平行十二面体区域上的快速离散傅立叶变换及其并行实现*1)

姚继锋　孙家昶

(中科院软件所并行计算实验室　北京 100080)

HFFT ON PARALLEL DODECAHEDRON DOMAINS AND ITS PARALLEL IMPLEMENTATION

Yao Jifeng　Sun Jiachang

(*Parallel Computing Lab, Institute of Software, Academica Sinica, Beijing,* 100080)

Abstract

In this paper, we propose a fast algorithm for computing the Discrete Generalized Fourier Transforms on parallel dodecahedron domains with 3 dimensions and 4 directions. Our fast algorithm (HFFT) reduces the computation complexity of DGFT from $O(N^6)$ to $O(N^3 \log N)$. A parallel implementation is given and it has been run on a Linux Cluster up to 32 CPUs.

Key words: 3D FFT, Parallel Dodecahedron Domain, HFFT

§1. 引　言

快速傅立叶变换在信号处理、多媒体压缩、模式识别、计算化学等众多领域有着广泛的应用 [1], 它是公认的二十世纪最重要的十个算法之一 [2]. 2002 年高性能计算界影响最大的成果之一即是 Mitsuo Yokokawa 等在 Earth Simulator 上利用三维 FFT 成功的计算了网格尺寸为 $2048 \times 2048 \times 2048$ 的湍流问题 [3]. 但现有的快速傅立叶方法在实现高维傅立叶变换 (HFT) 时多是通过张量积方法将高维问题转化为低维问题来解决, 它所能处理的区域在二维和三维情形下分别为平行四边形和平行十二面体区域. 如何得到并计算非规则区域上的傅立叶变换, 一直是棘手的问题. 孙家昶等在最近的一系列工作中给出了二维三方向、三维四方向以及一般的 m 维 m+1 方向区域上广义傅立叶变换的定义及理论证明 [4–8]. 在此工作基础上, 孙家昶、姚继锋利用区域分解和多色排序的思想, 实现了平行六边形区域上的快速离散傅立叶变换 [9]. 对边长为 N 的平行六边形区域, 该算法将计算复杂度从直接计算的 $O(N^4)$ 降到了 $O(N^2 \log N)$. 本文将此算法推广到三维情形, 给出三维四方向平行十二面体区域上的快速广义离散傅立叶变换 (HFFT, High-dimension FFT).

* 2003 年 9 月 24 日收到.

1) 国家基金委项目 (60173021) 及中科院 "超级计算环境建设与应用" 项目 (INF105-SCE-02-05) 资助.

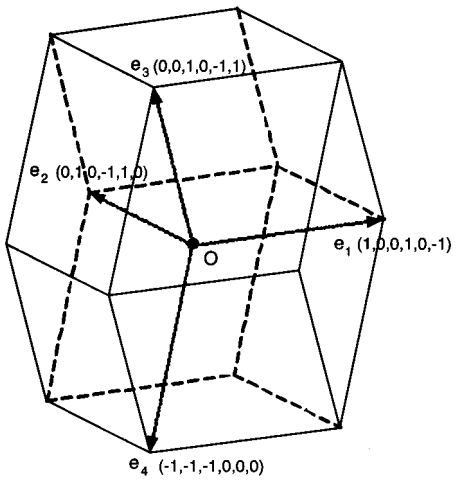

图 1 三维四方向平行十二面体

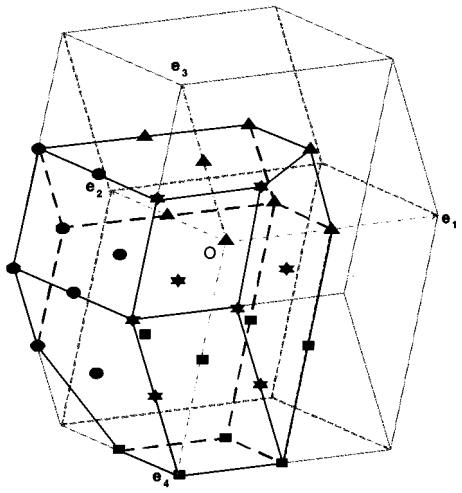

图 2 剔除周期边界的平行十二面体区域 Λ_N

孙家昶、施锡泉在研究高维空间上的 B 样条时发现平面六边形在空间的自然推广是三维四方向平行十二面体 [10]. 这种平行十二面体可用如下的方式给出:

给定空间三个线性无关向量 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, 记 Gram 矩阵为

$$G(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \mathbf{e}_1 \cdot \mathbf{e}_2 & \mathbf{e}_1 \cdot \mathbf{e}_3 \\ \mathbf{e}_2 \cdot \mathbf{e}_1 & \mathbf{e}_2 \cdot \mathbf{e}_2 & \mathbf{e}_2 \cdot \mathbf{e}_3 \\ \mathbf{e}_3 \cdot \mathbf{e}_1 & \mathbf{e}_3 \cdot \mathbf{e}_2 & \mathbf{e}_3 \cdot \mathbf{e}_3 \end{pmatrix}.$$

令 $\mathbf{e}_4 = -\mathbf{e}_1 - \mathbf{e}_2 - \mathbf{e}_3$, $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}[G(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)]^{-1}$, 则有

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{n}_1 & \mathbf{e}_1 \cdot \mathbf{n}_2 & \mathbf{e}_1 \cdot \mathbf{n}_3 \\ \mathbf{e}_2 \cdot \mathbf{n}_1 & \mathbf{e}_2 \cdot \mathbf{n}_2 & \mathbf{e}_2 \cdot \mathbf{n}_3 \\ \mathbf{e}_3 \cdot \mathbf{n}_1 & \mathbf{e}_3 \cdot \mathbf{n}_2 & \mathbf{e}_3 \cdot \mathbf{n}_3 \end{pmatrix} = G(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)[G(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)]^{-1} = I. \tag{1.1}$$

对空间任意一点 P, 定义六向坐标 $(t_1, t_2, t_3, t_4, t_5, t_6)$, 其中

$$t_i = P \cdot \mathbf{n}_i \quad (i = 1, 2, 3), \quad t_4 = t_1 - t_2, \quad t_5 = t_2 - t_3, \quad t_6 = t_3 - t_1,$$

则由 (1.1) 知 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ 的坐标分别为 $(1,0,0,1,0,-1)$, $(0,1,0,-1,1,0)$, $(0,0,1,0,-1,1)$, $(-1,-1,-1,0,0,0)$.

定义区域

$$\Omega_0 = \{P(t_1, t_2, t_3, t_4, t_5, t_6) | -1 \le t_i \le 1, i = 1, 2, \ldots, 6\}.$$

如图 1 所示, 它是空间的平行十二面体, 有 12 个平行四边形面, 24 条棱和 14 个顶点.

十二面体 Ω_0 有着许多有趣的几何性质，其中最重要的一条是如同任意一个平行六边形可以通过三个方向的平移铺满整个平面一样，它可以只通过六个方向的平移，无缝、无重叠的填满整个空间. 这是定义此区域上的周期函数的一个必要条件.

基于周期性的考虑，和一维的 $[-1,1)$ 区间类似，定义研究的平行十二面体半开区域为

$$\Omega = \{(t_1,t_2,t_3,t_4,t_5,t_6)| -1 \le t_1,t_2,\dots,t_5 < 1, -1 < t_6 \le 1,\\ t_4 = t_1 - t_2, t_5 = t_2 - t_3, t_6 = t_3 - t_1\}.$$

将区域 Ω 离散化，单位边长 N 等分，则其中的整数点集为

$$\Lambda_N = \{P|P = (k_1,k_2,k_3,k_4,k_5,k_6),\ -N \le k_1,k_2,k_3,k_4,k_5 < N, -N < k_6 \le N,\\ k_4 = k_1 - k_2, k_5 = k_2 - k_3, k_6 = k_3 - k_1\}.$$

Λ_N 即是我们所要研究的快速傅立叶变换的作用区域，其形状仍为一个平行十二面体，但此时十二个面中只有六个仍为平行四边形，相对应的另六个面为五边形 (如图 2 所示)，且 $dim(\Lambda_N) = 4N^3$. $N = 2$ 时，$dim(\Lambda_2) = 32$, 即是图 2 中的 32 个点，其中外层没有标记的点是周期意义下重复的点. 注意到 Λ_N 可以分割为四个平行六面体，它们分别由 e_1,e_2,e_3,e_4 中的三个向量张成：

$$\Lambda_N = \{P|P \in \Lambda_N, k_1,k_2,k_3 \ge 0\} \bigcup \{P|P \in \Lambda_N, k_3 < 0, k_5, -k_6 \ge 0\}\\ \bigcup \{P|P \in \Lambda_N, k_2,k_5 < 0, k_4 \ge 0\} \bigcup \{P|P \in \Lambda_N, k_1,k_4 < 0, k_6 > 0\}. \tag{1.2}$$

图 2 中用四种不同形状标示出 $N = 2$ 时，Λ_2 分割成为四个边长为 2 的平行六面体.

在本文中，如果不作特殊声明，黑体字符 $\mathbf{C},\mathbf{P},\mathbf{j},\mathbf{k}$ 总是表示相应的六向坐标，如 $\mathbf{C} = (C_1,C_2,C_3,C_4,C_5,C_6)$, 其中 $C_4 = C_1 - C_2, C_5 = C_2 - C_3, C_6 = C_3 - C_1$, $\mathbf{O}$ 代表原点 $(0,0,0,0,0,0)$.

为便于区域分解及多色排序，当每个方向细网格节点个数为 N, 粗网格节点个数为 $n = \dfrac{N}{m}$ 时，我们用如下三元组定义更一般的空间平行十二面体

$$\Lambda(\mathbf{C},n,m) = \{m\mathbf{k} + \mathbf{C}|\mathbf{k} \in \Lambda_n\}.$$

它表示中心为 $\mathbf{C}$, 每方向有 n 个点，相邻两点间隔为 m 的粗网格平行十二面体. 取 $\mathbf{C} = \mathbf{O}, m = 1$, 则有 $\Lambda(\mathbf{O},N,1) = \Lambda_N$, 它即是整体的细网格十二面体.

由孙家昶工作 [7] 知，平行十二面体区域 Λ_N 上的广义离散傅立叶变换可定义为

$$F_{\mathbf{k}} = \sum_{\mathbf{j} \in \Lambda_N} g_{\mathbf{jk}}^N f_{\mathbf{j}}, \quad \mathbf{k} \in \Lambda_N, \tag{1.3}$$

其中

$$g_{\mathbf{jk}}^N = e^{i\frac{\pi}{2}\frac{\mathbf{j}\cdot\mathbf{k}}{N}} = e^{i\frac{\pi}{2N}(j_1k_1+j_2k_2+j_3k_3+j_4k_4+j_5k_5+j_6k_6)}. \tag{1.4}$$

令 $G_N=(g_{\mathbf{jk}}^N)$, 它是一个 $4N^3$ 阶的方阵, 记区域 Λ_N 上 $F_{\mathbf{k}}$ 和 $f_{\mathbf{j}}$ 依次排列分别对应 $4N^3$ 阶列向量 $\mathbf{F}$ 和 $\mathbf{f}$, 则 (1.3) 的矩阵形式是

$$\mathbf{F}=G_N\mathbf{f}. \tag{1.5}$$

和二维平行六边形区域相比, 三维平行十二面体区域无论是公式推导还是算法实现, 问题的复杂度都大大增加. 利用计算区域的几何性质, 借鉴多色排序和区域分解的思想, 我们同样给出快速算法. 将 (1.5) 的浮点运算次数由 $O(N^6)$ 降为 $O(N^3\log N)$ 量级.

本文第 2 节给出了快速算法的公式推导, 从 $N=mn$ 的特殊情形开始, 最后推广到一般的 $N=N_0m^p$ 情形. 第 3 节给出串行的算法实现, 着重阐述了数据结构和数据排序. 第 4 节给出了并行算法. 第 5 节给出了数值实验的结果, 并作了简单的讨论.

§2. 算法推导

首先考察 $N=mn$ 时的情形.

定义 $\mathbb{Z}^3$ 中点集 $\sigma=\{\mathbf{Q}|\mathbf{Q}=(k_\mu),\mu=1,\cdots,6,k_4=k_1-k_2,k_5=k_2-k_3,k_6=k_3-k_1,\sum_{\nu=1}^{3}k_\nu=0(mod4)\}$, [8] 中指出

$$\mathbb{Z}^3=\bigcup_{\mathbf{Q}\in\sigma}\Lambda(N\mathbf{Q},N,1),\quad \Lambda(N\mathbf{Q}_1,N,1)\bigcap_{\mathbf{Q}_1,\mathbf{Q}_2\in\sigma,\mathbf{Q}_1\neq\mathbf{Q}_2}\Lambda(N\mathbf{Q}_2,N,1)=\emptyset. \tag{2.1}$$

上式表明通过平移, 十二面体网格 Λ_N 可以无缝无重叠的铺满整个三维整数点空间. 据此, 给出如下的周期性引理.

引理 2.1. 任给定 $\mathbb{Z}^3$ 中一点 $\mathbf{k}$, 存在点 $\widehat{\mathbf{k}}\in\Lambda_N$, 使得对任意 $\mathbf{j}\in\Lambda_N$, 均有 $g_{\mathbf{jk}}^N=g_{\mathbf{j}\widehat{\mathbf{k}}}^N$.

证明. 由 (2.1), $\exists\mathbf{Q}\in\sigma$ 使得 $\mathbf{k}\in\Lambda(N\mathbf{Q},N,1)$, 令 $\widehat{\mathbf{k}}=\mathbf{k}-N\mathbf{Q}$, 则 $\widehat{\mathbf{k}}\in\Lambda_N$. 由集合 σ 定义, 不妨设 $Q_1+Q_2+Q_3=4\mu$. 则 $\forall\mathbf{j}\in\Lambda_N$ 有

$$g_{\mathbf{jk}}^N=e^{i\frac{\pi}{2N}\mathbf{j}\cdot(\widehat{\mathbf{k}}-N\mathbf{Q})}=e^{i\frac{\pi}{2N}\mathbf{j}\cdot\widehat{\mathbf{k}}}e^{i\frac{\pi}{2}((4Q_1-4\mu)j_1+(4Q_2-4\mu)j_2+(4Q_3-4\mu)j_3)}=g_{\mathbf{j}\widehat{\mathbf{k}}}^N.$$

引理 2.1 表明, 利用周期性, Λ_N 外任意一点总能在 Λ_N 内找到其对应点, 且 $\mathbb{Z}^3$ 中所有点只有 $dim(\Lambda_N)=4N^3$ 种. 我们可以在周期意义下依照多色排序以及区域分解分别将 Λ_N 剖分为 m_3 个边长为 n 的小十二面体.

令 $\mathbf{P}_1=(1,0,0,1,0,-1),\mathbf{P}_2=(0,1,0,-1,1,0),\mathbf{P}_3=(0,0,1,0,-1,1)$, 定义如下的 m^3 个点:

$$\mathbf{P}_{i_1,i_2,i_3}=i_1\mathbf{P}_1+i_2\mathbf{P}_2+i_3\mathbf{P}_3,\quad 0\le i_1,i_2,i_3\le m-1 \tag{2.2}$$

和 m^3 个每方向长 n, 相邻两点间隔为 m 的粗网格平行十二面体 $\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)$.

另一方面, 每个平行十二面体周围有 12 个相邻的平行十二面体, 各个方向取单位长度时, 这 12 个平行十二面体的中心分别为 $\pm(1,1,2,0,-1,1),\pm(1,2,1,-1,1,0),\pm(2,1,1,1,$

$0,-1),\pm(1,0,-1,1,1,-2),\pm(-1,1,0,-2,1,1),\pm(0,-1,1,1,-2,1)$[8], 取其中线性无关的三个点, 不妨令 $\mathbf{C}_1=(2,1,1,1,0,-1)$, $\mathbf{C}_2=(1,2,1,-1,1,0)$, $\mathbf{C}_3=(1,1,2,0,-1,1)$. 定义如下的 m^3 个点

$$\mathbf{C}_{i_1,i_2,i_3}=ni_1\mathbf{C}_1+ni_2\mathbf{C}_2+ni_3\mathbf{C}_3,\quad 0\le i_1,i_2,i_3\le m-1 \tag{2.3}$$

和 m^3 个每方向点有 n 个点, 相邻两点间隔为 1 的平行十二面体 $\Lambda(\mathbf{C}_{i_1,i_2,i_3},n,1)$.

引理 2.2. 依据多色排序及区域分解, 可对 Λ_N 作如下的两种划分:

$$1)\ \Lambda_N=\bigcup_{i_1,i_2,i_3=0}^{m}\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m),\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)\bigcap_{(i_1,i_2,i_3)\neq(\widehat{i}_1,\widehat{i}_2,\widehat{i}_3)}\Lambda(\mathbf{P}_{\widehat{i}_1,\widehat{i}_2,\widehat{i}_3},n,m)=\emptyset$$

$$2)\ \Lambda_N=\bigcup_{j_1,j_2,j_3=0}^{m}\Lambda(\mathbf{C}_{j_1,j_2,j_3},n,1),\Lambda(\mathbf{C}_{j_1,j_2,j_3},n,1)\bigcap_{(j_1,j_2,j_3)\neq(\widehat{j}_1,\widehat{j}_2,\widehat{j}_3)}\Lambda(\mathbf{C}_{\widehat{j}_1,\widehat{j}_2,\widehat{j}_3},n,1)=\emptyset$$

证明. 由 (2.2) 知, $\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)$ 和 $\Lambda(\mathbf{P}_{\widehat{i}_1,\widehat{i}_2,\widehat{i}_3},n,m)$ 中的点前三个坐标分量对 m 取模分别为 (j_1,j_2,j_3) 和 $(\widehat{j}_1,\widehat{j}_2,\widehat{j}_3)$, 它们不相等. 注意到 $N=mn$, 同样依据取模不同, 有

$$\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)\bigcap\Lambda(\mathbf{P}_{\widehat{i}_1,\widehat{i}_2,\widehat{i}_3},n,m)=\emptyset.$$

又

$$dim(\Lambda_N)=4N^3=m^3 4n^3=\sum dim(\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)),$$

故有

$$\Lambda_N=\bigcup_{i_1,i_2,i_3=0}^{m}\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m).$$

1) 得证.

容易验证 $C_{j_1,j_2,j_3}\in\sigma$, 则由 (2.1) 知

$$\Lambda(\mathbf{C}_{j_1,j_2,j_3},n,1)\bigcap\Lambda(\mathbf{C}_{\widehat{j}_1,\widehat{j}_2,\widehat{j}_3},n,1)=\emptyset.$$

同样再由 $dim(\Lambda_N)=\sum\dim(\Lambda(\mathbf{C}_{j_1,j_2,j_3},n,1))$, 2) 得证.

依据引理 2.2, 下面通过对 $\mathbf{F}$ 和 $\mathbf{f}$ 分别进行区域分解和多色排序, 来计算 (1.5).

首先是对 $\mathbf{f}$ 的排序. 记将 Λ_N 上的 $\mathbf{f}$ 重排为 m^3 个大小为 $4n^3$ 的小十二面体 $\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)$ 后, 它们对应的长度为 $4n^3$ 的向量为 $\mathbf{f}^{i_1,i_2,i_3}$, 排列矩阵为 Q_N, 则有

$$Q_N\mathbf{f}=(\mathbf{f}^{0,0,0}\quad\mathbf{f}^{0,0,1}\quad\cdots\quad\mathbf{f}^{m-1,m-1,m-1})^T \tag{2.4}$$

此时对 $\mathbf{k}\in\Lambda_N$, 总对应某个小十二面体 $\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)$ 中一点 $\widehat{\mathbf{k}}$, 使得对任意 $\mathbf{j}\in\Lambda_N$ 有

$$g^N_{\mathbf{jk}}=e^{i\frac{\pi}{2}\frac{\mathbf{j}\cdot(m\widehat{\mathbf{k}}+\mathbf{P}_{i_1,i_2,i_3})}{mn}}=e^{i\frac{\pi}{2}\frac{\mathbf{j}\cdot\mathbf{P}_{i_1,i_2,i_3}}{N}}e^{i\frac{\pi}{2}\frac{\mathbf{j}\cdot\widehat{\mathbf{k}}}{n}}=g^N_{\mathbf{jP}i_1,i_2,i_3}g^n_{\mathbf{j}\widehat{\mathbf{k}}}.$$

从而对给定的 $\mathbf{k}\in\Lambda_N$, (1.3) 可以化为

$$F_{\mathbf{k}}=\sum_{i_1,i_2,i_3=0}^{m-1}\sum_{\mathbf{j}\in\Lambda(\mathbf{P}_{i_1,i_2,i_3},n,m)} g^N_{\mathbf{jk}}f^{i_1,i_2,i_3}_{\mathbf{j}}=\sum_{i_1,i_2,i_3=0}^{m-1} g^N_{\mathbf{P}_{i_1,i_2,i_3}\mathbf{k}}\sum_{\mathbf{j}\in\Lambda_n} g^n_{\mathbf{jk}}f^{i_1,i_2,i_3}_{\mathbf{j}}. \tag{2.5}$$

再来考察 (1.5) 的左端, 记依据区域分解后得到的小平行十二面体对应的向量分别为 $\mathbf{F}^{i_1,i_2,i_3}$, 置换矩阵为 P_N, 则有

$$P_N\mathbf{F}=(\mathbf{F}^{0,0,0}\quad \mathbf{F}^{0,0,1}\quad \cdots\quad \mathbf{F}^{m-1,m-1,m-1})^T. \tag{2.6}$$

利用周期性, 容易得到

$$g^n_{\mathbf{jk},\mathbf{k}\in\Lambda(\mathbf{C}_{i_1,i_2,i_3},n,1)}=g^n_{\mathbf{jk},\mathbf{k}\in\Lambda_n},\quad 0\le i_1,i_2,i_3\le m-1,\quad \mathbf{j}\in\Lambda_n, \tag{2.7}$$

$$g^N_{\mathbf{P}_{j_1,j_2,j_3}\mathbf{k},\mathbf{k}\in\Lambda_n(\mathbf{C}_{i_1,i_2,i_3})}=e^{i\frac{\pi}{2}\frac{\mathbf{C}_{i_1,i_2,i_3}\cdot\mathbf{P}_{j_1,j_2,j_3}}{N}}g^N_{\mathbf{P}_{j_1,j_2,j_3}\mathbf{k},\mathbf{k}\in\Lambda_n}, \tag{2.8}$$

$$0\le i_1,i_2,i_3,j_1,j_2,j_3\le m-1.$$

定义 $m^3\times m^3$ 阶常数矩阵 $T_{m,m,m}=(e^{i\frac{\pi}{2}\frac{\mathbf{C}_{i_1,i_2,i_3}\cdot\mathbf{P}_{j_1,j_2,j_3}}{N}})$, 综合 (2.5),(2.7) 和 (2.8), 对每个固定 $\mathbf{k}\in\Lambda_n$

$$\begin{pmatrix} F^{0,0,0}_{\mathbf{k}}\\ F^{0,0,1}_{\mathbf{k}}\\ \cdots\\ F^{m-1,m-1,m-1}_{\mathbf{k}}\end{pmatrix}=T_{m,m,m}\begin{pmatrix} g^N_{\mathbf{P}_{0,0,0}\mathbf{k}}\sum_{\mathbf{j}\in\Lambda_n} g^n_{\mathbf{jk}}f^{0,0,0}_{\mathbf{j}}\\ g^N_{\mathbf{P}_{0,0,1}\mathbf{k}}\sum_{\mathbf{j}\in\Lambda_n} g^n_{\mathbf{jk}}f^{0,0,1}_{\mathbf{j}}\\ \cdots\\ g^N_{\mathbf{P}_{m-1,m-1,m-1}\mathbf{k}}\sum_{\mathbf{j}\in\Lambda_n} g^n_{\mathbf{jk}}f^{m-1,m-1,m-1}_{\mathbf{j}}\end{pmatrix}. \tag{2.9}$$

记 $4n^3$ 阶对角阵 $D^{j_1,j_2,j_3}_n=Diag\{g^N_{\mathbf{P}_{j_1,j_2,j_3}\mathbf{k}}\}$, 记 $D_N=Diag\{D^{0,0,0}_n,D^{0,0,1}_n,\cdots,D^{m-1,m-1,m-1}_n\}$, 引入 Kronecker 乘积 $\bigotimes$, 由 (2.9) 可得到如下定理:

定理 2.1. $N=mn$ 时傅立叶变换矩阵 G_N 有如下的分解:

$$G_N=P_N^{-1}(T_{m,m,m}\bigotimes I_{4n^3})D_N(I_{m^3}\bigotimes G_n)Q_N. \tag{2.10}$$

由 (2.10), 利用 Kronecker 乘积性质进行推导, 即可将定理 2.1 推广至一般的 $N=N_0m^p$ 的情形.

推论 2.1. 对一般的 $N=N_0m^p$, 傅立叶变换矩阵 G_N 有如下的分解:

$$\begin{aligned}G_N=&\Big(\prod_{i=1}^{p}\{(I_{m^{3(i-1)}}\bigotimes P^{-1}_{\frac{N}{m^{i-1}}})(I_{m^{3(i-1)}}\bigotimes(T_{m,m,m}\bigotimes I_{\frac{4N^3}{m^{3i}}}))(I_{m^{3(i-1)}}\bigotimes D_{\frac{N}{m^{i-1}}})\}\Big)\\ &(I_{m^{3p}}\bigotimes G_{N_0})\Big(\prod_{i=1}^{p}\{I_{m^{3(p-i)}}\bigotimes Q_{\frac{N}{m^{p-i}}}\}\Big)\end{aligned} \tag{2.11}$$

以上定理及推论的结果在形式上与二维平行六边形区域的相应结果相近 [9].

§3. 算法实现

3.1 数据结构

基于平行十二面体区域的几何复杂性, 选择一个好的数据结构是上述算法成功的关键, 它必须能够方便快捷的存取 Λ_N 中的点. 据 (1.2), Λ_N 可以拆分为四个大小为 N^3 的平行六面体, 我们采用四元数组 (m,i,j,k) 表示 Λ_N, 其中 $m=1,2,3,4$, 标示四个平行六边体, $i,j,k=1,2,\cdots,N$ 标示了一个边长为 N 的平行六面体. 给定此四元数组中的一个点 (m,i,j,k), 它对应的六向坐标 $\mathbf{J}$ 为

if (m==1) $\mathbf{J}=(i-1,j-1,k-1,i-j,j-k,k-i)$,

else if (m==2) $\mathbf{J}=(i-k-1,i+j-N-2,i-N-1,N+1-j-k,j-1,k-N)$,

else if (m==3) $\mathbf{J}=(i+j-N-2,i-N-1,i-k,j-1,k-N-1,N+2-j-k)$,

else $\mathbf{J}=(i-N-1,i-j,i+k-N-1,j-N-1,N+1-j-k,k)$.

反之, 给定六向坐标 $(j_1,j_2,j_3,j_4,j_5,j_6)$, 它对应的四元数组 (m,i,j,k) 为

if $(j_1,j_2,j_3\geq 0)$ $(m,i,j,k)=(1,j_1+1,j_2+1,j_3+1)$,

else if $(j_3\leq -1,j_5\geq 0,j_6\leq 0)$ $(m,i,j,k)=(2,j_3+N+1,j_5+1,j_6+N)$,

else if $(j_2,j_5\leq -1,j_4\geq 0)$ $(m,i,j,k)=(3,j_2+N+1,j_4+1,j_5+N+1)$,

else if $(j_1,j_4\leq -1,j_6\geq 1)$ $(m,i,j,k)=(4,j_1+N+1,j_4+N+1,j_6)$.

从而建立了存储数组到六向坐标之间的对应关系.

和整体自然排序等数据结构相比, 这种方式具有结构简单、易于编程等优点.

3.2 数据排列

快速变换的本质是对数据的重新排序, 而排序中的难点在于如何利用周期性找到 Λ_N 外一点在 Λ_N 内的对应点. 即对空间任意一点 $\mathbf{j}=(j_1,j_2,j_3,j_1-j_2,j_2-j_3,j_3-j_1)$, 找到 i_1,i_2,i_3 和 $\mathbf{k}=(k_1,k_2,k_3,k_1-k_2,k_2-k_3,k_3-k_1)$, 使得 $\mathbf{j}=N\mathbf{C}_{i_1,i_2,i_3}+\mathbf{k}$. 我们给出如下算法:

算法 3.1 计算空间一点 $\mathbf{j}$ 在 Λ_N 中的对应点 $\mathbf{k}$.

步 1. 通过模运算计算 $\mathbf{r}$ 和 $\mathbf{l}$, 满足 $\mathbf{j}=N\mathbf{r}+\mathbf{l}$, $l_1,l_2,l_3\in[0,N-1]$;

步 2.

1. $r_1+r_2+r_3=4q$ 时,

$(i_1,i_2,i_3)=(r_1-q,r_2-q,r_3-q),(k_1,k_2,k_3)=(l_1,l_2,l_3)$

2. $r_1+r_2+r_3=4q-1$ 时,

if $l_1\geq l_2,l_3$ $(i_1,i_2,i_3)=(r_1+1-q,r_2-q,r_3-q),(k_1,k_2,k_3)=(l_1-N,l_2,l_3)$

else if $l_3>l_1,l_2$ $(i_1,i_2,i_3)=(r_1-q,r_2-q,r_3+1-q),(k_1,k_2,k_3)=(l_1,l_2,l_3-N)$

else $(i_1,i_2,i_3)=(r_1-q,r_2+1-q,r_3-q),(k_1,k_2,k_3)=(l_1,l_2-N,l_3)$ $(i_1,i_2,i_3)=(r_1-q,r_2+1-q,r_3-q),(k_1,k_2,k_3)=(l_1,l_2-N,l_3)$

3. $r_1+r_2+r_3=4q-2$ 时,

if $l_3 \le l_1, l_2$ $(i_1, i_2, i_3) = (r_1+1-q, r_2+1-q, r_3-q), (k_1, k_2, k_3) = (l_1-N, l_2-N, l_3)$

else if $l_1 < l_2, l_3$ $(i_1, i_2, i_3) = (r_1-q, r_2+1-q, r_3+1-q), (k_1, k_2, k_3) = (l_1, l_2-N, l_3-N)$

else $(i_1, i_2, i_3) = (r_1+1-q, r_2-q, r_3+1-q), (k_1, k_2, k_3) = (l_1-N, l_2, l_3-N)$

4. $r_1+r_2+r_3 = 4q-3$ 时,

$(i_1, i_2, i_3) = (r_1+1-q, r_2+1-q, r_3+1-q), (k_1, k_2, k_3) = (l_1-N, l_2-N, l_3-N)$

3.3 HFFT 算法

在上述工作的基础上, 我们给出 $N = N_0 m^p$ 时依据 (2.11) 计算离散傅立叶变换 $\mathbf{F} = G_N \mathbf{f}$ 的快速算法. 令

$$\mathbf{f}_0 = (I_{m^{3p}} \bigotimes G_{N_0}) \prod_{\lambda=1}^{p} \{I_{m^{3(p-\lambda)}} \bigotimes Q_{\frac{N}{m^{p-\lambda}}}\} \mathbf{f},$$

$$\mathbf{f}_\mu = (I_{m^{3(p-\mu)}} \bigotimes P^{-1}_{\frac{N}{m^{p-\mu}}}) \Big(I_{m^{3(p-\mu)}} \bigotimes (T_{m,m,m} \bigotimes I_{\frac{4N^3}{m^{3(p+1-\mu)}}}) \Big)$$

$$(I_{m^{3(p-\mu)}} \bigotimes D_{\frac{N}{m^{p-\mu}}}) \mathbf{f}_{\mu-1}, \mu = 1, \cdots, p$$

则由 (2.11) 知 $\mathbf{F} = G_N \mathbf{f} = \mathbf{f}_p$. 据此给出如下算法:

算法 3.2 $N = N_0 m^p$ 时计算 $\mathbf{F} = G_N \mathbf{f}$.

步 1. 将 Λ_N 上的 $4N^3$ 阶向量 $\mathbf{f}$ 按 m^{3p} 个小区域 Λ_{N_0} 排列为向量 $\widehat{\mathbf{f}}$:

$\widehat{\mathbf{f}} = \prod_{\lambda=1}^{p} \{I_{m^{3(p-\lambda)}} \bigotimes Q_{\frac{N}{m^{p-\lambda}}}\} \mathbf{f} = (\mathbf{f}^1 \mathbf{f}^2 \cdots \mathbf{f}^{m^{3p}})^T$, 其中 $\mathbf{f}^i$ 为 $4N_0^3$ 阶向量;

步 2. 直接法计算 m^{3p} 个 N_0 规模的离散傅立叶变换

$G_{N_0} = (g^{N_0}_{\mathbf{jk}})$

for i=1:1:m^{3p}

$\mathbf{f}_0^i = G_{N_0} \mathbf{f}^i$

end

步 3. 计算 m^3 阶常数矩阵 $T_{m,m,m} = (e^{i\frac{\pi}{2} \frac{\mathbf{C}_{i_1,i_2,i_2} \cdot \mathbf{P}_{j_1,j_2,j_3}}{m}}), i_1, i_2, i_3, j_1, j_2, j_3 = 0, \cdots, m-1$.

步 4. 进入 p 步循环:

for i=1:1:p

1. $D_{\frac{N}{m^{p-i}}} = Diag\{g^{\frac{N}{m^{p-i}}}_{\mathbf{P}_j \mathbf{k}}\}, j = 1, 2, \cdots, m^3; \mathbf{k} \in \Lambda_{\frac{N}{m^{p+1-i}}}$;

2. 计算 $\dfrac{4N^3}{m^{3(p-i)}}$ 阶置换矩阵 $P_{\frac{N}{m^{p-i}}}$;

3. for j=1:1:$m^{3(p-i)}$

$\widetilde{\mathbf{f}}_i^j = D_{\frac{N}{m^{p-i}}} * (\mathbf{f}_{p-i}^{(j-1)m^3+1} \cdots \mathbf{f}_{i-1}^{jm^3})^T$

$\widehat{\mathbf{f}}_i^j = (T_{m,m,m} \bigotimes I_{\frac{4N^3}{m^{3(p+1-i)}}}) * \widetilde{\mathbf{f}}_i^j$

$\mathbf{f}_i^j = P_{\frac{N}{m^{p-i}}} * \widehat{\mathbf{f}}_i^j$

end

end

§4. 并行算法

当数据规模增大，超出单机内存时，需要并行实现 HFFT 算法. 和 FFT 一样，HFFT 也是典型的数据密集型的计算，数据通讯在计算过程中占绝对的主导地位，获得高并行性能算法的关键在于如何尽可能减少数据通讯. 常用的提高 FFT 并行效率的方法有：1) 变换过程中不进行排序，即所谓的 Unsorted FFT，此时全局通讯量为 0，并行性能自然较高，但这种方法仅适合某些特定的无需排序的应用 [11]；2) 在共享存储计算机上采用多线程技术实现并行计算 [12]；3) 利用高效并行文件系统的 Out-Of-Core 技术 [13]. 和上述方法不同，我们的并行 HFFT 适用于分布式存储的并行计算机，计算数据规模不超出系统总内存. 同样，我们需要考虑如何减少数据通讯.

由 (2.10) 和 (2.11) 知，HFFT 的全局通讯包括两次排序，分别对应于乘以矩阵 P 和 Q. 注意到，(2.6) 表明，P_N 对应排序的目的是将一个大十二面体 $\Lambda(\mathbf{O}, N, 1)$ 排列为 m^3 个小十二面体 $\Lambda(\mathbf{C}_{i_1,i_2,i_3}, n, 1)$. 如果大十二面体原来就按照这样的顺序排列，则有 $P_N = I_N$. 与此类似，将每个较大的十二面体都按照它剖分得到的 m^3 个小十二面体的顺序递归排列，则 (2.10) 和 (2.11) 可改写为

$$G_N = (T_{m,m,m} \bigotimes I_{4n^3}) D_N (I_{m^3} \bigotimes G_n) Q_N, \tag{4.1}$$

$$\begin{aligned} G_N =& (\prod_{i=1}^{p} \{(I_{m^{3(i-1)}} \bigotimes (T_{m,m,m} \bigotimes I_{\frac{4N^3}{m^{3i}}})(I_{m^{3(i-1)}} \bigotimes D_{\frac{N}{m^{i-1}}})\})(I_{m^{3p}} \bigotimes G_{N_0}) \\ & (\prod_{i=1}^{p} \{I_{m^{3(p-i)}} \bigotimes Q_{\frac{N}{m^{p-i}}}\}). \end{aligned} \tag{4.2}$$

此时，HFFT 只需要一次全局排序，在基于 MPI 的实现中，它对应在每个处理器上执行一次 MPI_Alltoall 调用.

(4.1) 具有内在的并行性，它可以分为两步：第一步乘以对角矩阵 D 可以在本地执行；第二步乘以矩阵 $T_{m,m,m}$ 最多在 m^3 个处理器上执行，在基于 MPI 的实现中，它对应两次 MPI_Alltoall 调用.

算法 4.1 HFFT 的并行算法 (单个节点)

1. 读入或生成计算数据，规模为总数据量除以处理器个数.

2. 执行一次全局排序.

3. 计算规模不超过本节点范围的局部 HFFT 计算.

4. 完成节点间并行 HFFT 计算，每次计算最多由 m^3 个节点参与.

§5. 数值实验

我们在 Linux 环境下用 C 语言实现了上述串行和并行算法，并进行了数值实验. 图 3 到图 6 分别给出了一些实验结果. 串行程序运行于我中心自行搭建的机群 RDCPS-II, 单节点配置为 PIII 1G 处理器， 2G 内存. 并行程序运行于 973 "大规模科学计算研究" 项目机群系统 LSSC-II, 它包括 256 个计算节点，其中每个节点拥有两个 Intel 2GHz Xeon 处理器和 1GB 内存，网络传输系统为 Myrinet 2000.

测试数据取 (0,1) 区间上的随机数，最大误差取经过一次正变换和一次逆变换后得到的值对初始值的相对误差.

图 3 和图 4 的结果表明， $N = 2^p$ 和 3^p 时，使用直接法，计算时间分别以 N^6(64 和 729) 倍左右的速度增长，而用本文给出的快速算法计算时间只以 $N^3 \log N$(分别接近于 8 和 27) 的速度增长.

图 5 表明随着计算量的减少，舍入误差相应减少，使得计算精度有数量级的提高. 如表 1 中 $N = 32$ 时，快速算法的精度比直接法高 5 个数量级，且随着 N 的增大，计算精度的差异也越来越大.

图 6 和表 2 中，当 CPU 数目由 4 个增加到 32 个时，计算时间由 101.9584 秒减少到了 21.9811 秒. 其中本地计算时间 (排序及 HFFT 计算) 随着本地计算规模的减小呈线性降低. 通讯时间则逐渐趋于稳定，这是因为 Myrinet 的通讯延迟较小，通讯时间主要由通讯量决定，而当 N 给定，并行 HFFT 的通讯量随处理器的增多逐渐趋于常数. 处理器个数为 4 时，单个节点处理的数据规模较大，系统利用了缓存，这导致运行时间稍大.

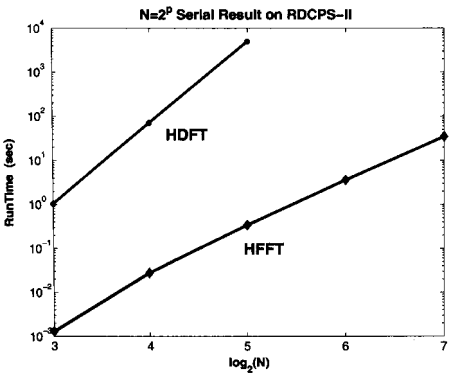

图 3 $N = 2^p$ 时串行计算时间对比

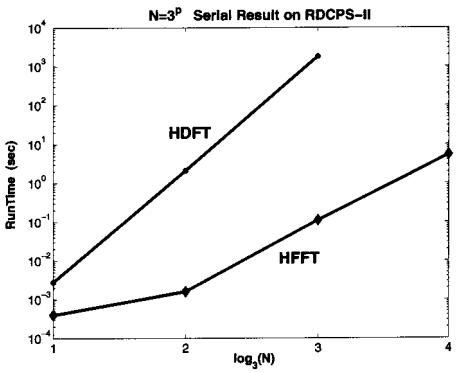

图 4 $N = 3^p$ 时串行计算时间对比

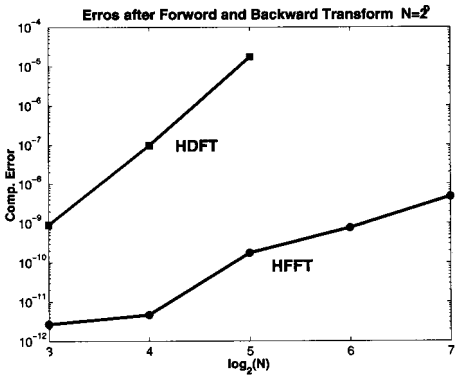

图 5　$N = 2^p$ 时串行计算误差对比

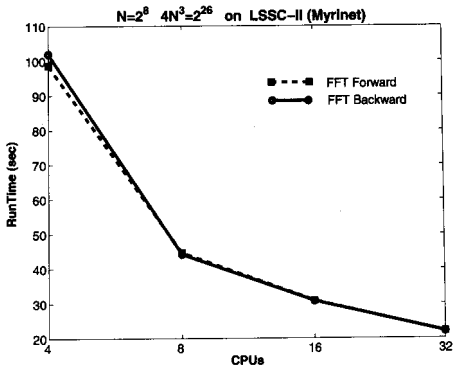

图 6　$N = 2^8$ 时并行计算结果

表 1　$N = 2^p$ 时最大计算误差对比

N	总点数	直接法	快速算法
8	2048	9.09233e-10	2.67963e-12
16	16384	9.72329e-08	4.72695e-12
32	131072	1.75461e-05	1.75852e-10
64	1048576	–	7.73052e-10
128	8388608	–	4.83871e-09

表 2　$N = 2^8$ 时并行 HFFT 单机运行时间（单位：秒）

CPU	总时间 (Forward)	通讯时间	本地计算时间
4	98.6230	29.8768	68.7462
8	44.6280	13.7874	30.8406
16	30.8549	13.4633	17.3916
32	21.9668	13.2962	8.6706

参 考 文 献

[1] E.O.Brigham,The Fast Fourier Transform and Its Applications. Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, 1988.

[2] SIAM News, 33(4), 2000.

[3] Mitsuo Yokokawa, Ken'ichi Itakura, Atsuya Uno,Takashi Ishihara and Yukio Kaneda,16.4-T flops Direct Numerical Simulation of Turbulence by a Fourier Spectral Method on the Earth Simulator,Supercomputing 2002.

[4] Sun Jiachang, Orthogonal piecewise polynomials basis on an arbitary triangular domain and its applications, Journal of Computational Mathematics 19 (2001) 55-66.

[5] Sun Jiachang, Multivariate Fourier Series over a Class of Non Tensor-product Partition Domains, Journal of Computational Mathematics 21:1 (2003).

[6] Huiyuan Li and Sun Jiachang, Generalized Fourier Approximations on Hexagons Based on 3-Directional Partition,The Annuals of RDCPS, ISCAS, 2002, http://www.rdcps.ac.cn.

[7] Sun Jiachang, m Dimension Fourier Series over Domains wiht $m+1$ Direction Mesh, The Annuals of RDCPS, ISCAS, 2002, http://www.rdcps.ac.cn.

[8] Sun Jiachang, Generalized Fourier transforms over parallel dodecahedron domains, The Annuals of RDCPS, ISCAS, 2002, http://www.rdcps.ac.cn.

[9] 孙家昶, 姚继锋, 平行六边形区域上的快速离散傅立叶变换, 计算数学, 26:3 (2004) 351-366.

[10] Sun Jiachang,Shi Xiquan,B-splines on 3-D tetrahedron partition in four-directional mesh, Science in China, 44:4 (2001).

[11] Izidor Gertner, Martin Rofhert,A Parallel Algorithm for 2-D DFT Computation with No Interprocessor Communication, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL1, No3, JULY 1990.

[12] Parilama Thulasiraman, Kevin B.Theobald, Ashfaq A. Khokhar, Guang R. Gao, Multithreaded Algorithms for the Fast Fourier Transform, Proceedings of the twelfth annual ACM symposium on Parallel algorithms and architectures, 2000.

[13] Thomas H. Cormen, David M. Nicol,Out-of-core FFTs with parallel disks,ACM SIGMETRICS Performance Evaluation Review, Volume 25 Issue 3, 1997.