DFT 的新算法

黄 新 民

提 要 本文通过对离散富里叶变换（Discrete Fourier Transform，简记作DFT）矩阵的分解与FFT算法相结合，提出了一个计算DFT的新算法。由对矩阵的分解把求$N=2^t$点的DFT问题化为求16个$N/16$阶方阵与相应列向量相乘的问题（$N\geqslant16$）。从而减少了乘法运算次数，且还具有良好的并行运算性质。

I. 引 言

很久以来，富里叶变换一直是某些领域如线性系统、光学、概率论、量子物理、雷达、天线和信号分析等的一个基本分析工具。所以，随着近代计算机的发展，对DFT的计算也被引入到计算机上来完成。但对DFT来讲，却实在令人失望，即使是用速度惊人的高速计算机来计算DFT所花费的时间也是太多了。因此，无法在计算机上广泛地进行DFT的计算。

1965年库利——图基(Cooley-tocky)首先提出了对DFT的快速算法，减少了计算DFT的运算量，提高了运算速度。从此开拓了计算DFT的一个新天地。本文将给出一个算法，它将比FFT更为减少计算DFT的运算量，且具有良好的并行运算性质，所以，采用本算法很容易实现用多台低速计算机网络来计算DFT，从而使对DFT的计算速度成倍地提高。

II. $N=2^t$点DFT矩阵的分解($t\geqslant4$)

在进行深入的讨论之前，先给出长为N点的一維DFT的一般定义式。

定义 对于给定的自然数N，长为N的序列$\{x_n\}$，$(n=0,\cdots,N-1)$的DFT为：

$$X_k=\sum_{n=0}^{N-1}x_nW^{nk}\quad(k=0,\cdots,N-1),W=e^{j\frac{2\pi}{N}}\tag{1}$$

称(1)式为长为N的一維DFT，(1)式的矩阵表达式为：

本文1982年5月8日收到

$$\begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W^1 & W^2 & \cdots & W^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)^2} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{2}$$

记

$$W_N = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W^1 & W^2 & \cdots & W^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)^2} \end{pmatrix} \tag{3}$$

下面的讨论将仅对$N=2^t$点的一維 DFT 进行，为了后面讨论的需要，先引入几组记号。

$N1$：

$$A=\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad B=\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad C=\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$D=\begin{bmatrix} 1 & 1 \\ j & j \end{bmatrix} \quad E=\begin{bmatrix} 1 & 1 \\ -j & -j \end{bmatrix} \quad F=\begin{bmatrix} 1 & -1 \\ j & -j \end{bmatrix}$$

$$G=\begin{bmatrix} 1 & -1 \\ -j & j \end{bmatrix} \quad H=\begin{bmatrix} 1 & j \\ j & -1 \end{bmatrix} \quad P=\begin{bmatrix} 1 & j \\ -j & 1 \end{bmatrix}$$

$$Q=\begin{bmatrix} 1 & -j \\ -j & -1 \end{bmatrix}$$

$N2$：记号“$\circledast$”表示两个给定了相同个分块的矩阵的对应子块作直积。

例 设给定矩阵R,S的分块为：

$$R=\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \qquad S=\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

则

$$R\circledast S=\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}\circledast\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}=\begin{bmatrix} R_{11}*S_{11} & R_{12}*S_{12} \\ R_{21}*S_{21} & R_{22}*S_{22} \end{bmatrix}$$

其中“$*$”表示距阵的直积。

$N3$：记号W_N'表示对矩阵W_N的行列均作二进制逆序重排所得的矩阵[1]。

有了上面这些记号，我们就可以进行一般的讨论了。先看一个例子；

当$N=4$时，序列(x_0,x_1,x_2,x_3)的 DFT 矩阵表达式为：

$$\begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^9 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad (W^4=1)$$

则

$$W_4' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W^4 & W^2 & W^6 \\ 1 & W^2 & W^1 & W^3 \\ 1 & W^6 & W^3 & W^9 \end{pmatrix} \triangleq \begin{pmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} W^1 & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{pmatrix}$$

$$\triangleq\begin{bmatrix}1 & 1\\ 1 & W^1\end{bmatrix}_{(4\times4)_1}\circledast\begin{bmatrix}A & B\\ B^T & C\end{bmatrix}$$

称矩阵

$$\begin{bmatrix}1 & 1\\ 1 & W^1\end{bmatrix}_{(4\times4)_1}\triangleq\begin{bmatrix}A_1 & B_1\\ B_1^T & C_1\end{bmatrix}_{(4\times4)_1}\qquad(W^4=1)$$

为 4 点 DFT 矩阵的一次分块剩余矩阵。

定义1 对于 $N=2^t(t\geqslant2)$，将 $W_{N/2}$ 矩阵的元素由$\frac{N}{2}$阶的复单位根换成N阶的复单位根，再对其行、列分别作二进制逆序重排，所得的矩阵就称为 N点 DFT 矩阵的一次分块剩余矩阵，为方便起见，下面恆记为：

$$\begin{bmatrix}A_1 & B_1\\ B_1^T & C_1\end{bmatrix}_{(N\times N)_1}$$

其中 A_1，B_1，C_1 均为$\frac{N}{4}$阶的子方阵，B_1^T 为 B_1 的转置。

有了这个定义，我们有：

定理1 对 $N=2^t(t\geqslant2)$点 DFT 矩阵 W_N 作初等变换得 W_N'，则W_N' 恆可分解为：

$$\begin{bmatrix}A_1 & B_1\\ B_1^T & C_1\end{bmatrix}_{(N\times N)_1}\circledast\begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\qquad(4)$$

定理的结论是显然的，事实上，可以从二进制逆序重排的构造及数集$\{0,\cdots,N-1;\}$经二进制逆序重排后数的序关系和一次分块剩余矩阵的定义直接看出。

当$N=2^t$ $(t\geqslant4)$时，对一次分块剩余矩阵再作一次矩阵分解，有

定义2 对 $N=2^t$ $(t\geqslant4)$，将 $W_{\frac{N}{4}}$矩阵的元素由$\frac{N}{4}$阶的复单位根换成N阶复单位根，再对其行、列分别作二进制逆序重排，所得的矩阵就称为N点 DFT 矩阵的二次分块剩余矩阵。为方便起见，下面恆记为：

$$\begin{pmatrix}A_2 & B_2 & D_2 & E_2\\ B_2^T & C_2 & F_2 & G_2\\ D_2^T & F_2^T & H_2 & P_2\\ E_2^T & G_2^T & P_2^T & Q_2\end{pmatrix}_{(N\times N)_2}\qquad(W^N=1)\qquad(5)$$

其中 A_2、B_2、C_2、D_2、E_2、F_2、G_2、H_2、P_2、Q_2 均为$\frac{N}{16}$ 阶的子方阵，B_2^T、D_2^T、E_2^T、F_2^T，G_2^T、P_2^T 为对应方阵的转置。

与定理 1 类似，有下面的结论。

定理2 对$N=2^t(t\geqslant4)$点 DFF 矩阵的一次分块剩余矩阵恆可分解为：

$$\begin{pmatrix} A_2 & B_2 & D_2 & E_2 \\ B_2^T & C_2 & F_2 & G_2 \\ D_2^T & F_2^T & H_2 & P_2 \\ E_2^T & G_2^T & P_2^T & Q_2 \end{pmatrix}_{(N\times N)_2} \circledast \begin{pmatrix} A & B & D & E \\ B^T & C & F & G \\ D^T & F^T & H & P \\ E^T & G^T & P^T & Q \end{pmatrix} \tag{6}$$

定理2的证明与定理1类似。

一般地，依上述过程对DFT矩阵的分解，在本算法中只讨论到二次分解为止。

III. $N=2^t(t\geq4)$点的一维DFT的计算

下面的讨论仅对(2)式进行，当$N=2^t$时，对输入样本$\{x_n\}$和输出样本$\{X_k\}$，(n，$k=0$，…，$N-1$；)作二进制逆序重排，则(2)式化为：

$$(X_{i_0},\cdots,X_{i_{N-1}})^T=W'_N(x_{i_0},\ \cdots,\ x_{i_{N-1}})^T \tag{7}$$

由(4)式有：

$$\begin{pmatrix} X_{i_0} \\ X_{i_1} \\ \vdots \\ X_{i_{N-1}} \end{pmatrix} = \left(\begin{bmatrix} A_1 & B_1 \\ B_1^T & C_1 \end{bmatrix}_{(N\times N)_1} \circledast \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}\right) \begin{pmatrix} x_{i_0} \\ x_{i_1} \\ \vdots \\ x_{i_{N-1}} \end{pmatrix} \tag{8}$$

为了解(8)式，记

S 1：
$$y_0\begin{bmatrix} 1 \\ 1 \end{bmatrix}=A\begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix}\ \cdots,\quad y_{\frac{N}{4}-1}\begin{bmatrix} 1 \\ 1 \end{bmatrix}=\mathrm{A}\begin{bmatrix} x_{i_{\frac{N}{2}-2}} \\ x_{i_{\frac{N}{2}-1}} \end{bmatrix}$$

S 2：
$$y_{\frac{N}{4}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}=\mathrm{B}\begin{bmatrix} x_{i_{\frac{N}{2}}} \\ x_{i_{\frac{N}{2}+1}} \end{bmatrix}\cdots,\quad y_{\frac{N}{2}-1}\begin{bmatrix} 1 \\ -1 \end{bmatrix}=\mathrm{B}\begin{bmatrix} x_{i_{N-2}} \\ x_{i_{N-1}} \end{bmatrix}$$

S 3：
$$y_{\frac{N}{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}=\mathrm{B}^T\begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix}\ \cdots,\quad y_{\frac{3}{4}N-1}\begin{bmatrix} 1 \\ 1 \end{bmatrix}=\mathrm{B}^T\begin{bmatrix} x_{i_{\frac{N}{2}-2}} \\ x_{i_{\frac{N}{2}-1}} \end{bmatrix}$$

S 4：
$$y_{\frac{3}{4}N}\begin{bmatrix} 1 \\ -1 \end{bmatrix}=\mathrm{C}\begin{bmatrix} x_{i_{\frac{N}{2}}} \\ x_{i_{\frac{N}{2}+1}} \end{bmatrix}\cdots,\quad y_{N-1}\begin{bmatrix} 1 \\ -1 \end{bmatrix}=\mathrm{C}\begin{bmatrix} x_{i_{N-2}} \\ x_{i_{N-1}} \end{bmatrix}$$

由$N1$中矩阵A、B、C的元素间所存在的内在关系知，这里的记法是合理的。所以：

$$\begin{pmatrix} X_{i_0} \\ \vdots \\ X_{i_{N-1}} \end{pmatrix} = \begin{pmatrix} A_1 * A & B_1 * B \\ B_1^T * B^T & C_1 * C \end{pmatrix} \begin{pmatrix} x_{i_0} \\ \vdots \\ x_{i_{N-1}} \end{pmatrix}$$

$$=\begin{bmatrix} (A_1 * A)(x_{i_0},\ \cdots,\ x_{i_{\frac{N}{2}-1}})^T+(B_1 * B)(x_{i_{\frac{N}{2}}},\ \cdots,\ x_{i_{N-1}})^T \\ (B_1^T * B^T)(x_{i_0},\cdots,\ x_{i_{\frac{N}{2}-1}})^T+(C_1 * C)(x_{i_{\frac{N}{2}}},\ \cdots,\ x_{i_{N-1}})^T \end{bmatrix} \tag{9}$$

而
$$(A_1*A)\begin{pmatrix}x_{i_0}\\ \vdots\\ x_{i_{\frac{N}{2}-1}}\end{pmatrix}=\begin{pmatrix}a_{11}^{(1)}A & a_{12}^{(1)}A, & \cdots a_{1\frac{N}{4}}^{(1)}A\\ \cdots & \cdots & \cdots\ \cdots\\ a_{\frac{N}{4}1}^{(1)}A & a_{\frac{N}{4}2}^{(1)}A, & \cdots a_{\frac{N}{4}\frac{N}{4}}^{(1)}A\end{pmatrix}\begin{pmatrix}x_{i_0}\\ \vdots\\ x_{i_{\frac{N}{2}-1}}\end{pmatrix}$$

$$=\begin{pmatrix}a_{11}^{(1)}y_0\begin{bmatrix}1\\1\end{bmatrix}+a_{12}^{(1)}y_1\begin{bmatrix}1\\1\end{bmatrix}+\cdots+a_{\frac{N}{4}}^{(1)}\ y_{\frac{N}{4}-1}\begin{bmatrix}1\\1\end{bmatrix}\\ \cdots\qquad\cdots\qquad\cdots\qquad\cdots\\ a_{\frac{N}{4}1}^{(1)}y_0\begin{bmatrix}1\\1\end{bmatrix}+a_{\frac{N}{4}2}^{(2)}y_1\begin{bmatrix}1\\1\end{bmatrix}+\cdots+a_{\frac{N}{4}\frac{N}{4}}^{(1)}y_{\frac{N}{4}-1}\begin{bmatrix}1\\1\end{bmatrix}\end{pmatrix}$$

$$=(A_1(y_0,\cdots,y_{\frac{N}{4}-1})^T)*\begin{bmatrix}1\\1\end{bmatrix}\tag{10}$$

同理可以推出：

$$(B_1*B)(x_{i_{\frac{N}{2}}},\ \cdots,\ x_{i_{N-1}})^T=(B_1\ (y_{\frac{N}{4}},\ \cdots,\ y_{\frac{N}{2}-1})^T)*\begin{bmatrix}1\\-1\end{bmatrix}$$

$$(B_1^T*B^T)(x_{i_0},\ \cdots,x_{i_{\frac{N}{2}-1}})^T=(B_1^T\ (y_{\frac{N}{2}},\ \cdots,y_{\frac{3}{4}N-1})^T)*\begin{bmatrix}1\\1\end{bmatrix}$$

$$(C_1*C)(x_{i_{\frac{N}{2}}},\ \cdots,\ x_{i_{N-1}})^T=(C_1\ (y_{\frac{3}{4}N},\ \cdots,y_{N-1})^T)*\begin{bmatrix}1\\-1\end{bmatrix}$$

故(9)式化为：

$$\begin{pmatrix}X_{i_0}\\ \vdots\\ X_{i_{N-1}}\end{pmatrix}=\begin{pmatrix}(A_1(y_0,\cdots,\ y_{\frac{N}{4}-1})^T)*\begin{bmatrix}1\\1\end{bmatrix}+(B_1(y_{\frac{N}{4}},\ \cdots,\ y_{\frac{N}{2}-1})^T)*\begin{bmatrix}1\\-1\end{bmatrix}\\ (B_1^T(y_{\frac{N}{2}},\cdots,y_{\frac{3}{4}N-1})^T)*\begin{bmatrix}1\\1\end{bmatrix}+(C_1(y_{\frac{3}{4}N},\ \cdots,y_{N-1})^T)*\begin{bmatrix}1\\-1\end{bmatrix}\end{pmatrix}\tag{11}$$

又记：

$$\text{A1:}\quad\begin{cases}z_0\begin{bmatrix}1\\1\end{bmatrix}=\mathrm{A}\begin{bmatrix}y_0\\y_1\end{bmatrix},\ \cdots,\ z_{\frac{N}{16}-1}\begin{bmatrix}1\\1\end{bmatrix}=\mathrm{A}\begin{bmatrix}y_{\frac{N}{8}-2}\\y_{\frac{N}{8}-1}\end{bmatrix}\\ z_{\frac{N}{16}}\begin{bmatrix}1\\-1\end{bmatrix}=\mathrm{B}\begin{bmatrix}y_{\frac{N}{8}}\\y_{\frac{N}{8}+1}\end{bmatrix},\ \cdots,\ z_{\frac{N}{8}-1}\begin{bmatrix}1\\-1\end{bmatrix}=\mathrm{B}\begin{bmatrix}y_{\frac{N}{4}-2}\\y_{\frac{N}{4}-1}\end{bmatrix}\\ z_{\frac{N}{8}}\begin{bmatrix}1\\1\end{bmatrix}=\mathrm{B}^T\begin{bmatrix}y_0\\y_1\end{bmatrix},\ \cdots,\ z_{\frac{3}{16}N-1}\begin{bmatrix}1\\1\end{bmatrix}=\mathrm{B}^T\begin{bmatrix}y_{\frac{N}{8}-2}\\y_{\frac{N}{8}-1}\end{bmatrix}\\ z_{\frac{3}{16}N}\begin{bmatrix}1\\-1\end{bmatrix}=\mathrm{C}\begin{bmatrix}y_{\frac{N}{8}}\\y_{\frac{N}{8}+1}\end{bmatrix},\ \cdots,\ z_{\frac{N}{4}-1}\begin{bmatrix}1\\-1\end{bmatrix}=\mathrm{C}\begin{bmatrix}y_{\frac{N}{4}-2}\\y_{\frac{N}{4}-1}\end{bmatrix}\end{cases}$$

$$
\text{A2:}\quad\left\{\begin{array}{l}
z_{\frac{N}{4}}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{D}\begin{bmatrix}y_{\frac{N}{4}}\\ y_{\frac{N}{4}+1}\end{bmatrix},\ \cdots,\ z_{\frac{5}{16}N-1}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{D}\begin{bmatrix}y_{\frac{3}{8}N-2}\\ y_{\frac{3}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{5}{16}N}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{E}\begin{bmatrix}y_{\frac{3}{8}N}\\ y_{\frac{3}{8}N+1}\end{bmatrix},\ \cdots,\ z_{\frac{3}{8}N-1}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{E}\begin{bmatrix}y_{\frac{N}{2}-2}\\ y_{\frac{N}{2}-1}\end{bmatrix}\\[2ex]
z_{\frac{3}{8}N}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{F}\begin{bmatrix}y_{\frac{N}{4}}\\ y_{\frac{N}{4}+1}\end{bmatrix},\ \cdots,\ z_{\frac{7}{16}N-1}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{F}\begin{bmatrix}y_{\frac{3}{8}N-2}\\ y_{\frac{3}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{7}{16}N}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{G}\begin{bmatrix}y_{\frac{3}{8}N}\\ y_{\frac{3}{8}N+1}\end{bmatrix},\ \cdots,\ z_{\frac{N}{2}-1}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{G}\begin{bmatrix}y_{\frac{N}{2}-2}\\ y_{\frac{N}{2}-1}\end{bmatrix}
\end{array}\right.
$$

$$
\text{A3:}\quad\left\{\begin{array}{l}
z_{\frac{N}{2}}\begin{bmatrix}1\\ 1\end{bmatrix}=\mathrm{D}^{T}\begin{bmatrix}y_{\frac{N}{2}}\\ y_{\frac{N}{2}+1}\end{bmatrix},\ \cdots,\ z_{\frac{9}{16}N-1}\begin{bmatrix}1\\ 1\end{bmatrix}=\mathrm{D}^{T}\begin{bmatrix}y_{\frac{5}{8}N-2}\\ y_{\frac{5}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{9}{16}N}\begin{bmatrix}1\\ -1\end{bmatrix}=\mathrm{F}^{T}\begin{bmatrix}y_{\frac{5}{8}N}\\ y_{\frac{5}{8}N+1}\end{bmatrix},\cdots,\ z_{\frac{5}{8}N-1}\begin{bmatrix}1\\ -1\end{bmatrix}=\mathrm{F}^{T}\begin{bmatrix}y_{\frac{3}{4}N-2}\\ y_{\frac{3}{4}N-1}\end{bmatrix}\\[2ex]
z_{\frac{5}{8}N}\begin{bmatrix}1\\ 1\end{bmatrix}=\mathrm{E}^{T}\begin{bmatrix}y_{\frac{N}{2}}\\ y_{\frac{N}{2}+1}\end{bmatrix},\ \cdots,\ z_{\frac{11}{16}N-1}\begin{bmatrix}1\\ 1\end{bmatrix}=\mathrm{E}^{T}\begin{bmatrix}y_{\frac{5}{8}N-2}\\ y_{\frac{5}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{11}{16}N}\begin{bmatrix}1\\ -1\end{bmatrix}=\mathrm{G}^{T}\begin{bmatrix}y_{\frac{5}{8}N}\\ y_{\frac{5}{8}N+1}\end{bmatrix},\cdots,\ z_{\frac{3}{4}N-1}\begin{bmatrix}1\\ -1\end{bmatrix}=\mathrm{G}^{T}\begin{bmatrix}y_{\frac{3}{4}N-2}\\ y_{\frac{3}{4}N-1}\end{bmatrix}
\end{array}\right.
$$

$$
\text{A4:}\quad\left\{\begin{array}{l}
z_{\frac{3}{4}N}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{H}\begin{bmatrix}y_{\frac{3}{4}N}\\ y_{\frac{3}{4}N+1}\end{bmatrix},\ \cdots,\ z_{\frac{13}{16}N-1}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{H}\begin{bmatrix}y_{\frac{7}{8}N-2}\\ y_{\frac{7}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{13}{16}N}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{P}\begin{bmatrix}y_{\frac{7}{8}N}\\ y_{\frac{7}{8}N+1}\end{bmatrix},\ \cdots,\ z_{\frac{7}{8}N-1}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{P}\begin{bmatrix}y_{N-2}\\ y_{N-1}\end{bmatrix}\\[2ex]
z_{\frac{7}{8}N}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{P}^{T}\begin{bmatrix}y_{\frac{3}{4}N}\\ y_{\frac{3}{4}N+1}\end{bmatrix},\cdots,\ z_{\frac{15}{16}N-1}\begin{bmatrix}1\\ j\end{bmatrix}=\mathrm{P}^{T}\begin{bmatrix}y_{\frac{7}{8}N-2}\\ y_{\frac{7}{8}N-1}\end{bmatrix}\\[2ex]
z_{\frac{15}{16}N}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{Q}\begin{bmatrix}y_{\frac{7}{8}N}\\ y_{\frac{7}{N}N+-}\end{bmatrix},\ \cdots,\ z_{N-1}\begin{bmatrix}1\\ -j\end{bmatrix}=\mathrm{Q}\begin{bmatrix}y_{N-2}\\ y_{N-1}\end{bmatrix}
\end{array}\right.
$$

由 $N1$ 中所定义的矩阵元素间存在的内在关系知，上面的记法是合理的。又由(6)式及类似于(11)式的推导。得：

$$
\begin{aligned}
&A_1(y_0,\cdots,y_{\frac{N}{4}-1})^T\\
&=\begin{pmatrix}
(A_2(z_0,\cdots,z_{\frac{N}{16}-1})^T)*\begin{bmatrix}1\\ 1\end{bmatrix}+(B_2(z_{\frac{N}{16}},\cdots,z_{\frac{N}{8}-1})^T)*\begin{bmatrix}1\\ -1\end{bmatrix}\\[2ex]
(B_2^T(z_{\frac{N}{8}},\cdots,z_{\frac{3}{16}N-1})^T)*\begin{bmatrix}1\\ 1\end{bmatrix}+(C_2(z_{\frac{3}{16}N},\cdots,z_{\frac{N}{4}-1})^T)*\begin{bmatrix}1\\ -1\end{bmatrix}
\end{pmatrix}
\end{aligned}
\qquad (12)
$$

$B_1(y_{\frac{N}{4}},\cdots,y_{\frac{N}{2}-1})^T$

$$
=\begin{pmatrix} (D_2(z_{\frac{N}{4}},\ \cdots,\ z_{\frac{5}{16}N-1})^T)*\begin{bmatrix}1\\ j\end{bmatrix}+(E_2(z_{\frac{5}{16}N},\ \cdots,\ z_{\frac{3}{8}N-1})^T)*\begin{bmatrix}1\\ -j\end{bmatrix} \\ \\ (F_2(z_{\frac{3}{8}N},\cdots,\ z_{\frac{7}{16}N-1})^T)*\begin{bmatrix}1\\ j\end{bmatrix}+(G_2(z_{\frac{7}{16}N},\ \cdots,\ z_{\frac{N}{2}-1})^T)*\begin{bmatrix}1\\ -j\end{bmatrix} \end{pmatrix} \tag{13}
$$

$$B_1^T(y_{\frac{N}{2}},\ \cdots,\ y_{\frac{3}{4}N-1})^T$$

$$
=\begin{pmatrix} (D_2^T(z_{\frac{N}{2}},\ \cdots,\ z_{\frac{9}{16}N-1})^T)*\begin{bmatrix}1\\ 1\end{bmatrix}+(F_2^T(z_{\frac{9}{16}N},\ \cdots,\ z_{\frac{5}{8}N-1})^T)*\begin{bmatrix}1\\ -1\end{bmatrix} \\ \\ (E_2^T(z_{\frac{5}{8}N},\cdots,\ z_{\frac{11}{16}N-1})^T)*\begin{bmatrix}1\\ 1\end{bmatrix}+(G_2^T(z_{\frac{11}{16}N},\ \cdots,\ z_{\frac{3}{4}N-1})^T)*\begin{bmatrix}1\\ -1\end{bmatrix} \end{pmatrix} \tag{14}
$$

$$C_1(y_{\frac{3}{4}N},\ \cdots,\ y_{N-1})^T$$

$$
=\begin{pmatrix} (H_2(z_{\frac{3}{4}N},\cdots,\ z_{\frac{13}{16}N-1})^T)*\begin{bmatrix}1\\ j\end{bmatrix}+(P_2(z_{\frac{13}{16}N},\ \cdots,\ z_{\frac{7}{8}N-1})^T)*\begin{bmatrix}1\\ -j\end{bmatrix} \\ \\ (P_2^T(z_{\frac{7}{8}N},\cdots,\ z_{\frac{15}{16}N-1})^T)*\begin{bmatrix}1\\ j\end{bmatrix}+(Q_2(z_{\frac{15}{16}N},\ \cdots,\ z_{N-1})^T)*\begin{bmatrix}1\\ -j\end{bmatrix} \end{pmatrix} \tag{15}
$$

由(11)～(15)式知，计算(2)式所定义的一长为$N=2^t(t\geqslant 4)$点的一維DFT问题，变成为主要计算16个$\frac{N}{16}$阶的方阵与16个对应列向量的相乘问题。而这16个方阵是由对DFT矩阵作二次分解所生成的N点DFT矩阵的二次分块剩余矩阵的子阵，在(5)式中给出了它的一般定义式。这里的记号完全同(5)式中所定义的。这16个列向量可由$A1$～$A4$的定义式求出，由$S1$～$S4$、$A1$～$A4$知，求出$\{z_0,\cdots,z_{N-1}\}$；（即16个列向量）只要作$2N$个复数加法和$\frac{N}{2}$个乘以$j=\sqrt{-1}$的乘法，但由于计算机在处理乘以±1和$\pm j$这一类平凡乘法（Trivial Multiplication）时是非常简单的。故通常在进行乘法次数估计时是不计的，所以，求出$\{z_0,\ \cdots,\ z_{N-1}\}$，实际上只要作$2N$个复数加法。

现在来讨论(12)～(15)式中16个矩阵与对应列向量相乘的问题。为了避免那些雷同的推导，下面仅以H_2与相应列向量的乘法为例来讨论。为叙述方便，我们记与H_2相乘的列向量$(z_{\frac{3}{4}N},\ \cdots,\ z_{\frac{13}{16}N-1})^T$为$(z_0(0),\cdots,z_0(m-1))^T$$\left(其中\ m=\frac{N}{16}\right)$，并记其输出结果为$(Z_0,\ Z_1,\ \cdots,\ Z_{m-1})^T$。先看一个例子，当$N=64$时，有

$$
H_2^{(64)}=\begin{pmatrix} W^1 & W^9 & W^5 & W^{13} \\ W^9 & W^{81} & W^{45} & W^{117} \\ W^5 & W^{45} & W^{25} & W^{65} \\ W^{13} & W^{117} & W^{65} & W^{169} \end{pmatrix}\ (W^{64}=1) \tag{16}
$$

由(15)式知，对应列向量为$\left(\frac{N}{16}=4\right)$：

$$(z_{\frac{3}{4}N},\ \cdots,\ z_{\frac{13}{16}N-1})^T=(z_{48},\ z_{49},\ z_{50},\ z_{51})^T$$

对矩阵$H_2^{(64)}$的行进行二进制逆序排列得$H_2^{(64)\prime}$，

有
$$H_2^{(64)\prime}\begin{pmatrix} z_0^{(0)} \\ z_0^{(1)} \\ z_0^{(2)} \\ z_0^{(3)} \end{pmatrix}=\begin{pmatrix} W^1 & W^9 & W^5 & W^{13} \\ W^5 & W^{45} & W^{25} & W^{65} \\ W^9 & W^{81} & W^{45} & W^{117} \\ W^{13} & W^{117} & W^{65} & W^{169} \end{pmatrix}\begin{pmatrix} z_{48} \\ z_{49} \\ z_{50} \\ z_{51} \end{pmatrix} \tag{17}$$

我们用图1所示的步骤来解(17)式如下：

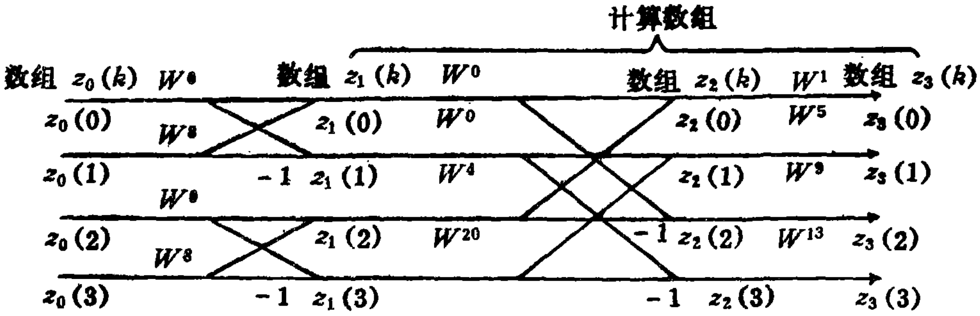

图 1　$N=64$，H_2与相应列向量相乘的运算流程图

从这个例子可以看出，这里的计算步骤完全与FFT类似，其唯一区别仅在于作乘法处理时，乘法因子W^p的指数p值与FFT中不一样。一般地，对$m=2^R$，我们说H_2与相应列向量的乘法也具有类似于FFT的计算步骤，而其作乘法处理时的乘法因子W^p的p值依下式确定。

设数组$z_l(k)(l=0,\cdots,R-1;\ k=0,\cdots,m-1;\ )$已求出，则对$z_l(k)$，（当

$$k=J\times 2^l+\alpha;\ \ \alpha=0,\cdots,2^l-1,\ J=0,\cdots,\frac{m}{2^l}-1;$$

时）作乘法处理，有

$$p=\begin{cases} 0 & 2\mid J \\ 2^{R+1-l}(1+2^2\alpha) & 2\nmid J \end{cases} \tag{18}$$

对$z_R(k)$，有$z_{R+1}(k)=W^{4k+1}z_R(k)$。

由$l=0,\cdots,R-1$；故需要对$z_0(k)$，$\cdots$，$z_{R-1}(k)$这k个数组分别进行一次乘法处理。每一数组进行一次乘法处理需$\frac{m}{2}$个复数乘法，再由$z_l(k)$经乘法处理后的数据去求$z_{l+1}(k)$要做m次复数加法，所以求出数组$z_R(k)$共要做$m\times R$次复数加法，$\frac{m}{2}\times R$次复数乘法。由$z_R(k)$求$z_{R+1}(k)$还要做m次复数乘法，综上，求出$z_{R+1}(k)$共要做$m\times R$次复数加法和$\frac{m}{2}\times R+m$次复数乘法。而$\{z_{R+1}(k)\}$即为$\{Z_K\}$依二进制逆序重排所得到的序

列。所以，计算 H_2 与向量$(z_{\frac{3}{4}N},\cdots,z_{\frac{13}{16}N-1})^T$的乘法，要做$\frac{m}{2}\times R+m$次复数乘法和$R\times m$次复数加法。

现在我们来证明上述矩阵乘法计算过程的合理性。

设矩阵 H_2 的阶数为 $m=2^R$，由 $m=\frac{N}{16}$，故 $N=2^{R+4}$，沿用前面规定的记号，有：

$$Z_k=\sum_{n=0}^{m-1}z_0(n)W^{L_2(K)L_2(n)} \qquad (p1)$$

其中 $L_2(x)$定义为：将整数 0，…，4m－1；依二进制逆序重排得

$$\Theta\triangleq\{i_0,\ i_1,\ \cdots,\ i_{4m-1}\}$$

取子集$\Phi_2=\{i_{2m},\ \cdots,\ i_{3m-1}\}\subset\Theta$，则定义集$\{0,\cdots,m-1\}$到 Φ_2 上的映射 L_2 为：

$$L_2:\qquad x\longrightarrow i_{2m+x}$$

容易证明 L_2 是一个双映射。

由于 $m=2^R$；$L_2(k),L_2(n)\in\Phi_2$，$(k,n\in\{0,\cdots,m-1\})$故 L_2 可用二进制表示为：

$$\begin{aligned}L_2(k)&=2^{R+1}k_{R+1}^{(2)}+2^Rk_R^{(2)}+\cdots+2^1k_1^{(2)}+2^0k_0^{(2)}\\ L_2(n)&=2^{R+1}n_{R+1}^{(2)}+2^Rn_R^{(2)}+\cdots+2^1n_1^{(2)}+2^0n_0^{(2)}\end{aligned} \qquad (p2)$$

注意到 L_2的像集为Φ_2，则对（$p2$)式，恆有：

$$k_0^{(2)}=1,\ n_0^{(2)}=1;\ k_1^{(2)}=0,\ n_1^{(2)}=0 \qquad (p3)$$

将$(p3)$代入$(p2)$式中得：

$$\begin{aligned}L_2(k)&=2^{R+1}k_{R+1}^{(2)}+\cdots+2^2k_2^{(2)}+1\\ L_2(n)&=2^{R+1}n_{R+1}^{(2)}+\cdots+2^2n_2^{(2)}+1\end{aligned} \qquad (p4)$$

又一个十进制数的二进制表示是唯一的，且互为可逆，所以 $L_2(x)$ $(x\in\{0,\cdots,m-1;\})$ 与一个分量为 0 式 1 的 R 維向量一一对应，而L_2 是一个双映射，故由复合函数的性质知：如

$$L_2(k)=2^{R+1}k_{R+1}^{(2)}+\cdots+2^2k_2^{(2)}+1$$

则
$$k\longleftrightarrow(k_{R+1}^{(2)},\ \cdots,\ k_2^{(2)}) \qquad (p5)$$

故
$$\begin{aligned}Z_k&=z_{(k_{R+1}^{(2)},\cdots,k_2^{(2)})}=\sum_{n=0}^{m-1}z_0(n)W^{L_2(k)L_2(n)}\\ &=\sum_{n_2=0}^{1}\cdots\sum_{n_R=0}^{1}\ \sum_{n_{R+1}=0}^{1}z_0(n_{R+1},\cdots,n_2)W^s\end{aligned} \qquad (p6)$$

其中 $S=(2^{R+1}k_{R+1}^{(2)}+\cdots+2^2k_2^{(2)}+1)(2^{R+1}n_{R+1}^{(2)}+\cdots+2^2n_2^{(2)}+1)$，而$W$为$N$阶的复单位根，则$(N=2^{R+4})$

$$\begin{aligned}W^s=&W^{L_2(k)}W^{(2^2k_2^{(2)}+1)2^{R+1}n_{R+1}^{(2)}}\cdots\cdots\\ &W^{(2^{R+1}k_{R+1}^{(2)}+\cdots+2^2k_2^{(2)}+1)\,2^2n_2^{(2)}}\end{aligned} \qquad (p7)$$

将$(p7)$代入$(p6)$得：

$$z_k=W^{L_2(k)}\sum_{n_2=0}^{1}\cdots\sum_{n_{R+1}=0}^{1}z_0(n_{R+1}^{(2)},\ \cdots,\ n_2^{(2)})\cdot$$

$$W^{(2^2 k_2^{(2)}+1)2^{R+1} n_{R+1}^{(2)}} \cdot \dots \cdot W^{(2^{R+1} k_{R+1}^{(2)}+\dots+2^2 k_2^{(2)}+1) n_2^{(2)} \cdot 2^2}$$

$$= W^{L_2(k)} \sum_{n_2=0}^{1} W^{(2^{R+1} k_{R+1}^{(2)}+\dots+2^2 k_2^{(2)}+1)2^2 n_2^{(2)}} \cdot \underbrace{\{\dots\{}_{R-2}$$

$$\cdot \left[\sum_{n_{R+1}=0}^{1} W^{(2^2 k_2^{(2)}+1)2^{R+1} n_{R+1}^{(2)}} z_0 (n_{R+1}^{(2)}, \cdots, n_2^{(2)}) \right] \} \cdots \} \qquad (p8)$$

由 $(p8)$可见，前述由数组 $z_l(k)$求数组 $z_{l+1}(k)$的计算过程对应着对 $n_{R+1-l}^{(2)}$ 作“Σ”和的过程。而数据乘以 $W^{2^{R+3} n_{R+1-l}^{(2)} k_{2+l}^{(2)}+p} = (-1)^{n_{R+1-l}^{(2)} k_{2+l}^{(2)}} W^p$的指数 p 与 (18) 式所规定的完全一样，且由乘法处理后的两个对偶结点[3]是作加法还是作减法依赖于$n_{R+1-l}^{(2)}$和$k_{2+l}^{(2)}$的值，当$n_{R+1-l}^{(2)}$与$k_{2+l}^{(2)}$的值均为 1 时，则做减法，否则做加法。显然，这个结论与图中的运算步骤是相吻合的，这样就证明了上述矩阵乘法计算过程的合理性。

综上，讨论了$N=2^t(t\geqslant 4)$阶的DFT矩阵的二次分块剩余矩阵的子阵H_2与其对应的列向量相乘的计算过程。对其他15个阵与其对应列向量相乘的问题，也完全可以类似地给出其计算过程来，事实上，每一个子阵与列向量相乘的计算过程是沒有什么差异的。而唯一的差异仅在于作乘法时，指数 p 的值互不相同（由此将造成平凡乘法次数的差异），在此不作详细的讨论。在附录中将给出不同子阵 p 值的表达式。

IV. 运算量的估计及综合评述

在Ⅲ中曾对 H_2 与相应列向量相乘的运算量进行了估计，其共需$\frac{m}{2}\times R+m$ 次复数乘法，$m\times R$ 次复数加法。而计算 16 个矩阵乘法的运算过程是完全相同的。故算出16个矩阵乘法（不考虑不同子阵乘法因子 W^p 的指数 p 不同所造成的影响）。其共需要$16\left(\frac{m}{2}\times R+m\right)$次复数乘法和$16(m\times R)$次复数加法。

又由附录知，在计算 A_2、B_2^T、D_2^T、E_2^T时，有 $z_R(k)=z_{R+1}(k)$，$(k=0,\cdots,m-1;)$，故不要作最后一次输出乘法处理，在计算 A_2、B_2、D_2、E_2时，由 $z_0(k)$求$z_1(k)$ 和由 $z_1(k)$求$z_2(k)$以及在计算 B_2^T、C_2、G_2、F_2时，由 $z_0(k)$求 $z_1(k)$所进行的乘法处理均为平凡乘法。在进行运算量估计时，这样$10m$ 个平凡乘法是可以考虑不计的。故求出16个矩阵乘法的运算量（不计平凡乘法）为：

$$M_1=16\left(\frac{m}{2}\times R+m\right)-10m=\frac{N}{2}\left(\log_2 N-\frac{13}{4}\right) \qquad (19)$$

$$A_1=16(m\times R)=N(\log_2 N-4)$$

而在求$(z_0,\cdots,z_{N-1})$要做 $2N$ 个复数加法，将 16 个矩阵乘法的结果代入 (12)～(15) 式计算要做N个复数加法。再将算得的结果代入(11)式计算要做N个复数加法，这样就算出了序列$\{x_n\}$的 DFT 的一个排列，其运算量（不计平凡乘法）为：

$$M=\frac{N}{2}\left(\log_2 N-\frac{13}{4}\right) \qquad (20)$$

$$A=N(\log_2 N) \tag{21}$$

众所周知，对于一个$N=2^t$点 DFT 依 FFT 算法来计算，运算量为：

$$M_F=\frac{N}{2}\log_2 N-\frac{3}{2}N+2 \tag{22}$$

$$A_F=N\log_2 N \tag{23}$$

与 FFT 比较，本算法在乘法次数上减少了$\frac{N}{8}$次，且速乘次数比 FFT 减少了两次，相应地减少了由舍入所引起的积累误差。又因本算法中绝大部分的运算量都集中在16个矩阵乘法上，所以很容易实现并行算法，从而可成倍地提高计算 DFT 的速度。

最后总结一下本算法的运算步骤：

1° 对输入样本依二进制逆序重排。

2° 依 $S1$，$S2$，$S3$，$S4$ 求序列y_0，…，y_{N-1}.

3° 依 $A1$，$A2$，$A3$，$A4$求序列 z_0，…，z_{N-1}. 并依次分该序列为 16 个长为$\frac{N}{16}$的子序列。

4° 对 DFT 矩阵的二次分块剩余矩阵依(5)式分块，求每一个子阵与相应列向量的乘法。

5° 对 16 个矩阵乘法的结果序列分别进行二进制逆序重排。

6° 将重排的结果代入(12)～(15)式中进行计算。

7° 把(12)～(15)式算得的结果代入(11)式计算，并将结果进行二进制逆序重排，输出结果。

附　录

在Ⅲ中我们指出了矩阵乘法的运算过程中作乘法处理时，W^p 的指数 p 值随不同的子阵而改变，下面给出不同子阵 p 值的表达式（仍沿用Ⅲ中规定的记号）：对$l=0,\cdots,R-1$；$k=0,\cdots,m-1$；记 $k=J\times 2^l+\alpha$.$\left(\alpha=0,\cdots,2^l-1;\ J=0,\cdots,\frac{m}{2^l}-1;\right)$：

对矩阵 A_2：$z_l(k)$有：

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2^2\alpha) & (2\nmid J)\end{cases}$$

而数组$z_R(k)$有 $z_R(k)=z_{R+1}(k)$.

对矩阵 B_2^T：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2+2^2\alpha) & (2\nmid J)\end{cases}$$

而数组 $z_R(k)$有 $z_R(k)=z_{R+1}(k)$.

对矩阵 D_2^T：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(1+2^2\alpha) & (2\nmid J)\end{cases}$$

而数组 $z_R(k)$ 有 $z_R(k)=z_{R+1}(k)$.

对矩阵E_2^T：$z_l(k)$ 有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(3+2^2\alpha) & (2\nmid J)\end{cases}$$

而数组 $z_R(k)$ 有 $z_R(k)=z_{R+1}(k)$.

对矩阵 B_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{2(4k)}z_R(k)$.

对矩阵 C_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{2(4k+2)}z_R(k)$.

对矩阵F_2^T：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(1+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有$z_{R+1}(k)=W^{2(4k+1)}z_R(k)$.

对矩阵 G_2^T：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(3+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{2(4k+3)}z_R(k)$.

对矩阵 D_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{4k}z_R(k)$.

对矩阵F_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{4k+2}z_R(k)$.

对矩阵H_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(1+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$，有 $z_{R+1}(k)=W^{4k+1}z_R(k)$.

对矩阵 P_2^T：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(3+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$，有 $z_{R+1}(k)=W^{4k+3}z_R(k)$.

对矩阵E_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{3(4k)}z_R(k)$.

对矩阵 G_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(2+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{3(4k+2)}z_R(k)$.

对矩阵 P_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(1+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{3(4k+1)}z_R(k)$.

对矩阵 Q_2：$z_l(k)$有

$$p=\begin{cases}0 & (2\mid J)\\ 2^{R+1-l}(3+2^2\alpha) & (2\nmid J)\end{cases}$$

对 $z_R(k)$有 $z_{R+1}(k)=W^{3(4k+3)}z_R(k)$.

参考文献

[1] 蒋增荣著，数论变换，上海科学技术出版社。

[2] 张远达，熊金淹编，线性代数，人民教育出版社。

[3] E·O·布赖姆著，快速富里叶变换，上海科学技术出版社。

The New Algorithm of DFT

Huang Xin-min

Abstract

By dissolving Discrete Fourier Transform matrix and combining with the FFT algorithm, this paper has put out a new method to compute DFT. Because of dissolving the matrix, one can change the problem of extracting DFT of $N=2^t$ points into a problem to multiply 16 square matrixes of $\frac{N}{16}$ order with the corresponding column vectors ($N\geqslant 16$). Thus, this method can decrease the calculating times and has a satisfied parallel algorethm features.