

非对称广义特征值问题的并行 QZ 算法^{***}

薛长峰

(盐城工学院基础科学部, 盐城 224003)

摘 要 非对称广义特征值问题的并行处理是并行计算结构力学的重要部分, 这方面的工作目前在国内外研究得还很少。将串行计算中求解非对称广义值问题 $AX = \lambda BX$ (A, B 为 n 阶实矩阵) 行之有效的 QZ 算法^[1]作了并行化处理, 并就得到的并行 QZ 算法在 Challenge L 并行机上进行了试算。数值结果表明: 当处理机个数为 2 或 4 时并行效率良好, 但当处理机个数为 8 时并行效率不高。

关键词 广义特征值 并行处理 QZ 算法 并行 QZ 算法

分类号 O29

引 言

广义特征值问题 $AX = \lambda BX$ (A, B 是 n 阶实矩阵) 的并行处理是并行计算结构力学中的重要组成部分, 周树荃、邓绍忠、曾岚等人已对对称广义特征值问题的并行处理进行了深入而又系统的研究, 它们的工作包括: 并行子空间迭代法^[1], 并行块子空间迭代法^[2], 并行保域行列式查找法^[3], 并行保域多分法^[4], 并行 EBE 向量迭代法^[5], 并行 EBE-Lanczos 方法^[6]以及并行 EBE 子空间迭代法^[7], 相对于对称情形, 非对称广义特征值问题的并行处理, 目前在国内外研究得尚少, 能查阅到的文献不多。本文工作是将串行中公认有效的求解非对称广义特征值问题的 QZ 算法作并行化处理, 并将得到的并行 QZ 算法在 Challenge L 并行计算机上进行了试算。

1 QZ 算法

QZ 算法 1973 年由 C. B. Moler 和 G. W. Stewart 提出^[8]它是公认的计算非对称广义特征值问题 $AX = \lambda BX$ (A, B 为实矩阵) 的有效串行算法, 该方法是计算标准特征值问题的 QR 算法的推广, 当 $B=I$ 时, QZ 算法退化为 QR 算法。

考虑非对称广义特征值问题 $AX = \lambda BX$ 其中 A, B 为 n 阶实矩阵 (1.1)

根据[9]P133 的定理 5.7 及[10]P295 的定理 7.7.2 问题(1.1)可转化为问题(1.2)

$$AX = \lambda BX \quad (1.2)$$

其中 A 是上 Hessenberg 阵, B 是上三角阵。

QZ 算法共分二步, 第一步是对 A, B 的约化过程, 第二步是对 A, B 的迭代过程。

QZ 算法:

• 收稿日期: 1996-09-03

•• 国防科技预研与江苏省自然科学基金资助课题

1.1 约化 A 为上 Hessenberg 阵, B 为上三角阵

(1) 对 $k=1, 2, \dots, n-1$

a. 选择 $Q_k \in H_{n-(k+1)}(k)$ 去零化 $b_{k+1,k}, b_{k+2,k}, \dots, b_{nk}$

b. $B \leftarrow Q_k B \quad A \leftarrow Q_k A$

(2) 对 $k=1, 2, \dots, n-2$ 再对 $l=n-1, n-2, \dots, k+1$

a. 选取 $Q_l \in H_2(l)$ 消去 $a_{(+)k}$

$$(Q_l)_{ll} = (Q_l)_{l+1l+1} = p = -\frac{a_{lk}}{\sqrt{a_{lk}^2 + a_{l+1k}^2}}$$

$$(Q_l)_{l+1l} = -(Q_l)_{ll+1} = q = -\frac{a_{l+1k}}{\sqrt{a_{lk}^2 + a_{l+1k}^2}}$$

b. 计算 $A \leftarrow Q_{k+1} A \quad B \leftarrow Q_l B$

c. 选取 Z_l 消去 b_{l+1l}

$$(Z_l)_{ll} = (Z_l)_{l+1l+1} = r = -\frac{b_{l+1l}}{\sqrt{b_{l+1l}^2 + b_{l+1l+1}^2}}$$

$$(Z_l)_{l+1l} = -(Z_l)_{ll+1} = s = \frac{b_{l+1l}}{\sqrt{b_{l+1l}^2 + b_{l+1l+1}^2}}$$

d. 计算 $B \leftarrow B Z_l, A \leftarrow A Z_l$

1.2 迭代过程

对 $A_k, B_k, k=1, 2, \dots, A_1=A, B_1=B$

(1) 选取原点位移序列 $\{\tau_k\}, k=1, 2, \dots$

设 V_k, U_k 分别为 A_k 与 B_k 的右下角的二阶矩阵, 解关于 X 的二次方程 $|V_k - xU_k| = 0$ 取 τ_k 为接近于 $a_{nn}^{(k)}$ 的那个根。

(2) 对于 $k=1, 2, \dots$

a. 选取 $Q_k = Q_{n-1} \cdots Q_1, Q_l \in H_2(l) \quad l=1, \dots, n-1$ 使 $S_k = Q_k(A_k - \tau_k B_k)$ 为上三角阵。

b. 确定 $Z_k = Z_1, Z_2, \dots, Z_{n-1}, Z_l \in H_2(l) \quad l=n-1, \dots, 1$ 使 $B_{k+1} = Q_k B_k Z_k$ 为上三角阵。

c. 计算 $A_{k+1} = \tau_k B_{k+1} + S_k Z_k$ 。

这样便产生了矩阵序列 $\{A_k\}, \{B_k\}, k=1, 2, \dots$ 可以证明, 当 $k \rightarrow +\infty$ 时 $\{A_k\}$ 序列将趋于一个上三角形矩阵 A_+ , $\{B_k\}$ 序列也趋于一个三角形矩阵 B_+ , 用 B_+ 的主对角线元素分别对应除以 A_+ 的主对角线元素, 便得到了所要求的特征值。

2 QZ 算法的并行化

QZ 算法的并行化是对现有的串行 QZ 算法的并行化改造, 它包括约化过程的并行化和迭代过程的并行化, 下面分别予以介绍。

2.1 约化过程

(1) 并行计算 $Q_1 A = A_1, Q_2 A_1 = A_2, \dots, Q_{n-1} A_{n-2} = A_{n-1} = A', Q_1 B = B_1, Q_2 B_1 = B_2, \dots$

$Q_{n-1} B_{n-2} = B_{n-1} = B',$ 其中 $Q_k = I + VU^T$ 记 $R = \text{Sign}(b_{kk})(b_{kk}^2 + b_{k+1k}^2 + \dots + b_{nk}^2)^{\frac{1}{2}}$

$$V_j=0, j=1, 2, \dots, k-1, V_k=-\frac{b_{kk}+R}{R}, V_j=-\frac{b_{jk}}{R} (j=k+1, \dots, n)$$

$$U_j=0, j=1, 2, \dots, k-1, U_k=1, U_j=\frac{V_j}{V_k} (j=k+1, \dots, n) (k=1, 2, \dots, n-1)$$

(2) 对 $k=1, 2, \dots, n-2$

a. 选择 Q'_k 去零化 $a'_{k+2k}, a'_{k+3k}, \dots, a'_{nk}$, 其中 $Q'_k=I+VU^T$

记 $R=\text{Sign}(a'_{k+1k})(a'^2_{k+1k}+a'^2_{k+2k}+\dots+a'^2_{nk})^{\frac{1}{2}}$

$$V_j=0, j=1, 2, \dots, k, V_{k+1}=-\frac{a'_{k+1k}+R}{R}, V_j=-\frac{a'_{jk+1}}{R} (j=k+2, \dots, n)$$

$$U_j=0, j=1, 2, \dots, k, U_{k+1}=1, U_j=\frac{V_j}{V_{k+1}} (j=k+2, \dots, n)$$

b. 并行计算 $Q'_k B' = B'', Q'_k A' = A''$ 此时 B'' 为 H 阵。

c. 用 $Z=Z_n \cdots Z_{k+2}$ 右乘 B'' 将 B'' 恢复为上三角阵

$$Z_{n-j+1} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & r & -s \\ & & s & r \end{bmatrix} \begin{matrix} n-j \text{ 行} \\ n-j+1 \text{ 行} \end{matrix} \quad j=1, \dots, n-k+1$$

$$r = -\frac{b_{n-j+1, n-j+1}}{\sqrt{b_{n-j+1, n-j}^2 + b_{n-j+1, n-j+1}^2}}, \quad s = \frac{b_{n-j+1, n-j}}{\sqrt{b_{n-j+1, n-j}^2 + b_{n-j+1, n-j+1}^2}}$$

其中 b_{ij} 为矩阵 $BZ_n \cdots Z_{n-j+2}$ 的元素。

d. 计算 $A''Z, B''Z$ 。

2.2 迭代过程

对 $A_k, B_k, k=1, 2, \dots, A_1=A, B_1=B$

(1) 选取原点位移序列 $\{\tau_k\}, k=1, 2, \dots$ 解一元二次方程

$$\begin{vmatrix} a_{n-1n-1}^{(k)} - xb_{n-1n-1}^{(k)} & a_{n-1n}^{(k)} - xb_{n-1n}^{(k)} \\ a_{nn-1}^{(k)} - xa_{nn-1}^{(k)} & a_{nn}^{(k)} - xb_{nn}^{(k)} \end{vmatrix} = 0$$

选接近于 $a_{nn}^{(k)}$ 的那个根作为 τ_k 。

(2) 选取 Q_k 使 $S_k=Q_k(A_k-\tau_k B_k)$ 为上三角阵, 对 $C_k=A_k-\tau_k B_k$, 用 $Q_k=Q_{n-1}Q_{n-2} \cdots Q_1$ 使 $Q_k C_k$ 为上三角阵,

$$Q_i = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & p & -\bar{q} \\ & & q & \bar{p} \end{bmatrix} \begin{matrix} i \text{ 行} \\ i+1 \text{ 行} \end{matrix}$$

$$p = -\frac{\bar{C}_{ii}}{\sqrt{|C_{ii}|^2 + |C_{i+1i}|^2}}, q = -\frac{C_{i+1i}}{\sqrt{|C_{ii}|^2 + |C_{i+1i}|^2}} (i=1, 2, \dots, n-1)$$

(3) 选取 Z_k , 使 $B_{k+1}=Q_k B_k Z_k$ 为上三角阵, 对 $Q_1 B$, 找 Z_1 使 $Q_1 B Z_1$ 上三角化

$$Z_1 = \begin{bmatrix} r & -\bar{s} \\ s & \bar{r} \\ & & 1 \end{bmatrix} \begin{matrix} 1 \text{ 行} \\ 2 \text{ 行} \end{matrix}, r = -\frac{b_{22}}{\sqrt{|b_{21}|^2 + |b_{22}|^2}}, s = \frac{b_{21}}{\sqrt{|b_{21}|^2 + |b_{22}|^2}}$$

对 $Q_2 Q_1 B Z_1$ 找 Z_2 , 使 $Q_2 Q_1 B Z_1 Z_2$ 上三角化,

$$Z_2 = \begin{bmatrix} 1 & & & \\ & r & -\bar{s} & \\ s & & \bar{r}_1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \begin{matrix} 2 \text{ 行} \\ 3 \text{ 行} \end{matrix}$$

$$r = -\frac{b_{33}}{\sqrt{|b_{32}|^2 + |b_{33}|^2}}, s = \frac{b_{32}}{\sqrt{|b_{32}|^2 + |b_{33}|^2}}, \dots$$

对 $Q_i Q_{i-1} \dots Q_1 B Z_2 \dots Z_{i-1}$, 找 Z_i , 使 $Q_i Q_{i-1} \dots Q_1 B Z_1 Z_2 \dots Z_{i-1} Z_i$ 使上三角化

$$Z_i = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & r & -\bar{s} \\ & & s & \bar{r}_1 \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \begin{matrix} j \text{ 行} \\ j+1 \text{ 行} \end{matrix}$$

$$r = -\frac{b_{i+1i+1}}{\sqrt{|b_{i+1i+1}|^2 + |b_{i+1i}|^2}}, s = \frac{b_{i+1i}}{\sqrt{|b_{i+1i+1}|^2 + |b_{i+1i}|^2}} \dots i = 1, 2, \dots, n-1$$

(4) 计算 $A_{k+1} = \tau_k B_{k+1} + S_k \tau_k$

(5) 收缩

在每做完一次 QZ 迭代以后, 检查一下 A 的次对角线元素的值, 若某元素相对于相邻的元素可以忽略, 例如, 有 l , 使 $|a_{l+1l}| < \epsilon(|a_{ll}| + |a_{l+1l+1}|)$ (ϵ 为充分小的正数) 则可将 a_{l+1l} 看作为零, 此时 (A, B) 的特征值问题变成了两个独立的子问题特征值问题, 而下面所做的 QZ 迭代也完全可以分开独立运行。

3 数值试验及讨论

我们的算法是在北京应用物理与计算数学研究所 Challenge L 并行计算机上试算的, Challenge L 是一个由 8 个 CPU 组成的具有共享存储的服务系统, 该系统分别以 PowerFortran 语言 (PFL) 和 PowerFortran 加速器 (PFA) 作为编程语言和开发工具, 其中 PFA 是在 Fortran77 语言上的扩充, 主要增加一些高层指令, 用来实现 do 循环的并行化, 而 PFA 是辅助编程的开发工具, PFA 对源程序进行预处理和分析, 识别可并行化的 do 循环, 自动插入并行命令, 将串行程序转换为并行程序, 关于 Challenge L 及 PFA 和 PFL 的详细介绍可参阅文献 [11]。

下面, 我们用表 1 列出并行 QZ 算法在 Challenge L 上的运行结果, 在该机上我们就阶数 N 为 640 的矩阵进行了试算, 其中的矩阵对我们随机选取的稠密矩阵对。

表 1 $N=640$, 时间单位: s

P(CPU 个数)	1	2	4	8
运行时间	1427.350	1151.090	637.200	464.510
系统时间	47.7800	41.3200	24.9100	18.7600
总时间	1475.130	1192.410	662.110	483.270
加速比	1.000	1.240	2.240	3.073

在算法的实际实施过程中, 我们主要考虑了约化过程的并行化和迭代过程中收缩时的并

行化,在约化过程中我们将第2步中 a 部分的运算作适当的合并,这样做是为了加大第2步 b 部分矩阵乘的运算量,从而有利于 b 部分的并行运算。而对QZ迭代过程我们仍然执行串行算法,这是因为:第一,分析迭代过程,其中的每步运算前后相互依赖,数据相关性很大,第二,各步计算,主要花销是在矩阵乘法的计算上,而这里的矩阵乘是由一个矩阵与一个Givens阵的左乘或右乘,运算量不大,考虑到调用并行的开销相对较大,因此在迭代步使用串行运算,因而整个过程实际使用的是一种串并行混合算法。

表1中的加速比反映了本算法的并行化程度。对640阶这种规模的问题使用2个CPU的效率是62%,使用4个CPU的效率是56%,对非对称广义特征值问题达到这样的效率还是可以的,但使用8个CPU时的效率是38.41%,这样的结果令人难以满意,这一方面说明QZ算法内蕴的串行性较强,另一方面也表明探讨其它更有效的求解广义特征值问题的并行算法的必要性。而关于其它的一些高效并行算法(如拟Jacobi方法等)将另撰文发表。

作者衷心感谢周树荃教授的指导并感谢北京应用物理与计算数学研究所提供的赞助。

参考文献

- 1 曾岚,周树荃.并行子空间迭代法.中国工业与应用数学学会第三次大会论文集.清华大学出版社.1994,704~708
- 2 周树荃,曾岚.并行分块子空间迭代法.中国计算数学学会第五届年会论文集.1995
- 3 周树荃,曾岚.求解广义特征值问题的并行保域行列式查找法.南京航空航天大学学报.1995,27(2),147~155
- 4 Zeng Lan, Zhou Shu Quan, parallel region-preserving multi-section method for solving generalized eigenproblem, International Conference On Parallel Algorithms, 1995, Wuhan China
- 5 邓绍忠,周树荃.广义特征值问题的并行EBE向量迭代法.计算结构力学及其应用.1994,11(4),401~407
- 6 周树荃,邓绍忠.广义特征值问题的EBE-Lanczos并行算法.工程力学.1995,12(1),116~122
- 7 邓绍忠,周树荃.大型结构特征值问题的并行EBE子空间迭代法.南京航空航天大学学报.1994,26(5),575~581
- 8 C. B. Moler and G. W. Stewart (1973), An algorithm for generalized matrix eigenvalue problems, SIAM J. Num. Anal. 10, 241~256
- 9 周树荃,孙克明.数值代数.航空专业教材编审组.1985
- 10 G. H. Golub and C. Van. Loan Matrix computations, John Hopkins university press, Baltimore, 1984(中译本:廉庆荣等译.矩阵计算.大连理工大学出版社.1988)
- 11 孙绍麟编著. Power Fortran 语言和 Power Fortran 加速器用户指南.北京应用物理与计算数学研究所.1994