第22卷 第3期 自 动 化 学 报 Vol. 22, No. 3
1996 年 5 月 ACTA AUTOMATICA SINICA May, 1996

大型 Lyapunov 方程的并行求解[1)]

陈陆平 席裕庚 张钟俊
(上海交通大学自动控制系 上海 200030)

摘 要

借鉴于求解大型矩阵主特征对方法中子空间迭代的概念，给出了求解 Lyapunov 方程的新方法，进而推导出 Lyapunov 方程直接迭代的高效并行算法，同时也给出了算法的收敛性证明和解的误差分析.

关键词: Lyapunov 方程，并行计算，特征子空间，迭代法.

1 引言

Lyapunov 方程在控制系统分析和设计的数值计算中占有十分重要的地位. 低阶 Lyapunov 方程的求解方法研究已经取得了许多成果[1]. 但对于大型 Lyapunov 方程，由于受计算机内存容量的限制以及对计算精度和计算速度的要求，这些已有的方法往往难以奏效. 最近，许多工作对求解大型和超大型 Lyapunov 方程进行了专门研究，其中多为求部分解或直接找解的主特征对的近似方法[2]. 目前并行算法被认为是解决大型计算问题的最有效途径之一，但如何设计高效并行算法仍是期待解决的焦点问题. 大型 Lyapunov 方程的求解在利用已有算法实现并行化的基础上，还需要研究新的高效并行算法[3].

传统的 Lyapunov 方程求解方法[1]中变换求解方法被认为是解决小型密集矩阵的最有效手段，它们通常利用一些矩阵特征分析的解析手段和方法，对方程进行变换和化简后再求解. 一般来说，变换求解法对计算机的内存要求比较高，而且预处理过程中容易出现难以控制的积累误差，特别是在方程维数很高的情况下，对解的准确性有较大的影响. 另外，在有复特征值的情况下，实数 Lyapunov 方程也要在复数域内进行运算，这些都不利于求解大型 Lyapunov 方程，加上算法的复杂性，给变换求解法的并行实现也带来了困难. 在变换求解法中，利用实 Schur 分解的方法应用最广，也是目前研究 Lyapunov 方程并行算法的基础[1]. 文[2]及所引文献综述了这方面的情况.

对于大型矩阵的计算问题，像线性方程组的求解以及矩阵特征对的求解等，迭代法一直被认为是有效的途径. 其优点是方法简单，所需计算空间小，可以利用矩阵的稀疏性等结构特性，并且可以控制解的误差. 另外，由于其计算过程主要为矩阵运算，在时间域和

---

1) 国家自然科学基金、国家教委资助回国留学人员和优秀青年教师基金资助课题.
本文于1994年6月2日收到.

空间域运算操作量均衡,在多处理器的条件下,无论是动态分配还是静态分配都容易达到负载平衡,所以目前也被广泛作为研究高效并行算法的基础[4,5]. 同样,对于大型 Lyapunov 方程的求解,矩阵迭代法以其矩阵运算易于高度并行的优点,又重新引起了人们的重视. 通常这是一种对称双线性变换的方法,考虑了满足解的对称性,但增加了计算的复杂度[1].

这里利用 Lyapunov 方程可以转化为其扩展矩阵特征问题的特点,借鉴大型矩阵求解特征对的子空间迭代法,给出了求解 Lyapunov 方程的子空间迭代法,进而演化为 Lyapunov 方程直接迭代的高效并行算法. 前者通过增加空间维数减少了整体的时间复杂性,最大并行度高,符合构造并行算法树要"浅而宽"的原则;后者与原有的迭代法相比,算法简洁直观、计算量小,并且参加迭代运算的矩阵维数基本一致,因而负载平衡、计算成本低,易于构成高效的并行算法. 所以,两种并行算法各有所长.

2　Lyapunov 方程的特征子空间并行迭代法

考虑 Lyapunov 方程

$$XA + A^TX = -C, \tag{1}$$

其中 $A,X,C \in R^{n\times n}$,C 为正定矩阵,A 的特征值实部为负.
为叙述方便给出如下定义:

定义 1. Lyapunov 方程的扩展矩阵是指

$$H = \begin{bmatrix} A & 0 \\ -C & -A^T \end{bmatrix}.$$

此外,定义符号: $\sigma(\bullet)$——特征值集合; $\inf(\bullet)$——集合的下确界; $\sup(\bullet)$——集合的上确界; $\mathrm{real}(\bullet)$——复变量的实部; $\mathrm{imag}(\bullet)$——复变量的虚部.

引理 1[7]. 在复数域内,对于 $n\times n$ 阶满秩矩阵 K, 如果可以找到 $r\times r$ 阶满秩矩阵 F 和 $n\times r$ 阶满秩矩阵 $\Phi(r<n)$ 满足 $K\Phi=\Phi F$, 则 Φ 为 K 的 r 维特征子空间, F 为 K 在特征子空间 Φ 上的映射矩阵, F 所包含的特征值是 K 相应于 Φ 的那一部分.

引理 2[7]. (移轴定理) $n\times n$ 阶满秩矩阵 $K-\lambda_0 I$ 与 K 具有相同的特征空间. 其中 I 为 $n\times n$ 阶单位矩阵; λ_0 为原点移动量,并且若 K 的特征值为 $\lambda_i(i=1\sim n)$, 则 $K-\lambda_0 I$ 的特征值为 $\lambda_i-\lambda_0(i=1\sim n)$.

有了以上两个引理, 便可以采用矩阵特征子空间迭代构造 Lyapunov 方程的求解方法.

定理 1. Lyapunov 方程(1)的解 X 可以由其扩展矩阵相应的特征空间构成,即若

$$\begin{bmatrix} A & 0 \\ -C & -A^T \end{bmatrix}\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} A, \tag{2}$$

则 $X = Z_2 Z_1^{-1}$. 其中 A 的特征值的实部小于零,并且 A^{-1} 存在.

证明. 把(2)式展开,有

$$AZ_1 = Z_1 A \Longleftrightarrow AZ_1^{-1} = Z_1^{-1}A, \tag{3}$$

$$-CZ_1 - A^TZ_2 = Z_2 A. \tag{4}$$

将式(2)两边右乘 Z_1^{-1}，有

$$-C - A^T Z_2 Z_1^{-1} = Z_2 A Z_1^{-1}. \tag{5}$$

记 $X = Z_2 Z_1^{-1}$，则由式(3),(5)可知X满足方程(1).

分析式(2)可知，Z是 $2n$ 阶扩展矩阵H的 n 维特征子空间，其在H中对应的特征值与 $\sigma(A)$ 相同.

对于大型矩阵求解Z最有效的途径之一是子空间迭代法，其基本型是求主特征空间的方法，并且可以统一归纳为下列迭代格式：

迭代方法 1， $A\Phi^{(i+1)} = \Phi^{(i)}R^{(i)}$，$i = 1,2\cdots$； (6)

迭代方法 2， $A\Phi^{(i)} = \Phi^{(i+1)}R^{(i)}$， $i = 1,2\cdots$. (7)

迭代方法1的结果收敛至最小主特征子空间，记之为逆向迭代；迭代方法2的结果收敛至最大主特征子空间，记之为正向迭代. 在迭代格式中 $R^{(i)}$ 的不同选择是各种方法的区别所在，它的作用包括对X的各列正交化、归一化以及加速迭代的收敛等，因而往往它包含有多个复杂过程. 其中子空间求解的直接迭代法[12]，从式(2)出发，利用广义相似变换构造了迭代过程中的 $F^{(i)}$，即 $F^{(i)} = \Phi^{(i)+}A\Phi^{(i)}$. 其中 $\Phi^{(i)+} = (\Phi^{(i)^T}\Phi^{(i)})^{-1}\Phi^{(i)^T}$ 为广义逆，并取 $R^{(i)} = F^{(i)}$. 数值计算表明，直接迭代法运算操作简单、积累误差小、收敛精度高，并保持了式(6)的收敛速度，所以适于高阶矩阵的计算.

在特征子空间迭代方法中，主空间是指原矩阵模最大的一部分特征值所对应的特征向量构成的子空间. 分析H的特征谱可知，$\sigma(H) = \sigma(A)\cup\sigma(-A^T)$. 所以$A$的特征值在扩展矩阵$H$中不是"优势"特征值. 但原问题的解只与扩展矩阵的主特征向量矩阵有关，所以可采用正向迭代，并结合极点移轴的方法建立新的 Lyapunov 方程迭代解法.

设扩展矩阵原点移轴至实轴上 $\lambda_0 > 0$ 点，则 Lyapunov 方程的解也可以由 $\tilde{H} = H - \lambda_0 I$ 的主特征子空间Z构成，并满足下式：

$$\begin{bmatrix} A - \lambda_0 I & 0 \\ -C & -A^T - \lambda_0 I \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} (A - \lambda_0 I). \tag{8}$$

$\tilde{H}$ 的特征值由两部分组成：其一与矩阵 $A - \lambda_0 I$ 的特征值相同；其二与 $-A - \lambda_0 I$ 的特征值相同。下文将证明，如果合理选择 λ_0 可以使 $A - \lambda_0 I$ 和 $-A - \lambda_0 I$ 的特征值分离，即

$$\min\{|\lambda_i|, \lambda_i \in \sigma(A - \lambda_0 I)\} > \max\{|\lambda_i|, \lambda_i \in \sigma(-A - \lambda_0 I)\},$$

进而使Z成为主特征子空间. 所以能够保证所建立的求解 Lyapunov 方程的特征子空间迭代算法是收敛的. 这样，n 维 Lyapunov 矩阵方程的求解可以通过 $2n$ 维扩展矩阵H的 n 维特征子空间迭代来完成. 其并行算法如下：

1) 给定初始值 $Z^{(0)}$、移轴量 λ_0 和允许误差 ε^*；

2) 并行计算 $a_{ii} \pm \lambda_0 (i = 1\cdots n) \Longrightarrow \tilde{H} = H - \lambda_0 I$；

3) 并行计算 $(R^{(i-1)})^{-1} = (F^{(i-1)})^{-1} = (Z^{(i-1)+}\tilde{H}Z^{(i-1)})^{-1}$；

4) 并行计算 $Z^{(i)} = \tilde{H}Z^{(i-1)}R^{(i-1)^{-1}}$；

5) 并行计算 $\varepsilon = \|(H - \lambda_0 I)Z^{(i)} - Z^{(i)}F^{(i)}\|$；

6) 如果 $\varepsilon < \varepsilon^*$，则 $X = Z_2^{(i)}Z_1^{(i)^{-1}}$；否则转向 3).

上述迭代格式中保留了 $R^{(i)}$ 的标记. 这样，它还可以根据某些特殊要求，采纳其它

的特征子空间迭代方法。

迭代过程中主要包含矩阵运算。关于矩阵的并行计算问题已有了充分的研究[7]，这里不再重复。利用常用的内积并行算法，并考虑H的稀疏性对上述算法的估计可知[8]，其最大并行度为 $5n^3$，计算步为 $\log^3 n + 4\log n + 7$，平均并行度为 $O(n^3/\log^2 n)$。因而对于大型 Lyapunov 方程，该算法是高效并行算法。同时，根据以上方法还可以推导出计算成本更低的大型 Lyapunov 方程的直接迭代并行算法。

3 Lyapunov 方程的并行直接迭代解法

首先由特征子空间迭代方法来构造 Lyapunov 方程的直接迭代算法。求解 Z 的子空间迭代格式为

$$\begin{bmatrix} A-\lambda_0 I & 0 \\ -C & -A^T-\lambda_0 I \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}^{(i)} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}^{(i+1)} R^{(i)}. \tag{9}$$

展开上式有

$$(A-\lambda_0 I)Z_1^{(i)} = Z_1^{(i+1)}R^{(i)}, \tag{10}$$

$$-CZ_1^{(i)} - (A^T+\lambda_0 I)Z_2^{(i)} = Z_2^{(i+1)}R^{(i)}. \tag{11}$$

由式(10)可得

$$R^{(i)}Z_1^{(i)^{-1}} = Z_1^{(i+1)^{-1}}(A-\lambda_0 I). \tag{12}$$

将式(11)两边分别乘以 $Z_1^{(i)^{-1}}$，则有

$$-C-(A^T+\lambda_0 I)Z_2^{(i)}Z_1^{(i)^{-1}} = Z_2^{(i+1)}R^{(i)}Z_1^{(i)^{-1}}. \tag{13}$$

由式(12)代入上式右端得

$$Z_2^{(i+1)}R^{(i)}Z_1^{(i)^{-1}} = X^{(i+1)}(A-\lambda_0 I). \tag{14}$$

所以

$$-C-(A^T+\lambda_0 I)X^{(i)} = X^{(i+1)}(A-\lambda_0 I). \tag{15}$$

式(15)称为X的直接迭代算法。可以看出它与主特征子空间的迭代算法 (9) 有相同的收敛性。而且同样，迭代方程式(15)只包含矩阵的运算操作，因而容易实现高度并行化，具体步骤：

1) 选定移轴量 λ_0，给出初值 $X^{(0)}$，和收敛误差 ϵ^*；

2) 并行计算 $a_{ii}\pm\lambda_0(i=1\sim n)$，产生 $\underline{A}=-A^T-\lambda_0 I$ 和 $\bar{A}=-A+\lambda_0 I$；

3) 并行计算矩阵乘积 $\underline{A}X^{(i)}$；

4) 并行计算矩阵加法 $-C+\underline{A}X^{(i)}$；

5) 并行求解线性方程组 $X^{(i+1)}\bar{A}=-\underline{A}X^{(i)}+C$，解得 $X^{(i+1)}$；

6) 如果 $\|X^{(i+1)}-X^{(i)}\|>\epsilon^*$，返回 3)；否则结束迭代 $X=X^{(i+1)}$。

在 5) 中考虑到求逆产生的误差对最后结果的影响，因而采用了并行计算线性方程组的方法。具体计算过程中的简便方法是首先对 $\bar{A}$ 进行 LU 分解，以后每次迭代中只需回代求解即可。第一个迭代过程的计算步骤为 $4+\log n+2\log^2 n+\log^3 n$，最大并行度为 n^3。如果 n 阶矩阵的加法和乘法皆按实际中常采用的 n^2 或 $O(n)$ 并行度，容易看出 3)，4)，5)皆可达到负载平衡，因而计算效率在 $O(l)$ 左右，是高效的并行算法。

由于迭代法的计算量受迭代次数的影响事先难以估计，而目前没有同类方法进行加速比的相互比较，似乎对 Lypunov 方程的直接迭代并行算法的优劣难以评述。但要说明的是：①从串行角度来看，对于矩阵特征分析、控制中代数方程的特征空间解法与类似的迭代法和其他解析方法相比，在精度和速度等方面都更胜一筹，特别是对于大型问题，迭代法更为有效；②目前解决 n 阶 Lypunov 方程的方法包括已经实现的并行算法，并行度都受 n^2 阶矩阵的运算所制约，而实现本文方法的最大平均并行度也在于能否利用高并行度的 n^2 阶矩阵运算；③在运算的主要步骤中可采用的处理器个数都为 $O(n^2)$ 到 $O(n^3)$ 阶，因而保证了处理器的负载平衡，并且不需过多的协调。

4 算法的收敛性和误差分析

4.1 收敛性证明

通过分析式(9)和式(15)可知，直接迭代的收敛取决于特征子空间迭代法(7)的收敛性。而特征子空间迭代法的收敛规律是：设H的 $2n$ 个特征值依次为 $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_{2n}|$，若有 $|\lambda_r| > |\lambda_{r+1}|$，则 r 维子空间的正向迭代收敛于相应 $\lambda_1 \cdots \lambda_r$ 的特征子空间，而 $n-r$ 维子空间的逆向迭代收敛于相应 $\lambda_{r+1} \cdots \lambda_n$ 的特征子空间，两者的收敛速度都为 $O(|\lambda_r|/|\lambda_{r+1}|)$。所以 Lyapunov 迭代的收敛性根本上决定于是否可以通过移轴使 $\inf(|\sigma(\bar{A})|) > \sup(|\sigma(\underline{A})|)$。

设 $\sigma(A) \in \Omega_1, \sigma(-A^T) \in \Omega_2$，$\Omega_1$ 和 Ω_2 分别为复平面上以原点为圆心、r 为半径的圆型域的左、右各半圆。因为A的特征值在左半特征平面上，不妨设 $\inf(|\mathrm{real}(\sigma(A))|) = \alpha$，则当移轴至实轴上 $\lambda_0 > \alpha$ 点时，Ω_2 至新极点 λ_0 的最远距离 $\sup(|\underline{A}|) = ((\lambda - \alpha)^2 + (r^2 - \alpha^2))^{1/2}$，而 Ω_1 至新极点 λ_0 的最近距离为 $\inf(|\bar{A}|) = \lambda_0 + \alpha$。若要 $\inf(|\sigma(\bar{A})|) > \sup(|\sigma(\underline{A})|)$，则可推出 $\lambda_0^2 - 2\lambda_0\alpha + \alpha^2 + r^2 - \alpha^2 < \lambda_0^2 + 2\lambda_0\alpha + \alpha^2$，即 $4\lambda_0\alpha > r^2 - \alpha^2$，则当 $\lambda_0 > (r^2 - \alpha^2)/4\alpha$，算法是收敛的。但并不是 λ_0 选择的越大越好，因为算法收敛速度为

$$O(\inf(|\sigma(\bar{A})|)/\sup(|\sigma(\underline{A})|)),$$

所以合理选择 λ_0 可以提高算法的收敛性。

4.2 误差分析

Lyapunov 方程的子空间迭代法的误差范围可以由矩阵特征子空间迭代法的误差分析得到，因而下面主要分析直接迭代法的误差。

用X表示真实解，采用收敛判据

$$\|X^{(i+1)} - X^{(i)}\| \leqslant \varepsilon^*.$$

迭代中，基于 $X^{(i)}$ 求 $X^{(i+1)}$，可以有两种方式：

若用 $X^{(i+1)} = (-C - (A^T + \lambda_0 I)X^{(i)})(A - \lambda_0 I)^{-1}$ 求 $X^{(i+1)}$，设 $\bar{A}(\bar{A})^{-1} - I = e$，则有

$$X^{(i+1)}(A - \lambda_0 I) = (-C - (A^T + \lambda_0 I)X^{(i)})(I + e);$$

又

$$\|X^{(i+1)} - X^{(i)}\| \leqslant \varepsilon^* \Longleftrightarrow \|X^{(i)}(A - \lambda_0 I) + (C + (A^T + \lambda_0 I)X^{(i)})(I + e)\|$$

$$\leqslant \|(A-\lambda_0 I)\| \cdot \varepsilon^*,$$

$$\|X^{(i)}(A-\lambda_0 I)+C+(A^T+\lambda_0 I)X^{(i)}\| \leqslant \|(A-\lambda_0 I)\| \cdot \varepsilon^* + \|e\| \cdot \|C + (A^T+\lambda_0 I)X^{(i)}\|,$$

所以， $\|A^T(X^{(i)}-X)+(X^{(i)}-X)A\| \leqslant \|\bar{A}\| \cdot \varepsilon^* + \|e\| \cdot \|C - \underline{A}X^{(i)}\|$.
因而最终解的误差范围不仅与求逆的精度有关,同时也与迭代解的稳定性有关.

若采用解线性方程组的方法,基于 $X^{(i)}$ 去求解 $X^{(i+1)}$，设方程组求解误差为 e:

$$\|X^{(i+1)}(A-\lambda_0 I)-C-(A^T+\lambda_0 I)X^{(i)}\| \leqslant e.$$

由 $\|X^{(i+1)}-X^{(i)}\| \leqslant \varepsilon^*$ 可推出

$$\|X^{(i+1)}(A-\lambda_0 I)-X^{(i)}(A-\lambda_0 I)\| \leqslant \|(A-\lambda_0 I)\varepsilon^*\|,$$ 即

$$\|X^{(i)}(A-\lambda_0 I)+C+(A^T+\lambda_0 I)X^{(i)}\| \leqslant \|(A-\lambda_0 I)\cdot\varepsilon^*\| + \|e\|.$$

所以，$\|(X^{(i)}-X)A+A^T(X^{(i)}-X)\| \leqslant \|e\|+\|\bar{A}\|\cdot\varepsilon^*$. 可知 $\|(X^{(i)}-X)\| \leqslant (\|e\|+\|\bar{A}\|\cdot\varepsilon^*)/2\mu(A)$.
$\mu(A)$ 是 A 的 log-范数[2]，因而适当提高方程组解和迭代解的精度便可以求得 Lyapunov 方程的满意解.

以上分析说明直接采用方程组求解来代替乘逆的方法更容易控制 Lyapunov 方程直接迭代解的误差.

5　问题与讨论

本文给出了大型 Lyapunov 方程的特征子空间迭代法和相应的直接迭代方法，它们有相同的收敛速度. 其中,两个有待研究的问题是: (1)本文只是证明了所给算法的收敛性,它既能满足构造优势特征值的需要,又能满足有最大收敛速度的移轴量 λ_0. 但还没有定量的方法可以确定,这主要是在 A 有复变量特征值时,模最小的特征值其实部不一定最小. 因为所给算法具有稳定的收敛率，所以实际应用中,可以在计算时通过观察收敛率,随时调整移轴量来提高收敛速度. (2)在从子空间迭代法到直接迭代法的推导过程中,起加速作用和保持子空间不减秩的乘子 R 不再保留了，因而减少了计算量. 但理论推导和数值结果表明这并不影响直接迭代法的收敛性. 关于这方面的研究可以参见系统分解的 Riccati-like 迭代法和不变子空间法的关系的讨论[9].

参　考　文　献

[1] Hammarling S J. Numerical solution of the stable nonnegative definite Lyapunov equation. *IMA Journal of Numerical Analysis*, 1982, **2**(3): 303—323.

[2] A Scotledward Hodel, Kameschwar Poolla. Numerical solution of very large sparse Lyapunov equations through approximate power iteration. Proceedings of the 29th IEEE Conf. on Decision and Control, 1990, **1**: 291—296.

[3] Datta B N. Large-scale and parallel matrix computations in linear control: A tutorial. In Proceedings of the 31th IEEE Conf. on Decision and Control, 1992, **1**: 137—141.

[4] Victor Pan, John Reif. Fast and efficient parallel solution of sparse linear system. *SIAM J. Comput.*, 1993, **22**(6): 1227—1250.

[5] Bertskhas D P *et al*. Some aspects of parallel and distributed iteration——A survey. *Automatica*, 1991, **27**(1).

[6] 陈陆平. 非 Hermitian 矩阵不变子空间求解的直接迭代法,大连理工大学学报,1994(5): 502—506.
[7] J. Vonzur Gathen. Parallel algorithms for algebraic problems. *SIAM J. Comput.*, 1984, **13**(4): 802—824.
[8] 李晓梅等. 并行算法,国防工业出版社,1991.
[9] Mahmoud M S. Discrete systems with multiple time scales. Control and Dynamic Systems, 1988, 307—367.

PARALLEL ALGORITHM FOR THE SOLVING OF LARGE-SCALE LYAPUNOV EQUATION

CHEN LUPING XI YUGENG ZHANG ZHONGJUN
(*Dept, of Automatic Control, Shanghai Jiaotong Univ., Shanghai* 200030)

ABSTRACT

In this paper, a new method to solve the large-scale Lyapunov equation based on the subspace iteration concept used in finding the sub-eigenpairs of large-scale matrix is presented. A parallel method with high efficiency is also proposed. The convergency and the error bound of the algorithm are analysed.

Key words: Lyapunov equation, parallel computing, eigen-subspace, direct iteration.

陈陆平 1964 年生于辽宁, 1987 年毕业北京大学力学系, 1990 年和 1993 年于大连理工大学工程力学研究所分获硕士和博士学位,1993—1995年为上海交通大学自动化系博士后科研人员,现为该系副教授。研究领域为大系统理论和柔性结构并行控制。

席裕庚, 张钟俊 简介及照片见本刊第 19 卷第 5 期。