国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第12卷第3期 1990年9月 Vol.12 No.3

矩阵转置的一种快速算法

邹庆云 黄新民 赵 玲
（系统工程与应用数学系） （自动控制系）

摘 要 本文给出一个数据体积超过计算机可用内存容量的$n\times n$矩阵快速转置的新方法。与常用的算法相比，所需的计算时间显著减小。

关键词 矩阵，算法，记录，转置，内存，分块
分类号 O241.6

对体积超过计算机内存容量的大型矩阵进行转置，是快速富里叶变换、二维离散富里叶变换及许多大型计算中常遇到的问题。关于这个问题的算法已有报道[1~4]。本文提出一种对于一般大型方阵快速转置的新算法，它比上面列举的算法都简单而速度更快，例如在IBM PC/XT微机上利用32×1024单元内存对1024×1024矩阵转置，所用时间只及[1]的56.8%．不过这里比[1]多占一倍的外存，这在通常是允许的，因为一般计算机的外存容量都比内存容量要大得多。

1 算 法

设A为$n\times n$方阵，以行为记录（行号为记录号）存放在磁盘文件“AD”中，计算机可用内存为M单元，$M\geqslant2n+2$．

首先确定整数m,p：$m\geqslant2$，$m>p\geqslant0$使

$$k=(n+p)/m$$

为整数，且

$$m(n+p)\leqslant M<(m+1)(n+p)$$

这样，$n+p$阶方阵$\begin{pmatrix}A&0\\0&0\end{pmatrix}$或$A$（$p=0$时）可写成

$$\begin{pmatrix}A_{11}&\cdots&A_{1k}\\\cdots&\cdots&\cdots\\A_{k1}&\cdots&A_{kk}\end{pmatrix}$$

其中k^2个分块$A_{ij}(i,j=1,\cdots,k)$均为$m\times m$方阵。

1987年6月20日收稿

接下来分两步完成 A 的转置：

第一步，分 k 次将 A 读入内存中 $m\times(n+p)$矩阵 B（即二维数组 B），B 的其余元素（即第$n+1$至$n+p$列）为零：

$$B=(A_{i1}\cdots A_{ik})\qquad(i=1,\cdots,k)$$

以分块为记录按行顺序依次将$A_{i1},\cdots,A_{ik}$写到新打开的磁盘文件“BD”的第$k(i-1)+1,\cdots,k(i-1)+k$个记录。

第二步，分 k 次每次将“BD”中相间为 $k-1$ 个记录的 k 个记录即第 $k(1-1)+j,\cdots,k(k-1)+j(j=1,\cdots,k)$记录按列顺序分别读入 B 的 k 个分块：

$$B=(A'_{1j}\cdots A'_{kj})$$

这里$A'_{1j},\cdots,A'_{kj}$ 分别为 $A_{1j},\cdots,A_{kj}$ 的转置。然后将 B 每行前 n 个元素按行顺序分别写到文件“AD”的第$m(j-1)+1,\cdots,m(j-1)+m$记录（第 k 次当$p>0$时只将 B 的前 $m-p$ 行的前 n 个元素写到“AD”文件）。

第二步完成时，磁盘文件“AD”中以行为记录存放的$n\times n$矩阵为

$$\begin{pmatrix}A_{11}' & \cdots & A_{k1}'\\ \cdots & \cdots & \cdots\\ A_{1k}' & \cdots & A_{kk}'\end{pmatrix}=\begin{pmatrix}A_{11} & \cdots & A_{1k}\\ \cdots & \cdots & \cdots\\ A_{k1} & \cdots & A_{kk}\end{pmatrix}'$$

或其左上角的$n\times n$分块，即A'．A转置完成。

2　计算时间估计

算法的计算时间包括做加、乘等运算的时间与内外存之间交换数据的时间，而加、乘运算在这个算法里只在确定记录号$k(i-1)+j$时有，远远少于数据交换占用的时间。另外，按一般算法（如[1]），内存与原始数据文件（如前面“AD”）交换数据的时间已包含在具体应用（如FFT等）的算法时间里。因此内存与中间数据文件“BD”交换数据的时间，即读写各一次的时间为所要估计的时间。

设将一个长度为 n 单元的记录输入和输出的时间分别为 T_{read}，T_{write}；又设访问一个记录的时间为T_{acc}；读或写记录中一个数据的时间为T_{tran}．则有

$$T_{\mathrm{read}}=T_{\mathrm{write}}=T_{\mathrm{acc}}+nT_{\mathrm{tran}}$$

这样算法的整个计算时间T_{count}也就是分k^2个记录读、写$(n+p)^2$个数据各一次的时间，即

$$T_{\mathrm{count}}=2k^2T_{\mathrm{acc}}+2(n+p)^2T_{\mathrm{tran}}$$

式中正整数

$$k=(n+p)/m$$

即

$$k=n/m\quad\text{或}\quad k=\left[\frac{n}{m}\right]+1$$

而[1]算法的计算时间为

$$T'_{\mathrm{count}}=2(k'-1)nT_{\mathrm{acc}}+2(k'-1)n^2T_{\mathrm{tran}}+\frac{1}{4}tn^2T_{\mathrm{chang}}$$

式中 t，k'为正整数。$t=\log_2 n$，$k'=\log_2 n/\log_2 m$

或 $$k'=[\log_m n]+1$$

两相比较，T'_{count}比T_{count}多一项$\frac{1}{4}tn^2T_{change}$，而第一项$2(k'-1)nT_{acc}$在$n/m$不大时大于$2k^2T_{acc}$，第二项$2(k'-1)n^2T_{tran}$则在$n/m$较大时大于$2(n+p)^2T_{tran}$. 总的结果，本文算法的计算时间在一般情况下都要显著少于[1]算法的计算时间，根据在 IBM PC/XT 微型机上对 $n=64,128,256,1024$；$m=2,4,8,16,32$各种情况的运行结果（见表1），运

表1 本文算法与[1]算法计算时间比较

矩阵体积 $n\times n$	内 存 $m\times n$	本算法 T_{count}（秒）	[1]算法 T'_{count}（秒）	比较 T_{count}/T'_{count}
1024×1024	32×1024	2506	4411	0.568
256×256	32×256	186.1	252.7	0.736
256×256	16×256	197.0	238.3	0.827
128×128	32×128	44.2	76.2	0.580
128×128	16×128	46.6	68.0	0.686
128×128	8×128	66.8	142.3	0.470
64×64	32×64	11.1	32.0	0.347
64×64	16×64	11.9	26.0	0.458
64×64	8×64	17.9	23.9	0.748
64×64	4×64	38.3	47.5	0.808
64×64	2×64	79.9	116.9	0.683

算速度提高20%～188%不等。例如：当$n=64$，$m=32$时，$T_{count}/T'_{count}=0.347$；当$n=1024$，$m=32$时，$T_{count}/T'_{count}=0.568$. 明显看得出提高了计算速度的地方是只对中间文件进行一次读写，而[1]相当于对中间文件进行了多次读写。

文[2]、[3]的算法改进了[1]，即将适用的矩阵范围由$2^t\times 2^t$阶推广到一般方阵和一般矩阵，其计算速度与[1]相当。本文的算法适用于一般方阵，也可直接推广到一般矩阵。

算法的提出与总结得到国防科大蒋增荣教授的指导与支持，在此深表谢意。

参 考 文 献

[1] Eklundh J O. A Fast Computer Method for Matrix Transposing. IEEE Trans. 1972,C-21(7):801

[2] Schumann U. Comment on A Fast Computer Method for Matrix Transposing and Application to the Solution's of Posson's Equation. IEEE Trans. 1973,C-22(5):542

[3] Allotop W O. A Computer Algorithm for Transposing a Nonsquare Matrix. IEEE Trans. 1975, C-24(10):1038

[4] 唐权钧. 矩阵转置的快速计算方法. 石油物探, 1983,(2)

（下转第85页）

Unified Definition of Divergence, Curl and Gradient Based on a Common Model and It's Application in a General Coordinate System

Xu Liping
(Department of Aerospace Technology)

Abstract

In this paper, the unified definition of divergence, curl and gradient independent of the coordinate system is given by the extended divergence theorem. Stressed on using a common model which is a volume element enclosed by the three pairs of coordinate surfaces, all the divergence, curl and gradient are denoted uniformly in the form of the limiting value of a differential quantity. So the known expressions of these three quantities in tensor analysis can be derived directly.

Key words: tensor analysis, divergence, gradient, curl, unified definition

(上接第78页)

A Fast Computer Algorithm for Matrix Transposing

Zou Qingyun　Huang Xinmin
(Department of Applied Mathematics and System Engineering)
Zhao Ling
(Department of Automatical Control)

Abstract

A new fast computer algorithm for the transposing of an $n \times n$ data matrix, which requires more internal storage than the available to a certain computer, is given in this paper. The required computation time is much less than the conventional algorithm.

Key words: matrix, algorithm, record, transpose, internal storage, block