-1)\leqslant n-k<0;\end{cases}$$

故

$$\hat{y}_n=\sum_{\substack{k=0\\0\leqslant n-k\leqslant N-1}}^{N-1}x_kh_{n-k}.$$

由假设,当 $n<0$ 时, $h_n=0$, 故

$$\hat{y}_n=\sum_{k=0}^{N-1}x_kh_{n-k}=y_n.\qquad\text{证毕.}$$

读者如果用矩阵形式写出序列 $\hat{x}_n$ 和 $\hat{h}_n$ 的循环卷积 $\hat{y}_n$ $(n=0,\ 1,\ \cdots,\ 2N-1)$, 则可明显的看出 $\hat{y}_n$ 的前面 N 个值恰是 x_n 和 h_n 的卷积 $y_n(n=0,\ 1,\ \cdots,\ N-1)$ 的值.

在实际应用中, 还可能遇到一种有别于 (1) 式的卷积和 (2) 式的循环卷积, 称为恒定对角卷积(Constant Diagonal Convolution):

$$y_n=\sum_{k=0}^{N-1}x_kh_{n-k}\quad(n=0,\ 1,\ \cdots,\ N-1).\tag{6}$$

式中下标出现负值时, 不再如(1)式那样有 $h_n=0$, 也不如(2)式那样是周期的($h_n=h_{N+n}$). (6)式的矩阵形式是

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} h_0 & h_{-1} & h_{-2} & \cdots & h_{-(N-1)} \\ h_1 & h_0 & h_{-1} & \cdots & h_{-(N-2)} \\ h_2 & h_1 & h_0 & \cdots & h_{-(N-3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}. \quad (7)$$

卷积(1)和循环卷积(2)都是这种卷积的特殊情况. 式(6)可看作两个序列 $(x_0, x_1, x_2, \cdots, x_{N-1})$ 和 $(h_{-(N-1)}, h_{-(N-2)}, h_{-(N-3)}, \cdots, h_0, \cdots, h_{N-1})$之间的一种卷积.

引理2 设两个序列 x_n $(n=0, 1, \cdots, N-1)$ 和 h_n $(n=-N+1, -N+2, \cdots, 0, \cdots, N-1)$,其恒定对角卷积(6)可通过如下的两个长为 $2N$ 的序列 $\hat{x}_n$ 和 $\hat{h}_n$ 的循环卷积来计算.

设

$$\hat{x}_n = \begin{cases} x_n, & n=0, 1, \cdots, N-1, \\ 0, & \text{其它}; \end{cases} \quad (8)$$

$$\hat{h}_n = \begin{cases} 0, & n=0, \\ h_{-N+n}, & n=1, 2, \cdots, 2N-1; \end{cases} \quad (9)$$

$\hat{x}_n$ 和 $\hat{h}_n$ 的循环卷积记为 $\hat{y}_n$, 即

$$\hat{y}_n = \sum_{k=0}^{2N-1} \hat{x}_k \hat{h}_{\langle n-k \rangle_{2N}} \quad (n=0, 1, \cdots, 2N-1),$$

则

$$\hat{y}_{n+N} = y_n \quad (n=0, 1, \cdots, N-1). \quad (10)$$

证明 当 $n=0, 1, \cdots, N-1$ 时,由(8)式知,

$$\hat{y}_{n+N} = \sum_{k=0}^{2N-1} \hat{x}_k \hat{h}_{\langle n+N-k \rangle_{2N}} = \sum_{k=0}^{N-1} x_k \hat{h}_{\langle n+N-k \rangle_{2N}}.$$

由于当 $n, k=0, 1, \cdots, N-1$ 时,有

$$1 \leqslant n+N-k \leqslant 2N-1,$$

故由(9)式知, $\hat{h}_{\langle n+N-k \rangle_{2N}} = \hat{h}_{n+N-k} = h_{n-k}$.

于是 $$\hat{y}_{n+N}=\sum_{k=0}^{N-1}x_k h_{n-k}=y_n.$$ 证毕.

这个引理中的 $\hat{x}_n$ 和 $\hat{h}_n(n=0,\ 1,\ \cdots,\ 2N-1)$ 是由 x_n 和 h_n 通过补零和适当移位形成的. 读者仍可用循环卷积的矩阵形式(式(2′)), 来表示 $\hat{x}_n$ 和 $\hat{y}_n$ 的循环卷积 $\hat{y}_n\ (n=0,\ 1,\ \cdots,\ 2N-1)$, 利用矩阵的分块相乘法, 不难看出 $\hat{y}_n$ 的后面 N 个值就是 x_n 和 h_n 的恒定对角卷积 $y_n(n=0,\ 1,\cdots,\ N-1)$的值.

引理 1 和引理 2 分别将(1)式和(6)式化作循环卷积. 循环卷积可用变换法计算. 一般常用的变换为离散傅里叶变换(DFT).

DFT 的定义如下: 设序列 $x_n(n=0,\ 1,\ \cdots,\ N-1)$, 变换

$$X_k=\sum_{n=0}^{N-1}x_n W_N^{nk}\quad(k=0,\ 1,\ \cdots,\ N-1)\tag{11}$$

称为 DFT, 其逆变换(IDFT)为

$$x_n=\frac{1}{N}\sum_{k=0}^{N-1}X_k W_N^{-nk}\quad(n=0,\ 1,\ \cdots,\ N-1),\tag{12}$$

其中 $W_N=e^{-j\frac{2\pi}{N}}$.

利用复数域上 N 阶单位根的性质

$$\frac{1}{N}\sum_{n=0}^{N-1}W_N^{pn}=\begin{cases}1, & p\equiv 0\pmod N,\\ 0, & p\not\equiv 0\pmod N,\end{cases}\tag{13}$$

不难证明(11)和(12)确是一对互逆变换. 事实上, 将(12)式代入(11)式, 得

$$\sum_{n=0}^{N-1}\left(\frac{1}{N}\sum_{m=0}^{N-1}X_m W_N^{-nm}\right)W_N^{nk}$$

$$=\frac{1}{N}\sum_{m=0}^{N-1}X_m\left(\sum_{n=0}^{N-1}W_N^{n(k-m)}\right)\quad(k=0,\ 1,\ \cdots,\ N-1),$$

利用(13)式, 可知上式右端为 X_k.

(11)式和(12)式可写作如下矩阵形式:

$$(X)=T_N(x) \tag{11'}$$

$$(x)=T_N^{-1}(X) \tag{12'}$$

其中，
$$(x)=\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}, \qquad (X)=\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix},$$

$$T_N=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix},$$

$$T_N^{-1}=\frac{1}{N}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{-(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}.$$

DFT 最重要的性质是循环卷积特性，即两个序列 x_n 和 h_n 的 DFT 的乘积等于其循环卷积 y_n 的 DFT：

$$Y_k=X_k\cdot H_k \quad (k=0,\ 1,\ \cdots,\ N-1),$$

或

$$\mathrm{DFT}\{y_n\}=\mathrm{DFT}\{x_n\}\cdot\mathrm{DFT}\{h_n\}. \tag{14}$$

这是由于

$$Y_k=\sum_{n=0}^{N-1}y_nW_N^{nk}=\sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1}x_mh_{\langle n-m\rangle_N}\right]W_N^{nk}$$

$$=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x_mh_{\langle n-m\rangle_N}W_N^{nk},$$

记 $n-m=l$，则

$$Y_k=\sum_{m=0}^{N-1}x_m\left[\sum_{l=-m}^{N-1-m}h_{\langle l\rangle_N}W_N^{k(l+m)}\right];$$

而
$$\sum_{l=-m}^{N-m-1} h_{\langle l\rangle_N} W_N^{k(l+m)} = \sum_{l=-m}^{-1} h_{\langle l\rangle_N} W_N^{k(l+m)} + \sum_{l=0}^{N-m-1} h_{\langle l\rangle_N} W_N^{k(l+m)},$$

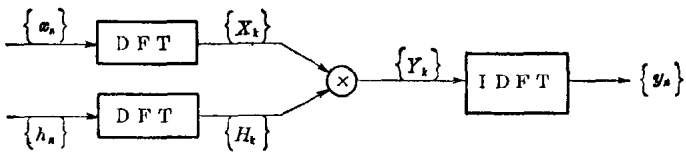
由于
$$h_{\langle l+N\rangle_N} = h_{\langle l\rangle_N}, \quad W_N^{k(l+m+N)} = W_N^{k(l+m)},$$

故
$$\sum_{l=-m}^{-1} h_{\langle l\rangle_N} W_N^{k(l+m)} = \sum_{l=N-m}^{N-1} h_l W_N^{k(l+m)};$$

所以
$$\sum_{l=-m}^{N-m-1} h_{\langle l\rangle_N} W_N^{k(l+m)} = \sum_{l=0}^{N-1} h_l W_N^{k(l+m)}.$$

于是
$$Y_k = \sum_{m=0}^{N-1} x_m \left[\sum_{l=0}^{N-1} h_l W_N^{k(l+m)} \right] = \sum_{m=0}^{N-1} x_m W_N^{mk} \cdot \sum_{l=0}^{N-1} h_l W_N^{lk}$$
$$= X_k \cdot H_k.$$

利用 DFT 的循环卷积特性可以计算两个序列 x_n 和 $h_n (n=0, 1, \cdots, N-1)$ 的循环卷积 y_n，这只要分别计算 x_n 和 h_n 的 DFT，即 X_k，H_k，将它们相乘就得到 y_n 的 DFT，即 $Y_k = X_k \cdot H_k$ $(k=0, 1, \cdots, N-1)$，最后将 Y_k 进行反变换 (IDFT)，就得到 y_n. 示意图如下：



由上可知，利用 DFT 的循环卷积特性计算长为 N 的序列的循环卷积，需要两次正变换，一次逆变换和 N 次乘法. 一次变换需要 N^2 次乘法，所以共需要 $3N^2+N$ 次乘法. 当 N 较大时，计算量很大，比不用变换法而直接计算循环卷积的计算量大得多. 但是如果 N 是高度复合数，特别当 $N=2^m$ (m 为自然数) 时，用快速傅里叶变换 (FFT) 进行计算，计算量大

为减少. 一个 N 点的变换用 FFT 计算约需 $N\log_2 N$ 次乘法，降低了两个数量级. 如果 $\{h_n\}$ 的变换预先计算好，那么用 FFT 实现 N 点的循环卷积只需 $2N\log_2 N+N$ 次乘法. 正是由于 FFT 的出现，DFT 才成为实用的方法.

以数论为基础的计算循环卷积的方法，在国内外已引起了重视，这种方法叫做数论变换(NTT). 特别引人注目的是 NTT 中有一种 Fermat 数变换(FNT)，这种变换只需加法(减法)及移位操作而不用乘法，从而提高了运算速度，这一点已为在通用计算机上的运算结果所证实. 对于实现长度不超过 256 的序列的循环卷积，FNT 比 FFT 缩短了时间达三至五倍. 下表列出了 R. C. Agarwal 与 C. S. Burrus 在 IBM 370/155 计算机上实现不同长度序列的循环卷积时，FFT 与 FNT 所需时间的比较:

表 1　实现长度为 N 的实序列的循环卷积计时

N	FFT(ms)	FNT 或 RT(ms)	N	FFT(ms)	FNT 或 RT(ms)
32	16	3.3	256	123	80.0(*)
64	31	7.4	512	245	166.0(*)
128	60	16.6	1024	530	340.0(*)
256	123	40.0	2048	1260	720.0(*)

其中，RT 为 FNT 的一种快速算法；(*)是用的二维 RT.

FNT 还消除了 FFT 带来的舍入误差，故能得到高精度的卷积，并且也不需要基函数的存贮，从而节省了存贮器. 但是，FNT 也有缺点，主要是它没有明显的物理意义；序列 $\{x_n\}$ 的变换 $\{X_k\}$ 不再是频谱，因此中间过程不能如 DFT 那样用来测速或测频，同时估计误差有困难；再就是字长很受限制，不够灵活. 但随着数论变换研究的深入及其算法的普及，数论变换将会不断地完善起来.