

共轭对称数据的 DFT 及其 FFT 算法¹

陈建平 曹清林* 沈世德*

(南通工学院信息工程系 江苏南通 226007)

*(南通工学院机械系 江苏南通 226007)

摘要 该文对共轭对称复数序列的离散傅里叶变换 (DFT) 及其快速傅里叶变换 (FFT) 算法进行了研究, 获得共轭对称序列的 DFT 具有虚部为零的性质, 并开发出适用于共轭对称数据的 FFT 算法. 该算法与传统 FFT 算法相比减少了一半的计算量和存储单元, 运算速度提高了一倍.

关键词 离散傅里叶变换, 快速傅里叶变换, 对称数据

中图分类号 TN911.72

1 引言

离散傅里叶变换 (DFT) 及其快速算法 (FFT) 通常是针对一般复数序列定义和推导的, 而在不少应用场合, DFT 的输入具有某些特殊性, 如为实数序列等等. 这些特殊类型序列的 DFT 及 FFT 常具有特定的性质和更有效的算法. 人们先后已对实数序列、实对称数据序列以及转移复共轭序列等特殊序列的 DFT 和 FFT 进行了探讨^[1-3]. 我们在运用 FFT 研究机构学中的对称连杆曲线时, 注意到通过适当安排, 可以使输入数据呈现出一种共轭对称性. 对这种序列的 DFT 和 FFT 进行研究, 可以得到 DFT 输出虚部为零的性质, 并可推导出运算量和存储量减半的 FFT 算法. 本文对此作以下介绍.

2 共轭对称序列 DFT 的性质

N 点共轭对称复数序列 $x(n)$ 定义为

$$x(N-n) = x^*(n), \quad n = 1, 2, \dots, N/2-1 \quad (1)$$

$$x(0) \text{ 和 } x(N/2) \text{ 为实数} \quad (2)$$

序列 $x(n)$ 的 DFT 为

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, \dots, N-1 \quad (3)$$

式中 $W_N = e^{-j2\pi/N}$. 为推导共轭对称序列 DFT 的性质, 将 (3) 式写成前后两部分之和

$$X(k) = \sum_{n=0}^{N/2} x(n)W_N^{kn} + \sum_{n=N/2+1}^{N-1} x(n)W_N^{kn} \quad (4)$$

对上式中第 2 项令 $n = N-r$, (4) 式成为

$$X(k) = \sum_{n=0}^{N/2} x(n)W_N^{kn} + \sum_{r=N/2-1}^1 x(N-r)W_N^{k(N-r)} \quad (5)$$

¹ 1998-12-14 收到, 2000-01-14 定稿

江苏省教委留学回国人员科研启动经费和省教委自然科学基金资助项目

将 r 换回为 n , 得到

$$X(k) = \sum_{n=0}^{N/2} x(n)W_N^{kn} + \sum_{n=1}^{N/2-1} x(N-n)W_N^{k(N-n)} \quad (6)$$

因为 $x(n)$ 和 W_N^{kn} 均满足共轭对称性, 即 $x(N-n) = x^*(n)$, $W_N^{k(N-n)} = (W_N^{kn})^*$, (6) 式可改写为

$$X(k) = \sum_{n=0}^{N/2} x(n)W_N^{kn} + \sum_{n=1}^{N/2-1} x^*(n)(W_N^{kn})^* \quad (7)$$

将 $x(n)$ 表示成实部和虚部的形式 $x(n) = x_r(n) + jx_i(n)$ 并将其与 $W_N^{kn} = \cos[(2\pi/N)kn]$

$-j \sin[(2\pi/N)kn]$ 一起代入 (7) 式, 经整理后得

$$X(k) = x_r(0) + x_r(N/2) \cos k\pi + 2 \sum_{n=1}^{N/2-1} \left[x_r(n) \cos\left(\frac{2\pi}{N}kn\right) + x_i(n) \sin\left(\frac{2\pi}{N}kn\right) \right], \quad k = 0, 1, \dots, N-1 \quad (8)$$

(8) 式表明, 对任何 k 值, $X(k)$ 恒为实数, $X(k)$ 的虚部即相位等于零。这样, 我们得到共轭对称序列的 DFT 输出虚部为零的性质。

同样, 对共轭反对称复数序列 $x(n)$, 即 $x(n)$ 满足

$$x(N-n) = -x^*(n), \quad n = 1, 2, \dots, N/2-1 \quad (9)$$

$$x(0) \text{ 和 } x(N/2) \text{ 为虚数} \quad (10)$$

可以推导出其 DFT 为

$$X(k) = j[x_i(0) + x_i(N/2) \cos k\pi] + 2j \sum_{n=1}^{N/2-1} \left[x_i(n) \cos\left(\frac{2\pi}{N}kn\right) - x_r(n) \sin\left(\frac{2\pi}{N}kn\right) \right], \quad k = 0, 1, \dots, N-1 \quad (11)$$

此时, $X(k)$ 恒为虚数, DFT 的实部为零。

3 共轭对称数据 FFT 算法

上节推导得出, 共轭对称序列 DFT 的输出有一半数据为零, 因而可以设想其 FFT 的运算量能减少一半。我们分别对时间抽取基 2 FFT、频率抽取基 2 FFT 以及 Bruun FFT^[4,5] 等算法进行了尝试。从频率抽取基 2 FFT 出发, 可以较方便地实现这一点。

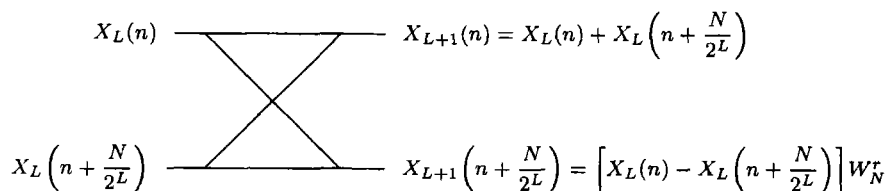


图 1 频率抽取基 2 FFT 的蝶形运算

频率抽取基 2 FFT 的蝶形运算如图 1 所示。图中, L 为级数 ($L = 1, 2, \dots, \log_2 N$)。第一级 $L = 1$, 输入 $X_1(n) = x(n)$, 输出为 $X_2(n)$ 。该级第 i 个蝶形和第 $(N/2 - i)$ 个蝶形 ($i = 1, 2, \dots, N/4 - 1$) 的输出分别为

$$X_2(i) = x(i) + x(i + N/2) \quad (12)$$

$$X_2(i + N/2) = [x(i) - x(i + N/2)]W_N^i \quad (13)$$

和

$$X_2(N/2 - i) = x(N/2 - i) + x(N/2 - i + N/2) = x(N/2 - i) + x(N - i) \quad (14)$$

$$X_2(N/2 - i + N/2) = X_2(N - i) = [x(N/2 - i) - x(N - i)]W_N^{N/2 - i} \quad (15)$$

对共轭对称输入序列 $x(n)$, (14) 式可改写成

$$\begin{aligned} X_2(N/2 - i) &= x(N - i) + x[N - (i + N/2)] \\ &= x^*(i) + x^*(i + N/2) = [x(i) + x(i + N/2)]^* \end{aligned} \quad (16)$$

对比 (12) 式, 上式表示

$$X_2(N/2 - i) = X_2^*(i) \quad (17)$$

同样, (15) 式可改写为

$$X_2(N - i) = \{x[N - (i + N/2)] - x(N - i)\}W_N^{N - i - N/2} \quad (18)$$

由于 $W_N^{-N/2} = -1$, 上式成为

$$\begin{aligned} X_2(N - i) &= \{x(N - i) - x[N - (i + N/2)]\}W_N^{N - i} \\ &= [x^*(i) - x^*(i + N/2)](W_N^i)^* = \{[x(i) - x(i + N/2)]W_N^i\}^* \end{aligned} \quad (19)$$

对比 (13) 式得到

$$X_2(N - i) = X_2^*(i + N/2) \quad (20)$$

(17) 式和 (20) 式表明, 第 i 个蝶形和第 $(N/2 - i)$ 个蝶形的上、下输出分别互为共轭对称值。其结果是第一级的输出为两个 $N/2$ 点的共轭对称序列。

同样, 在第二级的运算中, 上述两个 $N/2$ 点的共轭对称序列各自又被分别变换成两个 $N/4$ 点的共轭对称序列。这种规律一直保持下去, FFT 每一级的输出均由 2^L 个长度为 $N/2^L$ 的共轭对称子序列构成。图 2 画出了 16 点共轭对称数据 FFT 的信号流图。图中, 各级数据的第一列为实部, 第二列为虚部; 各组数据的对称性通过相同的字符标出。注意每个共轭对称子序列的第一个数据, 由于对应蝶形的输入为实数, 且相应的 $W_N^r = W_N^0 = 1$, 使得该数据的虚部为零。同样, 每个子序列对称中心点的数据由于相应蝶形的输入为共轭对称值, 且 $W_N^r = W_N^{N/4} = -j$, 其虚部亦为零。这些性质最终导致 FFT 倒数第二级和最后一级数据的虚部全为零。

根据以上的讨论, FFT 每一级的输出均由若干个对称数据组组成, 因此每一级的数据只需计算一半, 另一半可通过对称性获得。最后两级输出数据的虚部为零, 同样只需要计算实部这一半的数据。这样, 整个 FFT 的运算量减少一半。要去除一半的数据和蝶形的运算, 并不困难, 只需在 FFT 分组循环时, 计数前一半的组数。图 2 中将实际需要计算的蝶形用实线画出。

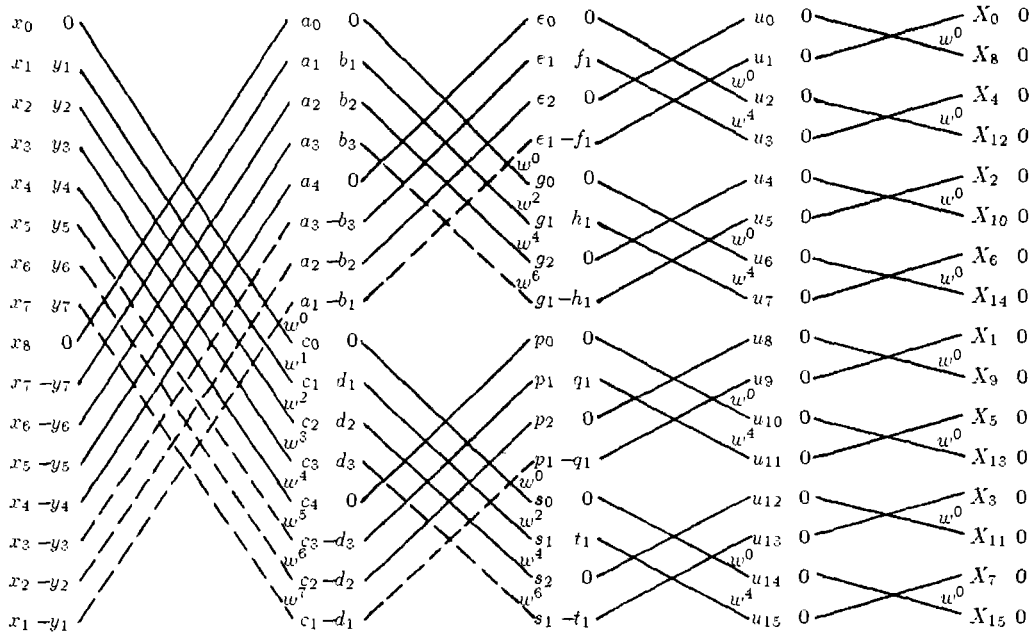


图 2 16 点共轭对称数据 FFT 信号流图

对共轭反对称输入序列可以得到类似的结论，即 FFT 各级的输出由 2^L 个长度为 $N/2^L$ 的共轭反对称子序列组成，最后两级输出的实部全为零，FFT 的运算量可减少一半。

4 原位运算的实现

在上节推导出运算量减半的 FFT 算法的基础上，通过对数据和蝶形进行适当的安排，可以进一步减少一半的存储单元。图 3 排出了图 2 所示 FFT 各级实际需要使用和计算的数据，图中将各级数据的实部和虚部并成一列。为实现原位运算，我们将蝶形进行组合。以第一级为例，第 0 个蝶形的输入为 x_0 和 x_8 ，输出为 a_0 和 c_0 。 a_0 对应于 x_0 的位置，而 c_0 对应于 y_4 的位置。 y_4 与 x_4 是第 4 个蝶形的输入，其输出为 a_4 和 c_4 。 a_4 对应于 x_4 的位置，而 c_4 恰好对应于 x_8 的位置。因此，只要将这两蝶形一起计算，就可以实现它们的原位运算。下一个蝶形即第 1 个蝶形的输入是 (x_1, y_1) 和 (x_7, y_7) ，输出为 (a_1, b_1) 和 (c_1, d_1) 。 (a_1, b_1) 对应 (x_1, y_1) 的位置，而 (c_1, d_1) 对应 (x_5, y_5) 的位置。 (x_5, y_5) 和 (x_3, y_3) 是第 3 个蝶形的输入，其输出是 (a_3, b_3) 和 (c_3, d_3) 。 (a_3, b_3) 对应于 (x_3, y_3) 的位置，而 (c_3, d_3) 恰好对应 (x_7, y_7) 的位置。同样，通过合并计算这两个蝶形，可以实现它们的原位运算。剩下的最后一个蝶形是第 2 个蝶形，其输入是 (x_2, y_2) 和 (x_6, y_6) ，输出是 (a_2, b_2) 和 (c_2, d_2) ，输出与输入的位置一一对应，直接可以原位计算。上述方法同样适用于其它各级蝶形的计算。其一般规律是每级第 0 组和第 $N/2^{L+1}$ 组中的蝶形逐一合并计算（如前所述， N 为 FFT 长度， L 为级数），并且这两个蝶形只需计算实部。然后依次将第 i 组和第 $N/2^{L+1} - i$ 组中的蝶形逐一合并计算 ($i = 1, 2, \dots, N/2^{L+2} - 1$)。最后第 $N/2^{L+2}$ 组中的蝶形每个独自计算。图 3 画出了上述三种蝶形合并或单独运算的蝶形图，它们分别如蝶形 a 、 b 、 c 所示（为清楚起见，未将它们画在同一级里）。最后两级蝶形的原位运算很容易实现，如图 3 中的蝶形 d 、 e 所示。这样，整个 FFT 的运算只需 N 个存储单元，比原来减少一半。

需要指出的是 FFT 原来每个蝶形的下输入数据为 $(x_n, -y_n)$ （见图 2）。在图 3 中，蝶形的下输入数据取的是其共轭对称值 (x_n, y_n) ， y_n 前面的负号被结合到蝶形运算的公式中去。

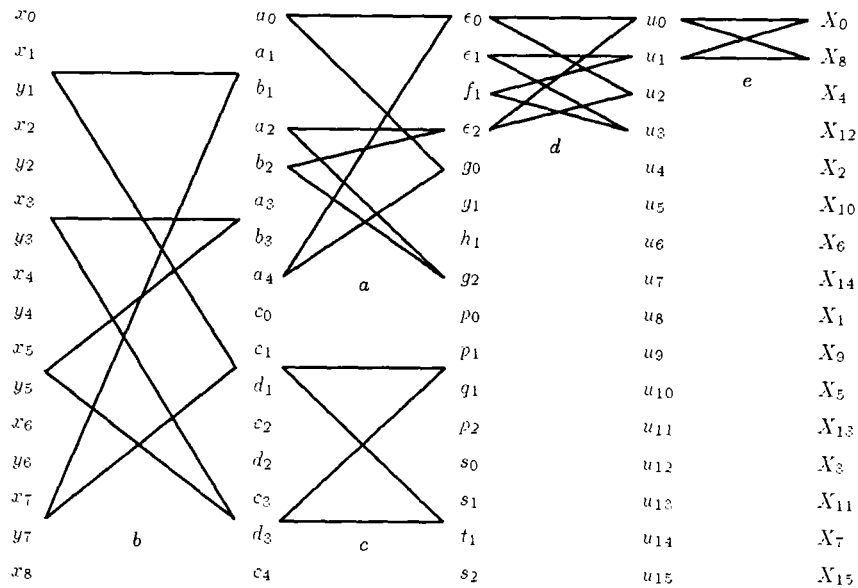


图 3 实现原位运算的蝶形图

5 实验结果

我们用 FORTRAN 语言实现了上述算法，其程序省略。最后两级的蝶形运算仅由简单的加法组成，编程时将这两级合并，以提高效率。FFT 输出数据的整序仍可采用常规的方法，并且只需要作实部的换位，更节省了操作。程序中数组 X 运算开始时包含 N 点共轭对称复数输入序列前一半的数据，各点数据的实部、虚部依次排列，如图 3 中所示。运算结束时 X 中存放的是 FFT 的输出数据，即输出实部的 N 个值，其排列顺序为自然顺序。

该程序在 PC 机上用 FORTRAN77 V5.10 编译环境调试通过。我们将其运算速度与传统的频率抽取基 2 FFT(DIFFFT) 进行了比较。表 1 给出了不同 FFT 长度下它们的运行时间。表中数据表明，本文提出的共轭对称数据 FFT(CSFFT) 比传统的 FFT 速度提高一倍，同时节约一半的存储单元。

表 1 CSFFT 与 DIFFFT 运算时间比较 (ms)

CSFFT 算法	长度	32	64	128	256	512	1024
	时间	0.11	0.27	0.61	1.27	2.75	5.83
DIFFFT 算法	长度	32	64	128	256	512	1024
	时间	0.24	0.60	1.26	2.58	5.49	11.69

6 结束语

作为 DFT 和 FFT 的一种特殊类型，本文提出和研究了共轭对称复数序列的 DFT 和 FFT。共轭对称 (或反对称) 复数序列的 DFT 具有虚部 (或实部) 为零的性质，其 FFT 内部呈现一定的对称规律。利用这些规律，可设法减少一半的运算量和存储单元，使 FFT 的速度提高一倍。我们将这种 DFT 和 FFT 应用于机构学中对称连杆曲线特征参数的提取，收到了很好的效果，得出了一套方便而有效的对称连杆曲线的综合方法 [6]。

参 考 文 献

- [1] H. V. Sorensen, *et al.*, Real-valued fast Fourier transform algorithms, IEEE Trans. on ASSP, 1987, ASSP-35(6), 849-863.
- [2] Chen Jianping, H. V. Sorensen, An efficient FFT algorithm for real-symmetric data, Proc. IEEE ICASSP, San Francisco, 1992, vol.5, 17-20.
- [3] Lu Chao, R. Tolimieri, New algorithm for the FFT computation of symmetric and translational complex conjugate sequences, Proc. IEEE ICASSP, San Francisco, 1992, Vol. 5, 13-16.
- [4] A. V. Oppenheim, R. W. Schaffer, Discrete-Time Signal Processing, Englewood Cliffs, New Jersey, Prentice-Hall, 1989, 581-609.
- [5] 陈建平, H. V. Sorensen, 复数、实数及实对称数据下 Bruun FFT 的实现, 苏州大学学报(自然科学), 1995, 11(4), 107-114.
- [6] 陈建平, 曹清林, 沈世德, 对称连杆曲线的 FFT 应用研究, 数据采集与处理, 1997, 12(3), 243-246.

DFT, FFT ALGORITHM FOR A COMPLEX
CONJUGATE-SYMMETRIC SEQUENCE

Chen Jianping Cao Qinglin Shen Shide

(Nantong Institute of Technology, Nantong 226007, China)

Abstract The discrete Fourier transform (DFT) and fast Fourier transform (FFT) for complex conjugate-symmetric input data are studied in this paper. The DFT of a complex conjugate-symmetric sequence has the nature that its imaginary part is zero. An efficient FFT algorithm is developed for such a sequence. It reduces the computation and storage requirements by half comparing to the traditional FFT algorithm.

Key words Discrete Fourier transform, Fast Fourier transform, Symmetric data

陈建平: 男, 1960年生, 副教授, 从事数字信号处理和快速算法等方面的研究.

曹清林: 男, 1962年生, 副教授, 从事机构学方面的研究.

沈世德: 男, 1944年生, 教授, 从事机构学方面的研究.