arXiv:2601.04429v1 [math.NA] 7 Jan 2026

迈向预条件类共轭梯度特征值求解器的真正高效性与簇鲁棒性

Ming Zhou 和 Klaus Neymeyr

Abstract. 特征值问题求解器（特征值求解器）的性能取决于预条件和特征值分布等多种因素。开发稳定且快速收敛的逐向量特征值求解器，是提高其分块实现整体效率的关键步骤。本文研究厄米特特征值问题的局部最优分块预条件共轭梯度法（LOBPCG），并受到近期针对其单向量版本 LOPCG 提出的两种替代方法的启发。这些特征值求解器的共同基础，是用于线性方程组的著名共轭梯度法。然而，共轭梯度搜索方向的最优性无法完美地转移到类共轭梯度特征值求解器中。特别是在计算聚集特征值时，LOPCG 及其替代方法会频繁出现延迟，导致阶梯状收敛行为，而现有估计无法解释这种现象。基于这一点，我们构造了一类具有簇鲁棒性的逐向量迭代方法：用渐近等价的二项递推替代 LOPCG，并通过选择较早的迭代作为增广项，及时校正搜索方向。新方法显著减少了所需步数和总计算时间。

1. 引言

用于求解厄米半正定矩阵线性方程组的共轭梯度法（CG），推动了高效且节省内存的迭代求解器的发展，使其能够处理更一般的非二次优化问题。在计算大型稀疏厄米矩阵或矩阵对的极端特征值时，已有多种所谓的类共轭梯度特征值求解器。其中典型的两种方法是：采用基于共轭梯度法的三项递推的局部最优块预条件共轭梯度法（LOBPCG） [11] ，以及在内迭代中使用共轭梯度法、且停止准则取决于瑞利商残差的雅可比–戴维森法（JDCG） [19]。尽管标题中没有体现，预条件也是 JDCG 的一项重要特征；数值测试已在多项研究中对这两种方法进行了比较，并由此产生了一些颇有争议的评论 [2, 29, 21]。越来越多的后续研究开始探讨它们的收敛理论和替代实现方式 [8, 17, 7, 31]。本文受到近期关于 LOBPCG 单向量版本 LOPCG 的研究成果 [27, 28] 的启发。该研究提出了两种收敛行为和内存需求相似的新方法，并给出了与文献 [11] 中的一个猜想相仿的收敛估计。由此引出一个问题：LOPCG 的收敛性分析和性能提升能在多大程度上借助这些结果？具体而言，我们希望弄清相关估计在大多数情况下是否有意义，以及在不显著扩展子空间的前提下，能否通过某些非块迭代大幅加速 LOPCG。理想情况下，若能采用自适应的向量迭代来加速，并结合消去，那么在计算大量（高度）聚集特征值时，其性能将优于其他特征值求解器。

2010 *Mathematics Subject Classification.* 主要分类：65F15、65N12、65N25。*Key words and phrases.* Hermitian 特征值问题、共轭梯度法、单向量特征值求解器、簇鲁棒性、二项递推、试探子空间增广。2026年1月7日。

LOPCG 方法在 [11] 中首次以一种算法的形式提出，用于求解实对称矩阵的广义特征值问题，并且很容易推广到厄米矩阵。这里我们关注引文中最常采用的问题设置，即：

$$Ax = \lambda Mx, \quad A, M \in \mathbb{C}^{n\times n} \text{ Hermitian}, \quad M \text{ positive definite}. \tag{1.1}$$

附录中针对偏微分算子、厄米定矩阵束 [14] 以及线性响应特征值问题 [4]介绍了一种更一般的表述。更早之前，文献 [9, 10]也讨论了 LOPCG 的基本思想。鉴于在 (1.1) 中，我们关注的是最小特征值 λ_1 及其对应的特征向量 x_1，可以先从理想情形出发进行启发式讨论：假设 λ_1 已知（如果 M 是单位矩阵，那么可以通过经典的 Lanczos 方法实现来近似求得 λ_1，但这种方法未必高效，而且无需确定近似特征向量 [6]）。因此，可以通过求解齐次线性系统来计算 x_1

$$A_{\lambda_1} x_1 = 0 \quad \text{for} \quad A_{\lambda_1} = A - \lambda_1 M \tag{1.2}$$

由于 A_{λ_1}具有半正定性，可以使用共轭梯度法求解。众所周知，共轭梯度法中针对迭代向量和搜索方向的二项递推形式，可以转换为三项递推，使新的迭代向量 $x^{(i+1)}$ 取决于其前两步迭代 $x^{(i-1)}$、 $x^{(i)}$ 以及残差 $A_{\lambda_1} x^{(i)}$。回到实际情形，即 λ_1 未知时，就必须改用近似特征值，并通过其他方式确定递推参数，例如按照 LOPCG 的做法，通过 Rayleigh-Ritz 过程隐式确定。值得注意的是，对于某些矩阵，这种实际方法的收敛速度几乎与启发式方法一样快，即使所用近似特征值在最初几步 [11]中并不接近 λ_1。从这个意义上说，在初始设置和内存需求相同的特征值求解器中，LOPCG 能够实现全局最优性。然而，其块版本 LOBPCG 在数值比较中并非总是明显胜出，尤其是在目标特征值数量较多且间隔可能较小时 [2, 29, 21]。有时，基于共轭梯度法二项递推的迭代性能略好；而在预条件子精度较低时，重启的广义 Davidson 方法或 Jacobi-Davidson 方法至少在这类情况下更具优势。当然，LOBPCG 偶尔表现欠佳在某种程度上取决于具体问题，并不影响其广泛应用和灵活性；参见应用 [4, 15, 14, 16, 1, 26]。近期，文献 [27, 28] 报告了针对单向量形式 LOPCG 的一些数值测试，指出了存在问题的情形，并促使我们以经济的方式加速 LOPCG，例如将试探子空间的维数限制为至多四维。

为此，我们需要依据 LOPCG 的现有收敛理论，按照特征值分布和预条件质量对测试问题进行分类。首个此类估计实际上是来自 [11] 的一个猜想，其依据是这样一种观察：向后扩展 LOPCG 的试探子空间，将若干个更早的迭代向量纳入其中，并不能显著加快收敛。在这种情况下，LOPCG 可视为一种近乎最优重启的广义 Davidson 迭代；特别是，当预条件极其精确时，可以通过 Invert-Lanczos 过程对其进行分析 [17]。由此得到的界表明，收敛速度主要受两个最小互异特征值之间的间隙 $\lambda_2 - \lambda_1$ 影响。我们注意到， [11] 中观察到的现象与其中测试问题的无害间隙有关。相反，如果 λ_1 和 λ_2 聚集，LOPCG 的特征值近似可能呈现阶梯状收敛行为（参见下文图 2.3），而 Invert-Lanczos 则保持稳健；如果目标特征值的簇大小超过块大小，LOBPCG 也会出现类似情况。此外，对于采用精度较低的预条件子的 LOPCG，分析时必须纳入某些质量参数。然而，这并不意味着可以直接修改 Invert-Lanczos 的估计。要完整推导 [11] 猜想所预示的收敛因子，仍然颇具挑战。现有估计大多通过比较分析推导而来，采用的简化迭代包括预条件最速下降

迭代（PSD），其中迭代向量 $x^{(i-1)}$ 会从 LOPCG 的试探子空间中移除。这些结果包括针对采用简单步长 [13] 或 Rayleigh–Ritz 步长 [18]的 PSD 所得的精确估计，其收敛因子仅取决于两个或三个特征值以及一个预条件质量参数。 [20] 中的类似估计还利用当前的近似特征值来构造动态收敛因子。尽管这些结果应用于 LOPCG 时并不精确，但仍清楚地表明，满足相关质量条件的优质预条件子能够确保出色的收敛速度。综上所述，构造 LOPCG 的挑战性问题，原则上可以通过选取具有聚集特征值且预条件精度较低的问题来实现。

为提升 LOPCG 在棘手情形下的性能，我们首先构造了一种直接改进方法，称为 LOPCGa：在试探子空间中加入一个辅助向量；若该向量与当前迭代的角度偏差超过阈值，则需要对其进行更新。这项技术可追溯至一种采用隐式消去的 PSD 实现 [5, 32] ，最近也应用于一种基于预条件和隐式凸性的特征值求解器（EPIC） [27], ，但尚未用于一种相关方法——带预条件的黎曼加速（RAP） [28]。RAP 和 LOPCG 的试探子空间维数为 3，EPIC 和 LOPCGa 的则为 4。根据相关研究及我们自己的数值测试，与所需步数和总计算时间相比，EPIC 均能加速 LOPCG；RAP 的竞争力较弱，但仍优于 PSD；而 LOPCGa 在大多数情况下比 EPIC 更高效。不过，EPIC 和 RAP 都是依赖参数的方法，在进一步发展中有望通过先进策略得到增强。对于 LOPCGa，我们随后通过检测残差范数收敛历史中的峰值，改进了辅助向量的简单更新准则，从而进一步提升性能。这促使我们以类似方式改进 [22] 中提出的具有二项递推的共轭梯度法特征值求解器；这类求解器在渐近意义上与 LOPCG 等价。

本文其余部分安排如下：第 2 节介绍基本设定，并回顾 LOPCG，说明其动机及其与内存需求相近的特征值求解器（包括改进方法 LOPCGa）相比的局限性。第 3 节讨论 LOPCG 及相关特征值求解器的最新估计结果。第 4 节探讨 LOPCGa 中辅助向量的适当更新策略，以及采用扩展二项递推的更高效方案。数值测试分布在上述各节中。

2. LOPCG 及可比方案

我们考虑具有特征值 $\lambda_1 < \lambda_2 \le \cdots \le \lambda_n$ 的广义特征值问题 (1.1)，并着重研究通过预条件共轭梯度法类迭代数值计算与 λ_1 相关联的特征向量 x_1 。为便于阅读，本文不讨论涉及重特征值的情形 $\lambda_1 = \cdots = \lambda_s < \lambda_{s+1}$ （通过投影到特征子空间进行改写十分直接）。用于逼近 x_1 的向量迭代记为 $x^{(i)}$ ，迭代索引为 i。相应的近似特征值通过计算瑞利商确定

$$\lambda : \mathbb{C}^n \backslash \{0\} \to \mathbb{R}, \quad \lambda(x) = \frac{x^* A x}{x^* M x} \tag{2.1}$$

其中，星号 * 表示复共轭转置。值 $\lambda(x^{(i)})$ 简记为 $\lambda^{(i)}$。一般而言，任何厄米正定矩阵 $T \in \mathbb{C}^{n\times n}$ 都可用作预条件子，并将其解释为

$$T \approx A_\sigma^{-1} \quad \text{with} \quad \kappa = \operatorname{cond}_2(T^{1/2} A_\sigma T^{1/2}) \quad \text{for} \quad A_\sigma = A - \sigma M, \quad \sigma < \lambda_1 \tag{2.2}$$

其中，cond_2 表示谱条件数，即对于 $T^{1/2}A_\sigma T^{1/2}$的极端特征值 α_1和 α_n， κ 与 α_n/α_1 相同。T的实际构造可以利用不完全

因式分解或关于 A_σ的线性方程组的近似解。所谓非预条件情形，是指 T 由 $n \times n$ 单位矩阵 I 给出。我们注意到，非预条件情形仅能使标准特征值问题（即$M = I$）的分析更加容易，而不能简化广义特征值问题的分析。

2.1. 基准方法

我们综述中的第一个方案是 Knyazev 在 [11, 第 3]节中提出的一种启发式基准方法，其中假设可以获得 λ_1 ，并采用标准预条件共轭梯度法求解齐次线性系统（1.2），例如采用算法 2.1 的形式。

算法 2.1： 关于线性系统（1.2）的启发式 PCG

```
生成 x^(0)；设置 r^(0) = −A_λ1 x^(0)； w = Tr^(0)； γ^(0) = w*r^(0)；
p^(0) = w； 对于 i = 0, 1, ..... 直到收敛执行
    w = A_λ1 p^(i)；
    δ = γ^(i)/(w*p^(i))；
    x^(i+1) = x^(i) + δp^(i)；
    r^(i+1) = r^(i) − δw；
    w = Tr^(i+1)； γ^(i+1) = w*r^(i+1)；
    p^(i+1) = w + (γ^(i+1)/γ^(i))p^(i)；
```

文献 [11], 中描述了一些收敛性质；然而，其中没有给出近似特征值的显式估计，因而无法直接与相关方法已有的估计进行比较。我们通过补充一些证明和说明来弥补这一不足。其中忽略了平凡（且在实际中罕见）的情形： $x^{(0)}$ 是与 λ_1 相关联的特征向量，或者 M与相应的特征子空间正交。

引理 2.1。 *考虑算法 2.1，其中 T取自 (2.2)，并定义 $B = T^{1/2}A_{\lambda_1}T^{1/2}$。设$V$ 是像空间 $\mathrm{im}(B)$的任意标准正交基矩阵，那么向量 $\tilde{x}^{(i)} = V^*T^{-1/2}x^{(i)}$, $i = 0, 1, \ldots$ 是求解线性系统 $\tilde{B}\tilde{x}_1 = 0$ （其中 $\tilde{B} = V^*BV$）的共轭梯度迭代。*

证明。 注意到 VV^* 是 im(B)的投影矩阵，因此有 $B = (VV^*)B(VV^*)$。于是， $x^{(i)}$ 的残差 $r^{(i)}$ 可转换为 $\tilde{x}^{(i)}$ 的残差 $\tilde{r}^{(i)}$ ，具体如下：

$$\begin{aligned} & r^{(i)} = -A_{\lambda_1}x^{(i)} \\ \Rightarrow\quad & T^{1/2}r^{(i)} = -T^{1/2}A_{\lambda_1}T^{1/2}T^{-1/2}x^{(i)} = -BT^{-1/2}x^{(i)} = -VV^*BVV^*T^{-1/2}x^{(i)} \\ \Rightarrow\quad & V^*T^{1/2}r^{(i)} = -V^*VV^*BVV^*T^{-1/2}x^{(i)} = -\tilde{B}\tilde{x}^{(i)} = \tilde{r}^{(i)}. \end{aligned}$$

接下来，我们检查关于搜索方向 $p^{(i)}$ 更新的项 $\gamma^{(i)}$ 和 $Tr^{(i)}$ 的转换，即：

$$\begin{aligned} \gamma^{(i)} = r^{(i)*}Tr^{(i)} &= (T^{1/2}r^{(i)})^*(T^{1/2}r^{(i)}) = (-VV^*BVV^*T^{-1/2}x^{(i)})^*(-VV^*BVV^*T^{-1/2}x^{(i)}) \\ &= (-V^*BVV^*T^{-1/2}x^{(i)})^*(-V^*BVV^*T^{-1/2}x^{(i)}) = \tilde{r}^{(i)*}\tilde{r}^{(i)}, \\ (V^*T^{-1/2})Tr^{(i)} &= V^*T^{1/2}r^{(i)} = \tilde{r}^{(i)}. \end{aligned}$$

˜因此，算法 2.1 对应于一种用于求解 $B\tilde{x}_1 = 0$ 的（非预条件）共轭梯度方案。 □

˜引理 2.1 使我们能够应用共轭梯度法众所周知的收敛估计，因为 B 显然是正定的。根据 [24, 定理 6.29], ，有

$$\frac{\|\tilde{x}^{(i)}\|_{\tilde{B}}}{\|\tilde{x}^{(0)}\|_{\tilde{B}}} = \frac{\|0 - \tilde{x}^{(i)}\|_{\tilde{B}}}{\|0 - \tilde{x}^{(0)}\|_{\tilde{B}}} \leq \left(\mathcal{C}_i(\tilde{\varphi})\right)^{-1} \quad \text{for} \quad \tilde{\varphi} = \frac{\tilde{\beta}_m + \tilde{\beta}_1}{\tilde{\beta}_m - \tilde{\beta}_1} \tag{2.3}$$

˜其中， $\mathcal{C}_i$ 是次数为 i 的第一类切比雪夫多项式， β_1˜ ˜是最小特征值， β_m 是 B的最大特征值。接下来，我们要将 (2.3) 扩展为算法 2.1 的显式估计。

定理 2.2. 由算法 *2.1* 得到的近似特征值 $\lambda^{(i)} = \lambda(x^{(i)})$、 $i = 0$、1、... 满足

$$\lambda^{(i)} - \lambda_1 \le \big(\mathcal{C}_i(\varphi)\big)^{-2} \frac{\|x^{(0)}\|_M^2}{\|x^{(i)}\|_M^2} (\lambda^{(0)} - \lambda_1) \quad for \quad \varphi = \frac{\eta+1}{\eta-1}, \quad \eta = \kappa \frac{(\lambda_n - \lambda_1)(\lambda_2 - \sigma)}{(\lambda_2 - \lambda_1)(\lambda_n - \sigma)} \tag{2.4}$$

其中， κ是(2.2)中定义的条件数。

证明. 通过检查 A_{λ_1} 和 T 的性质可知，引理 2.1 中给出的矩阵 $B = T^{1/2} A_{\lambda_1} T^{1/2}$ 是半正定的，且秩为 $n-1$。记其特征值为 $\beta_1 \le \beta_2 \le \cdots \le \beta_n$、˜，则对于 $j = 2, \ldots, n$，有 $\beta_1 = 0$、 $\beta_j > 0$。此外，由于 $B = V^* B V$ 是 B˜ ˜ ˜在 im(B) 上的限制， B 的极端特征值 β_1 和 β_m 分别与 β_2 和 β_n 相同。

另一方面， $B = T^{1/2} A_{\lambda_1} T^{1/2}$的特征值也是矩阵对 ($A_{\lambda_1}$， T^{-1}) 的特征值，因此与瑞利商相关

$$\beta(y) = \frac{y^* A_{\lambda_1} y}{y^* T^{-1} y}$$

它可以改写为另外两个瑞利商的乘积:

$$\beta(y) = \psi(y)\, \alpha(y) \quad \text{for} \quad \psi(y) = \frac{y^* A_{\lambda_1} y}{y^* A_\sigma y} \quad \text{and} \quad \alpha(y) = \frac{y^* A_\sigma y}{y^* T^{-1} y}.$$

相应地，我们将矩阵对 (A_{λ_1}， A_σ) 和 (A_σ， T^{-1}) 的特征值分别记为$\psi_1 \le \psi_2 \le \cdots \le \psi_n$和 $\alpha_1 \le \alpha_2 \le \cdots \le \alpha_n$（它们均为非负数）。于是， $\alpha_1\psi(y) \le \beta(y) \le \alpha_n\psi(y)$ 成立，而 Courant-Fischer 原理可推出

$$\begin{aligned}
\beta_2 &= \min_{\dim \mathcal{Y}=2} \max_{y \in \mathcal{Y}\setminus\{0\}} \beta(y) \ge \alpha_1 \min_{\dim \mathcal{Y}=2} \max_{y \in \mathcal{Y}\setminus\{0\}} \psi(y) = \alpha_1 \psi_2, \\
\beta_n &= \max_{y \in \mathbb{C}^n \setminus\{0\}} \beta(y) \le \alpha_n \max_{y \in \mathbb{C}^n \setminus\{0\}} \psi(y) = \alpha_n \psi_n.
\end{aligned}$$

由于 α_1 和 α_n 也是 $T^{1/2} A_\sigma T^{1/2}$ 的极端特征值，根据 (2.2)，有 $\kappa = \alpha_n/\alpha_1$。($A_{\lambda_1}$， A_σ) 的特征值与 $A_\sigma^{-1} A_{\lambda_1}$ 的特征值相同，并且可以用 (A， M) 的特征值表示。利用对角矩阵 $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ 以及以相应的 M-正交特征向量为列组成的矩阵 X，可得因式分解

$$A_\sigma^{-1} A_{\lambda_1} = \big(X(\Lambda - \sigma I)^{-1} X^*\big)\big(MX(\Lambda - \lambda_1 I) X^* M\big) = X(\Lambda - \sigma I)^{-1}(\Lambda - \lambda_1 I) X^{-1}$$

因此，$(A_{\lambda_1}, A_\sigma)$ 的特征值由 $(\Lambda - \sigma I)^{-1}(\Lambda - \lambda_1 I)$ 的对角线给出，即，对于 $j = 1, \ldots, n$， $\psi_j = (\lambda_j - \lambda_1)/(\lambda_j - \sigma)$。综上所述，

$$\frac{\tilde\beta_m}{\tilde\beta_1} = \frac{\beta_n}{\beta_2} \le \frac{\alpha_n \psi_n}{\alpha_1 \psi_2} = \kappa \frac{(\lambda_n - \lambda_1)(\lambda_2 - \sigma)}{(\lambda_2 - \lambda_1)(\lambda_n - \sigma)} = \eta, \tag{2.5}$$

并且式（2.3）中的切比雪夫项可根据以下内容进行界定

$$\tilde\varphi = \frac{\tilde\beta_m + \tilde\beta_1}{\tilde\beta_m - \tilde\beta_1} = \frac{\tilde\beta_m/\tilde\beta_1 + 1}{\tilde\beta_m/\tilde\beta_1 - 1} \ge \frac{\eta+1}{\eta-1} = \varphi > 1 \quad \Rightarrow \quad \big(\mathcal{C}_i(\tilde\varphi)\big)^{-1} \le \big(\mathcal{C}_i(\varphi)\big)^{-1}. \tag{2.6}$$

接下来，范数变换（再次使用 $B = (VV^*)B(VV^*)$）

$$\begin{aligned}
\|\tilde x^{(i)}\|_{\tilde B}^2 &= \tilde x^{(i)*} \tilde B \tilde x^{(i)} = (V^* T^{-1/2} x^{(i)})^* (V^* B V)(V^* T^{-1/2} x^{(i)}) \\
&= x^{(i)*} T^{-1/2} B T^{-1/2} x^{(i)} = x^{(i)*} T^{-1/2} (T^{1/2} A_{\lambda_1} T^{1/2}) T^{-1/2} x^{(i)} \\
&= x^{(i)*} A_{\lambda_1} x^{(i)} = x^{(i)*} A x^{(i)} - x^{(i)*} \lambda_1 M x^{(i)} = (\lambda^{(i)} - \lambda_1) \|x^{(i)}\|_M^2
\end{aligned}$$

以及针对 $x^{(0)}$ 的对应结果，结合 (2.6) 与 (2.3) 即可完成证明。 □

$x^{(i)}$ 与目标特征子空间之间的一些几何关系有助于推广定理 2.2。

引理 2.3. 令 $\mathcal{X}_1$ 为与 λ_1 相关联的特征子空间。那么，算法 *2.1* 生成的迭代向量 $x^{(i)}$、 $i=0$、1、.....满足

$$x^{(i)} - x^{(0)} \perp_{T^{-1}} \mathcal{X}_1 \quad \textit{and} \quad \|x^{(i)}\|_M^2 \geq \mu_1 \|x^{(0)}\|_{T^{-1}}^2 \big(\cos\angle_{T^{-1}}(x^{(0)},\mathcal{X}_1)\big)^2 \tag{2.7}$$

其中， μ_1 是矩阵对 (M,T^{-1}) 的最小特征值，而 $\angle_{T^{-1}}$ 表示相对于由 T^{-1} 诱导的内积所定义的角度。

证明。 共轭梯度性质

$$\begin{aligned} x^{(i)} &\in x^{(0)} + \operatorname{span}\{Tr^{(0)}, (TA_{\lambda_1})Tr^{(0)}, \ldots, (TA_{\lambda_1})^{i-1}Tr^{(0)}\} \\ &= x^{(0)} + (TA_{\lambda_1})\operatorname{span}\{x^{(0)}, (TA_{\lambda_1})x^{(0)}, \ldots, (TA_{\lambda_1})^{i-1}x^{(0)}\} \end{aligned}$$

保证了某个 w 对应的 $x^{(i)} - x^{(0)} = TA_{\lambda_1}w$。因此，对任意 $x_1 \in \mathcal{X}_1\backslash\{0\}$，均有

$$x_1^* T^{-1}(x^{(i)} - x^{(0)}) = x_1^* T^{-1} TA_{\lambda_1} w = (A_{\lambda_1}x_1)^* w = 0$$

由此得到式 (2.7) 中的正交性。进一步，记 x_1 为 $x^{(0)}$ 在 $\mathcal{X}_1$ 上的 T_{-1}-投影，则该正交性意味着

$$\begin{aligned} \cos\angle_{T^{-1}}(x^{(0)},\mathcal{X}_1) = \cos\angle_{T^{-1}}(x^{(0)},x_1) &= \frac{x_1^* T^{-1} x^{(0)}}{\|x_1\|_{T^{-1}}\|x^{(0)}\|_{T^{-1}}} \\ &= \frac{x_1^* T^{-1} x^{(i)}}{\|x_1\|_{T^{-1}}\|x^{(0)}\|_{T^{-1}}} \leq \frac{\|x_1\|_{T^{-1}}\|x^{(i)}\|_{T^{-1}}}{\|x_1\|_{T^{-1}}\|x^{(0)}\|_{T^{-1}}} = \frac{\|x^{(i)}\|_{T^{-1}}}{\|x^{(0)}\|_{T^{-1}}}. \end{aligned}$$

此外，注意到 $\|x^{(i)}\|_M^2/\|x^{(i)}\|_{T^{-1}}^2$ 是矩阵对 $(M,\ T^{-1})$ 在 $x^{(i)}$ 处的瑞利商，可得下界 μ_1。于是

$$\|x^{(i)}\|_M^2 \geq \mu_1\|x^{(i)}\|_{T^{-1}}^2 \geq \mu_1\|x^{(0)}\|_{T^{-1}}^2\big(\cos\angle_{T^{-1}}(x^{(0)},\mathcal{X}_1)\big)^2. \qquad \square$$

(2.7) 中的 T^{-1}-正交性表明， $x^{(i)}$ 与 $\mathcal{X}_1$ 之间的 T^{-1}-夹角可以衡量算法 2.1 的收敛性，而 (2.7) 中的不等式可以直接与 (2.4) 相加，从而消去项 $\|x^{(i)}\|_M^2$。

注 2.4。 在特殊情形 $T=M=I$ 下，将 (2.7) 与 (2.4) 结合可得到 Lanczos 方法的一个估计。其中 $\mu_1=1$，且 (2.7) 给出

$$\|x^{(i)}\|_2^2 \geq \|x^{(0)}\|_2^2\big(\cos\angle_2(x^{(0)},\mathcal{X}_1)\big)^2 \tag{2.8}$$

（其中下标 2 表示欧几里得项）。此外，沿着定理 2.2 的证明，取 $T=I$ 可得到 $\eta=(\lambda_n-\lambda_1)/(\lambda_2-\lambda_1)$。然后，利用 (2.8)，(2.4) 变为

$$\lambda^{(i)} - \lambda_1 \leq \big(\mathcal{C}_i(\varphi)\big)^{-2}\big(\cos\angle_2(x^{(0)},\mathcal{X}_1)\big)^{-2}(\lambda^{(0)}-\lambda_1) \quad \text{for} \quad \varphi = \frac{\eta+1}{\eta-1} = 1 + 2\,\frac{\lambda_2-\lambda_1}{\lambda_n-\lambda_2} \tag{2.9}$$

与其传统对应形式相比， [23, 定理 2]（通过改变特征值排列），差异仅体现在依赖初始猜测的项上。随后，由于对于 $T=M=I$ 算法 2.1，生成的 Krylov 子空间序列与关于 A的序列相同，且 $\lambda^{(i)}$ 不小于其在 Lanczos 方法中的对应值，即与当前 Krylov 子空间相关的最小 Ritz 值，因此 (2.9) 可直接应用于 Lanczos 方法。此外，由于 $\lambda^{(0)}-\lambda_1 \leq (\lambda_n-\lambda_1)\big(\sin\angle_2(x^{(0)},\ \mathcal{X}_1)\big)^2$，(2.9) 给出的界比传统界更紧。类似地，在特殊情形 $T=A_\sigma^{-1}$下，算法 2.1 生成关于 $A_\sigma^{-1}M$ 的 Krylov 子空间，因此 (2.4) 给出了对 Invert-Lanczos 方法的估计，与 [17]给出的估计相当。然而，情形 $T=I$, $M\neq I$ 与 Lanczos 方法和 Invert-Lanczos 方法无关。

注 2.5。在 $\lambda^{(i)}$ 接近 λ_1的最终阶段，算法 2.1 的表现类似于广义 Davidson 方法，因此可以进行比较分析。然而，将 (2.7) 与 (2.4) 结合，可能会对精度较低的预条件子造成显著高估，而估计的紧致程度取决于 (2.2) 中 $\sigma < \lambda_1$ 的选择。尽管 A_{λ_1} 的伪逆可带来一步收敛，但从 λ_1 附近选择 σ 来构造实用预条件子并不合理。例如，鉴于已知 A为正定，通常取 $\sigma = 0$，使得 $A_\sigma = A$且 $\kappa = \mathsf{cond}_2(T^{1/2}AT^{1/2})$；参见 [11, 27]。此外，$\varphi = (\eta+1)/(\eta-1)$ 的多步收敛因子 $\big(\mathcal{C}_i(\varphi)\big)^{-2}$，可基于切比雪夫多项式的等价改写，解释为具有如下形式的渐近平均单步收敛因子：$\big((\sqrt{\eta}-1)/(\sqrt{\eta}+1)\big)^2$。需要注意的是，这种解释对于最初几步并不适用，因为第一步共轭梯度迭代仍是一次简单的梯度迭代，因此其中的收敛因子应接近 $\big((\eta-1)/(\eta+1)\big)^2$。我们将在第 3 节继续讨论这一问题。

2.2. 实用特征值求解器

算法 2.1 中唯一妨碍实际实现的项是 A_{λ_1}，因为特征值 λ_1 实际上正是需要计算的特征值。一种简单的解决方法是用一个容易获得的近似值替代 λ_1，例如瑞利商值 $\lambda^{(i)} = \lambda(x^{(i)})$。当然，仅进行这一替换会使收敛性变差，因为需要重新定义残差和步长。考虑到 λ_1 是瑞利商 (2.1) 的最小值，可以使用缩放梯度作为残差。此外，通过计算一系列更新后子空间的 Ritz 值，可以实现对 (2.1) 的全局最小化。

定义 2.6. 我们记

$$x' \xleftarrow{\text{RRw}} x + \mathcal{U} \;:\; x' \in x + \mathcal{U} \text{ such that } \lambda(x') = \min_{w \in \mathrm{span}\{x, \mathcal{U}\}\setminus\{0\}} \lambda(w) \tag{2.10}$$

以首个基向量为权重输出、旨在求得最小 Ritz 值的 Rayleigh-Ritz 过程。

利用 (2.10)，可以将算法 2.1 中第 3 行至第 5 行修改为

$$x^{(i+1)} \xleftarrow{\text{RRw}} x^{(i)} + \mathrm{span}\{p^{(i)}\}; \quad r^{(i+1)} = Ax^{(i+1)} - \lambda^{(i+1)}Mx^{(i+1)};$$

以保持向 λ_1 的收敛。此外，搜索方向$p^{(i)}$ 的其他一些公式 [21]可以带来更好的性能。然而，这种修改（或块形式）不像 LO(B)PCG 方法那样普及。我们将在第 4 节讨论其可能的改进。

LOPCG 的一个特征是，在 Rayleigh-Ritz 过程中隐式地确定所有步长。在算法 2.1 中，将第 6、7 行与下一步的第 4 行结合起来，$x^{(i+2)}$显然是 $x^{(i+1)}$、$Tr^{(i+1)}$ 和 $p^{(i)}$的线性组合。这表明存在一个三维试探子空间，并导出了算法 2.2 所示的 LOPCG 基本版本。其中，$p^{(i)}$ 的迭代索引发生了位移，算法 2.1 中的步长被用于确定 $x^{(i+1)}$ 这一 Ritz 向量的某些副产物所取代。算法 2.2 中的试探子空间通常比使用 $p^{(i)}$而非 $x^{(i-1)}$ 的理论等价形式更加稳定。除了 [11]，中提到的这一众所周知的事实之外，人们似乎还不太了解：加权形式$x^{(i+1)} = x^{(i)} + \cdots$（与启发式预条件共轭梯度法一致）和 $x^{(i+1)} = Tr^{(i)} + \cdots$（由 [11]提出）在不进行任何正交化的情况下，可能表现出明显不同的性能。实际上，实用代码 [12, lobpcg.m] 采用了部分正交化，其中 $Tr^{(i)}$ 相对于 $x^{(i)}$进行正交化。此外，如果底层格拉姆矩阵条件不佳，$p^{(i)}$ 将从试探空间中移除。另一项稳定化建议 [8, 7] 是对三个基向量进行完全的 M-正交归一化，但这一策略会增加每步的计算时间，并且

算法 2.2： 用于计算 (λ_1， x_1)（见式 (1.1)）的 LOPCG

1 生成 $x^{(0)}$；设置 $p^{(0)}=0$;
2 **对于** $i=0$，1，.....直到收敛 **执行**
3 $\lambda^{(i)}=\lambda(x^{(i)})$; $r^{(i)}=Ax^{(i)}-\lambda^{(i)}Mx^{(i)}$;
4 $x^{(i+1)} \xleftarrow{\text{RRw}} x^{(i)}+\text{span}\{Tr^{(i)},p^{(i)}\}$，如式（2.10）所示;
5 $p^{(i+1)}=x^{(i+1)}-x^{(i)}$;

根据我们最近的测试，有时甚至会增加所需的步数；参见第2.3节。来自 [21] 的双项共轭梯度格式似乎不会受到这种不稳定性的影响。

LOPCG 的性能已在 [11, 第 7] 节中展示，其中使用了对角 A 和 $M=I$。预条件子 T 由随机对角矩阵与正交矩阵相乘构造。最小的两个特征值相距较远： $\lambda_1=1$ 和 $\lambda_2=2$。结论是，对于 $\sigma=0$，无论是具有 $\kappa=4$ 的理想 T，还是具有 $\kappa=1000$ 的精度较低的 T，在设置（2.2）下，LOPCG 的表现都与启发式预条件共轭梯度法相近。值得注意的是，尽管初始的近似特征值 $\lambda^{(i)}$ 通常远离 λ_1，但仍可在几步之内到达目标区间（λ_1， λ_2）。这也解释了为什么现有收敛理论中的一些结果会直接假设某些 $\theta\in$ 满足 $\lambda^{(0)}<\theta$ （λ_1， λ_2）。另一个事实是，LOPCG 显著加速了预条件最速下降迭代（PSD）；后者的试探子空间仅为 span$\{x^{(i)},Tr^{(i)}\}$。这与求解线性方程组时相应方法之间的比较类似。

在不少性能研究中，研究者会将 LOPCG 及其块扩展 LOBPCG 与其他实用特征值求解器进行数值比较，但各方法的内存需求并不总是相近。例如，在 [29, 31]，中，除了考虑 LOPCG 和双块 LOBPCG，还考虑了 Davidson 类方法；即使只计算一个极端特征对，这些方法使用的试探子空间也多达 18 维。值得注意的是，在 LOPCG 的收敛速度远慢于 Davidson 类方法的情形下，其收敛行为也与启发式预条件共轭梯度法有显著差异。我们认为，对 LOPCG 进行较有意义的比较分析，应聚焦于使用 2 至 4 维试探子空间的特征值求解器，例如上文 [21] 提到的双项共轭梯度格式，以及近期开发的方法 EPIC [27]和 RAP [28]。

我们简要介绍 EPIC 和 RAP 的一些特点，因为它们与启发式预条件共轭梯度法的关联较小，而是通过最小化与所考虑的特征值问题相关的瑞利商以外的某些函数推导而来。EPIC 中的方法（其中 T 记作 T^{-1}）利用辅助向量 q，并考虑满足 $q^*Mx>0$的 M-归一化向量 x。随后，针对这类 x的表示定义一个中间函数，并以此为基础，运用凸优化理论进一步构造方法。撇开细节上的差异，预条件子的设定确实与 (2.2) 类似，但针对的是中间函数的海森矩阵。其条件数表示为 L/μ，并在 [27, 推论 2.1] 中证明，对于 $\sigma=0$，它与参数 η 相近，二者的距离取决于 $\lambda(q)-\lambda_1$。有趣的是，参数 μ和 L也涉及 EPIC 的算法，因此在实际实现之前需要对其进行估计。然而， [27] 中的选择$\mu=L=6$ 相当依赖经验，因为这对应于一个能够实现一步收敛的完美预条件子，但其中数值测试所采用的预条件子显然不够精确。尽管如此，根据我们额外进行的尝试，使用精确的 η 作为 L/μ 并未比上述选择带来更快的收敛；参见第2.3节。另一方面，RAP 中的中间函数（其中 T 记作 B^{-1}）与矩阵对 $(T^{1/2}AT^{1/2},\ T^{1/2}MT^{1/2})$ 相关的瑞利商有关，且不依赖辅助向量。预条件的条件数也采用 L/μ的形式

并且渐近地与 (2.4) 中的 η 一致；参见 [28, 推论 16]。 [28] 中的参数选择 $\mu = 8$、$L = 50$ 似乎是根据一个具体数值示例确定的，但与 [28, 命题24、定理 26]中的条件 $L \geq 9\mu$ 相矛盾。

与 LOPCG 直接比较，EPIC 和 RAP 中的方法提出了维数相近（分别为 4 和 3）的新型试探子空间。根据相关数值测试和我们自己的数值测试，EPIC 的表现可能优于 LOPCG；而 RAP 的收敛速度通常慢于 LOPCG，但仍能加速预条件最速下降法（PSD）。这启发我们构造一种高效的 LOPCG 改进方法，其试探子空间与 EPIC 一样为四维。然而，正如 [11], 所报告的，将先前的迭代向量直接添加到 LOPCG 的试探子空间中，并不一定能显著减少所需步数。我们将这类方法中最简单的版本记为 LOPCGx，即以 span$\{x^{(i)}, Tr^{(i)}, p^{(i)}, x^{(i-2)}\}$ 作为试探子空间，并在第2.3节的数值比较中纳入 LOPCGx。

为了高效地改进LOPCG，我们注意到，EPIC中的辅助向量 q 由初始猜测 $x^{(0)}$ 初始化，并且一旦$|q^*Mx^{(i)}|$ 低于经验值0。5，EPIC就会以 $q = x^{(i)}$ 重新启动。之所以必须重新启动，是因为如理论构造所述，第一个 q 通常不是良好的近似特征向量。尽管给出的判据没有附带解释，但显然，项 $|q^*Mx^{(i)}|$ 是 $x^{(i)}$与span$\{q\}$之间的 M角余弦，因为EPIC的迭代向量始终经过 M归一化。因此，关键信息是，合适的 q 应与当前迭代向量保持较小的角度偏差。 q 在实践中的有效性直接促使我们对LOPCG进行首次改进，并将其称为“增广LOPCG”（LOPCGa）：将辅助向量 a 加入试探子空间，并按照判据 $|\cos \angle_M(a,\ x^{(i)})| < \tau$进行更新，阈值为 $\tau = 0.7$；见算法 2.3。不过，我们更倾向于将 a 解释为较早之前的迭代向量，因为与 q不同， a 不会用于生成试探子空间的替代基向量。换言之，LOPCGa在结构上仍与LOPCG和启发式预条件共轭梯度法相似。

算法 2.3： 用于计算 (1.1) 中 (λ_1, x_1) 的增广LOPCG（LOPCGa）

生成 $x^{(0)}$；设置 $a = x^{(0)}$;
对于 $i = 0,\ 1,$ ..... 直到收敛 **执行**
  $\lambda^{(i)} = \lambda(x^{(i)})$; $r^{(i)} = Ax^{(i)} - \lambda^{(i)}Mx^{(i)}$; $w = Tr^{(i)}$;
  若 $i > 0$ **则**
    若 |余弦 $\angle_M(a, x^{(i)})| < \tau$**则**
      $a = x^{(i)}$; $\mathcal{U}$ =张成空间$\{w, p^{(i)}\}$;
    else
      $\mathcal{U}$ = 张成空间$\{w, p^{(i)}, a\}$;
    $W \xleftarrow{M\text{-正交化}}$ 跨度$\{x^{(i)}, \mathcal{U}\}$;
    如果 W^*MW条件不佳**那么**
      $\mathcal{U}$ =跨度$\{w\}$ or $\mathcal{U}$ =跨度$\{w, p^{(i)}\}$;
  else
    $\mathcal{U}$ =跨度$\{w\}$;
  $x^{(i+1)} \xleftarrow{\text{RRw}} x^{(i)} + \mathcal{U}$ 如式（2.10）；$p^{(i+1)} = x^{(i+1)} - x^{(i)}$;

LOPCGa 的另一项重要特性受到 [12] 关于通过子空间约简实现稳定性的启发。如算法 2.3 所述，在试探子空间中进行 M-正交化，以判断当前步骤是否需要舍弃辅助向量 a 和搜索方向 $p^{(i)}$。更具体地说，对于第 10 行和第 11 行，我们将相应的（对角）格拉姆矩阵 W^*MW的对角元素记为 $\delta_1, \ldots, \delta_k (k \in \{3, 4\})$，并采用阈值 $\gamma =$ 1e26。然后

TABLE 2.1. 预条件子的条件数。

示例	T_1	T_2	T_3	T_4	T_5	T_6
(2.11, a)	4.61e6	1.15e5	5.98e4	4.75e3	3.19e2	6.86e0
(2.11, b)	1.59e0	1.53e0	1.07e0	1.01e0	1.00e0	1.00e0
(2.11, c)	9.31e3	4.75e2	5.78e1	7.04e0	1.45e0	1.02e0

在 $\delta_1 > \gamma\delta_3$ 的情况下，试探子空间约简为 span$\{x^{(i)}, w\}$；若满足 $k = 4$ 和 $\gamma\delta_3 \geq \delta_1 > \gamma\delta_4$，则约简为 span$\{x^{(i)}, w, p^{(i)}\}$。这种无归一化技术使 LOPCGa 比其采用普通正交归一化的初始版本更加稳定，表现为在最终阶段残差范数的下降更加平滑。在尝试构造进一步的改进版本之前，我们先通过几个数值算例报告 LOPCGa 的性能。

2.3. 数值比较

我们选取三个示例，以说明当 LOPCG 仍有改进空间时，LOPCGa 能显著加速 LOPCG。

$$
\begin{aligned}
&\text{(a)} \quad A = \texttt{boneS01}, \quad M = I, \quad n = 127224, \quad \lambda_1 \approx \texttt{2.847268e-3}, \quad \lambda_2 \approx \texttt{6.591631e-2} \\
&\text{(b)} \quad A = \texttt{finan512}, \quad M = I, \quad n = 74752, \quad \lambda_1 \approx \texttt{9.474684e-1}, \quad \lambda_2 \approx \texttt{9.502419e-1} \\
&\text{(c)} \quad (A, M) \text{ by finite elements}, \quad n = 178820, \quad \lambda_1 \approx \texttt{1.973967e1}, \quad \lambda_2 \approx \texttt{1.973975e1}
\end{aligned} \tag{2.11}
$$

示例 (a) 和 (b) 列于 (2.11)，选自 SuiteSparse 矩阵集合（佛罗里达大学），也用于 [27] 中的数值测试；在这些测试中，LOPCG 有时明显比 EPIC 收敛得慢。示例 (c) 来源于矩形区域 $[0, 2]\times[0, 1]$ 上拉普拉斯特征值问题的有限元离散化，其中带有狭缝 $\{1\}\times[0.1, 0.9]$，并采用齐次狄利克雷边界条件。该问题的两个最小特征值紧密聚集。

对于每个示例，我们使用 Matlab 中的 ichol 函数生成六个预条件子，即 T_1 采用 nofill，以及 T_i 采用 ict，设置 droptol=1e-i（i 为,...,6）；但 (2.11, a) 中的 T_2 采用 droptol=2e-3，以确保可行性。采用 $\sigma = 0$ 定义的相关条件数列于表 2.1。此外，对于这些预条件子，我们还测试了启发式预条件共轭梯度法（算法 2.1，其中通过 eigs 使用λ_1），并以 M归一化的全一向量作为初始猜测。图 2.1 展示了 Ritz 值误差 $\lambda^{(i)} - \lambda_1$和（相对）残差范数 $\|r^{(i)}\|_2/\|x^{(i)}\|_M$ 的收敛历史。除收敛最终阶段之前的延迟无法由任何现有估计反映外，性能与预条件数密切相关，这与第 2.1 节的分析一致。特别是，对于精度较低的预条件子，残差范数下降可能出现强烈振荡，因为输入的 λ_1值仍与精确特征值不同。为解决这一问题，我们在最终阶段用当前的 $\lambda^{(i)}$ 更新该值，从而减弱振荡，如左下子图中的 T_3 和 T_4 所示。这也表明，当检测到异常振荡时，可以立即切换到启发式预条件共轭梯度法，以改进最终阶段不稳定的实用特征值求解器。关于 Ritz 值误差，我们注意到 $\lambda^{(i)}$ 可能低于输入的 λ_1 值，因此无法通过对数刻度显示 $\lambda^{(i)} - \lambda_1$。在这种情况下，我们用虚线绘制 $|\lambda^{(i)} - \lambda_1|$；参见左上和右上子图。另一个事实是，与采用 Rayleigh-Ritz 过程的迭代相比，启发式预条件共轭梯度法中的 $\lambda^{(i)}$ 不一定单调收敛。在上述准备工作的基础上，我们选取六种组合，对实用特征值求解器进行数值比较；参见表 2.2。

注释 2.7。如 [27, 图 2(b)], 所示，示例（2.11，a）是 LOPCG 的一个棘手案例：在一次测试中，LOPCG 所需步数超过 1000 步，而 EPIC 约 500 步即可收敛。这里我们考察类似案例，并在比较中加入更多方案。对于

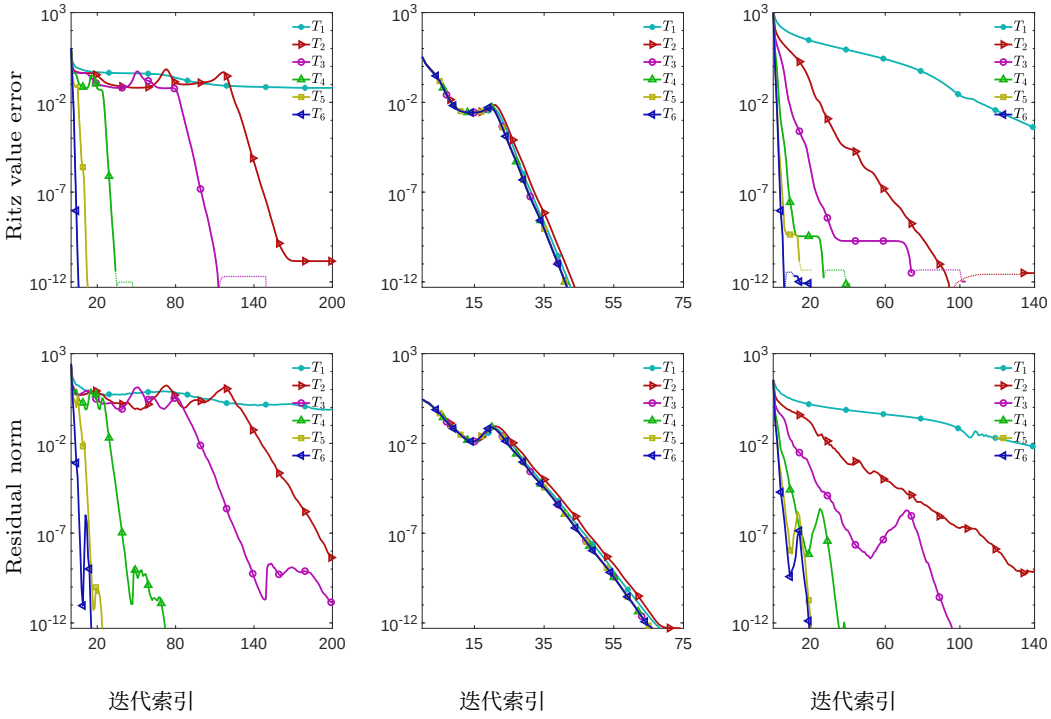


FIGURE 2.1. 通过观察启发式预条件共轭梯度法（算法 2.1）对示例 (2.11) 及表 2.1 中预条件子的收敛历史，选取典型案例。

TABLE 2.2. 数值比较的参数。

示例，T_i	droptol	κ	η	$\beta_{\min}$	$\beta_{\max}$	seed	图
(2.11, a), T_4	1e−4	4.75e3	4.96e3	2.31e-4	1.15e0	20	2.2 左
(2.11, a), T_3	1e−3	5.98e4	6.25e4	2.53e-5	1.58e0	99	2.2 中间
(2.11, a), T_2	2e−3	1.15e5	1.21e5	1.31e-5	1.58e0	258	2.2 右
(2.11, b), T_3	1e−3	1.07e0	3.53e2	2.82e-3	9.94e-1	246	2.3 左
(2.11, c), T_3	1e−3	5.78e1	1.40e7	7.98e-8	1.12e0	27	2.3 中间
(2.11, c), T_2	1e−2	4.75e2	1.15e8	1.12e-8	1.29e0	171	2.3 右

所用的预条件子，表 2.2 列出了相应的 ichol droptol 值以及根据（2.2）、（2.4）为 $\sigma = 0$定义的质量参数 κ、 η 。此外，我们还列出了 η 的推导（2.5）所得的 $\beta_{\min} = \alpha_1\psi_2$和 $\beta_{\max} = \alpha_n\psi_n$ ，因为 $2\beta_{\min}$ 和 $2\beta_{\max}$ 与 EPIC 和 RAP 中参数 μ 和 L 的理论设置接近。

我们首先按照算法 2.2 的形式实现 LOPCG，针对 $\{0, \ldots, 300\}$ 中采用不同随机数种子的各种初始猜测进行测试。通过这种方式，我们找到了一个对 LOPCG 较为困难的案例：它所需的步数至少是 EPIC 的两倍。为完整起见，我们还测试了 LOPCG 的另外两个版本。按照 [8] 进行完整的 M 正交化的版本与算法 2.2 相比没有明显差异，而代码 [12, lobpcg.m] 则不会遇到这一棘手案例；收敛历史见图 2.2 左列。其中，LOPCG、EPIC 和 RAP 的实线分别对应于 lobpcg.m、EPIC(6,6) 和 RAP(8,50)，参数 μ 和 L 采用 [27, 28], 中的经验值；虚线则分别表示算法 2.2、EPIC($2\beta_{\min}$, $2\beta_{\max}$) 和 RAP($2\beta_{\min}$, $2\beta_{\max}$)。有趣的是，尽管 EPIC 和 RAP 的经验设置似乎与理论构造不一致，但它们带来了更好的性能。对于 lobpcg.m，通过查看其具体细节，我们注意到加速

是在第 40 步切换到预条件最速下降法（PSD）后实现的，此时底层格拉姆矩阵的条件数超过了某个阈值。在我们针对该步附近残差范数异常振荡而进行的额外测试中，通过观察残差的降低情况，也实现了类似的有效切换。此外，对算法 2.2 的简单扩展 LOPCGx（将 $x^{(i-2)}$ 加入试探子空间）只能带来轻微的加速。相比之下，自适应扩展 LOPCGa（算法 2.3）仅比 EPIC 晚几步开始加速，并在最终阶段获得了好得多的收敛速度。对于残差范数而言，LOPCGa 的优势更加明显。我们还展示了启发式预条件共轭梯度法（算法 2.1），以说明后续修改中仍有改进空间。

图 2.2 的中间列展示了一种比较情形：所选初始猜测使 LOPCG 在随机测试中的收敛速度处于平均水平。在这种情况下，代码 lobpcg.m 和算法 2.2 的性能相当（因此红色虚线被覆盖），与 LOPCGx 也只有细微差异。此外，EPIC(6,6) 仍具竞争力，且优于 EPIC($2\beta_{\min}$，$2\beta_{\max}$)，但在最终阶段比 LOPCG 慢。RAP 的两种实现表现类似于预条件最速下降法，但 RAP($2\beta_{\min}$，$2\beta_{\max}$) 成为更优的版本。从与启发式预条件共轭梯度法的接近程度来看，LOPCGa 的优势十分显著。

图 2.2 的右列考察的是预条件子精度更低、且 LOPCG 在随机测试中表现最差的情况。我们注意到，EPIC(6, 6) 所需步数与 [27, 图 2(b)] 中的相当。LOPCG 的两个版本并未比 EPIC(6, 6) 慢很多，且 LOPCGx 也未能加速它们；而 LOPCGa 再次最接近启发式预条件共轭梯度法。其他方案的竞争力则要弱得多。另一个事实是，LOPCGa 每步的开销略高于 LOPCG，但低于 EPIC，因为步数主要由矩阵-向量乘法（使用 A、M 和T）决定。因此，LOPCGa 在总计算时间方面也占优。

图 2.3 展示了关于聚集特征值的例子 (2.11, b, c)，以及在随机测试中 LOPCG 最差情况的比较。值得注意的是，LOPCGx 的鲁棒性有所增强，并且大多能加速 LOPCG。此外，在 EPIC 的各个版本以及 RAP 中，EPIC(6, 6) 仍然表现最佳；更具体地说，EPIC 大多能从经验设置中获益，而 RAP 则很少受益。最后，LOPCGa 的出色效率促使我们探究，能否通过更可靠的增广策略进一步改进它。

3. LOPCG 的收敛理论

尽管包括 LOPCG 概念在内的类共轭梯度特征值求解器的收敛性分析起步相当早，但直接估计仅适用于与 LOPCG 相关的某些方案，即基于比较性研究的方案。首个此类方案称为抽象两阶段方法 [9, 第 3 节2]，在我们的设置下其形式为

$$x^{(i+1)} = f\big(T(A-\lambda^{(i)}M)\big)\,x^{(i)} \tag{3.1}$$

其中，f 表示一个抽象矩阵函数。尽管由于不兼容的共轭梯度项 $p^{(i)}$，该方案并不能真正涵盖 LOPCG，但通过将 f 指定为线性多项式，它对应于较弱的方案预条件最速下降法（即不含 $p^{(i)}$ 的 LOPCG），从而可以得到针对 LOPCG 的间接悲观估计。此外，若将 f 指定为移位切比雪夫多项式，则可将 (3.1) 解读为预条件最速下降法或兰索斯型方案的一种加速。在最终阶段，当 $\lambda^{(i)}$ 仅略有变化时，该方法渐近地相当于广义 Davidson 方法的多步迭代；当 $\lambda^{(i)}-\lambda_1$ 足够小时，则渐近地相当于启发式预条件共轭梯度法的多步迭代，由此可得到针对 LOPCG 的间接乐观估计。这类估计的框架如下：

$$\frac{\lambda^{(i+1)}-\lambda_1}{\lambda_2-\lambda^{(i+1)}} \le \xi\,\frac{\lambda^{(i)}-\lambda_1}{\lambda_2-\lambda^{(i)}} \tag{3.2}$$

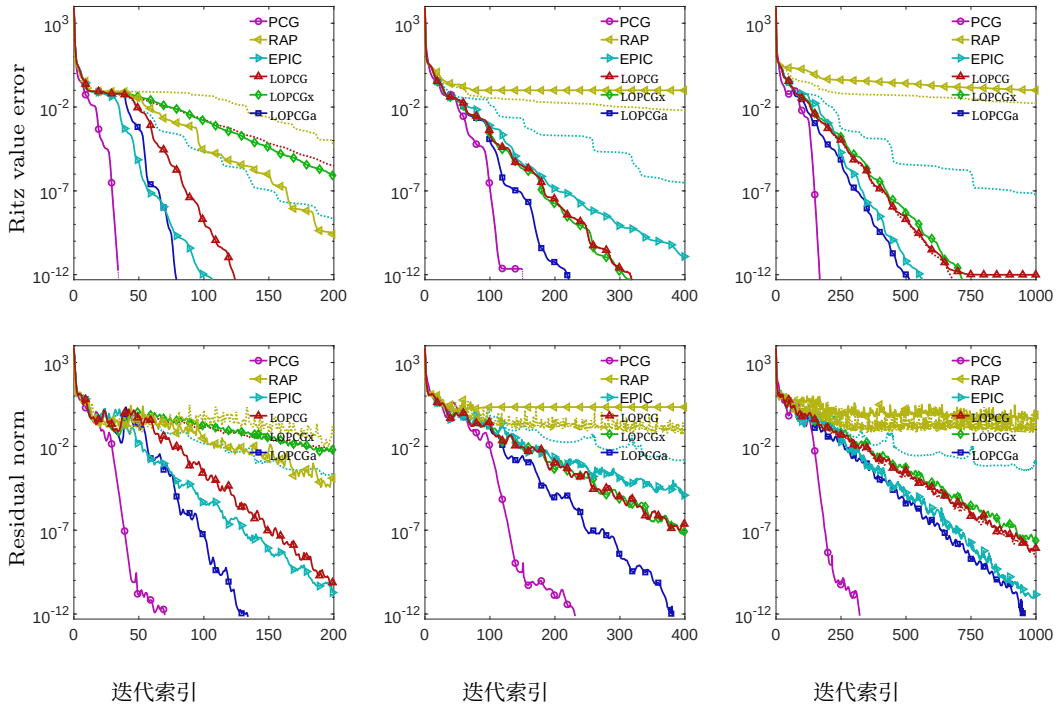


FIGURE 2.2. 示例 (2.11, a) 中类共轭梯度特征值求解器之间的数值比较；详情参见第 2.3节。子图展示了三种情况：LOPCG 实现的表现完全不如、并不逊于或部分不如 EPIC 实现。增广方案 LOPCGa（算法 2.3）显然效率更高。

在假设 $\lambda^{(i)} < \lambda_2$成立的条件下，收敛因子 ξ 可以用一种涉及 $\lambda^{(i)}$ 的繁复形式表示，也可以用一种不含 $\lambda^{(i)}$的简洁形式表示。特别地，在 (3.1) 的切比雪夫刻画中， ξ 与启发式预条件共轭梯度法的估计 (2.4) 中的 $\big(\mathcal{C}_i(\varphi)\big)^{-2}$ 相当。

然而，使用乐观估计有时会带来问题，尤其是在第 2.3 节提到的 LOPCG 困难情形中。此外，为 LOPCG 推导合理的悲观估计（例如，通过推导具有 Rayleigh–Ritz 步长的预条件最速下降法的尖锐估计）在技术上更具挑战性，因此 [9, 第 3 节4] 中的分析最终得到较弱的收敛因子；利用条件数 η（定义见 (2.4)）可将其写为 $\xi = 1 - \eta^{-1}$。预条件最速下降法的下一条重要估计见 [20, 第 6] 节，其形式为 ξ，其中取决于$\lambda^{(i)}$；而在对采用简单步长的预条件最速下降法进行了多项研究之后， [18] 最终完整证明了 [9] 中预期的最终形式 $\xi = \big((\eta-1)/(\eta+1)\big)^2$。这里我们重新表述 [18] 中的主要结果，以便进一步讨论。

引理 3.1。 考虑任意一个不是（A, M）的特征向量的 $x \in \mathbb{C}^n \backslash \{0\}$，并令 θ'为$\mathrm{span}\{x, T$（$Ax - \theta\, Mx$）$\}$ 中（2.1）瑞利商的最小值，其中 $\theta = \lambda$（x）。则$\theta' < \theta$，且

$$\frac{\theta' - \lambda_j}{\lambda_{j+1} - \theta'} \le \xi_j \frac{\theta - \lambda_j}{\lambda_{j+1} - \theta} \quad for \quad \xi_j = \Big(\frac{\eta_j - 1}{\eta_j + 1}\Big)^2, \quad \eta_j = \kappa \frac{(\lambda_n - \lambda_j)(\lambda_{j+1} - \sigma)}{(\lambda_{j+1} - \lambda_j)(\lambda_n - \sigma)} \tag{3.3}$$

对于预条件子的设定 (2.2)，条件是 $\lambda_j \le \theta < \lambda_{j+1}$。

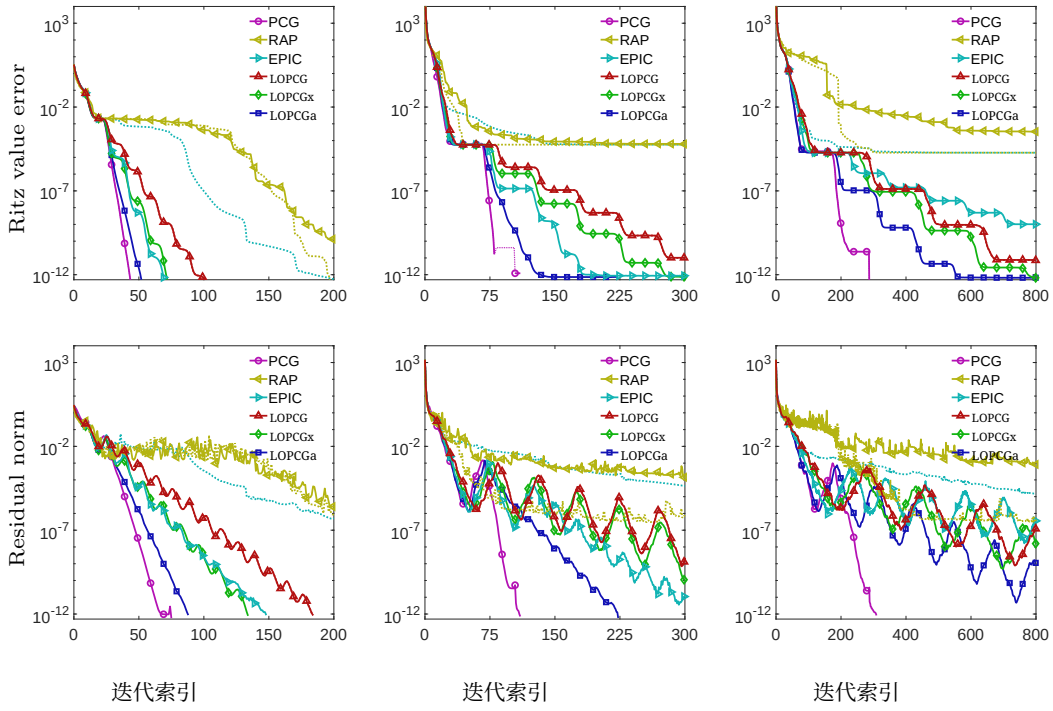


FIGURE 2.3. 示例 (2.11,b, c) 中类共轭梯度特征值求解器的数值比较；详情见第2.3节。各子图尤其表明了 LOPCGa（算法 2.3）的簇鲁棒性：残差范数下降平稳，或振荡幅度较小。

[18] 中的原始表述通过式 (2.2) 描述了预条件子 T，其中的参数对应于 $(\kappa-1)/(\kappa+1)$。其中的分析旨在加速一种受扰动的位移-反演特征值求解器，而非比较用于求解齐次线性系统 (1.2) 的预条件最速下降法与梯度迭代。

注 3.2. 通过考虑 $\theta=\lambda^{(i)}$ 和 $\theta'=\lambda^{(i+1)}$ 将引理 3.1 应用于预条件最速下降法，表明向量迭代至少能够到达与内部特征值 λ_j相关联的特征向量；简单不等式$\theta'<\theta$排除了在 λ_j 处停滞而未实现特征向量收敛的情形。对于从 (2.2) 中任意选取的 T，若不加额外假设，引理 3.1 无法保证收敛至 λ_1，因为存在满足 (3.3) 中等式的示例。其技术原因在于，底层分析以特征值为中心，并与涉及最慢收敛情形的三维子空间有关。一个例外是特殊情形 $T=A_\sigma^{-1}$，此时可基于类似 [9, 定理 1.2] 的夹角估计证明收敛至 λ_1。此外，引理 3.1 为 LOPCG 提供了中间界；LOPCG 的试探子空间包含预条件最速下降法的试探子空间，因此可依据 Courant-Fischer 原理得到特征值近似之间的不等式。相比之下，预条件最速下降法的夹角估计不能（直接）应用于 LOPCG。

注3.3。 式 (3.3) 中的收敛因子 ξ_j 与任何迭代向量都无关，因此该估计可以轻松应用于多个步数。特别地，式 (3.3) 得出了关于 $\lambda^{(i)}<\lambda_2$下预条件最速下降法最终阶段的以下估计：

$$\frac{\lambda^{(i+m)}-\lambda_1}{\lambda_2-\lambda^{(i+m)}}\le\xi^m\,\frac{\lambda^{(i)}-\lambda_1}{\lambda_2-\lambda^{(i)}}\quad\text{for}\quad\xi=\Big(\frac{\eta-1}{\eta+1}\Big)^2,\quad\eta=\kappa\,\frac{(\lambda_n-\lambda_1)(\lambda_2-\sigma)}{(\lambda_2-\lambda_1)(\lambda_n-\sigma)}.\tag{3.4}$$

在相同的设置下，Samokish [25] 提出的首个预条件最速下降法收敛估计可以改写为

$$\lambda^{(i+1)} - \lambda_1 \le \xi(\lambda^{(i)} - \lambda_1)\big(1 + \mathcal{O}(\sqrt{\lambda^{(i)} - \lambda_1})\big), \tag{3.5}$$

而式 (3.4) 与 $m = 1$ 则意味着

$$\lambda^{(i+1)} - \lambda_1 \le \xi(\lambda^{(i)} - \lambda_1)\Big(1 + \frac{(\lambda^{(i)} - \lambda_1) - (\lambda^{(i+1)} - \lambda_1)}{(\lambda_2 - \lambda_1) - (\lambda^{(i)} - \lambda_1)}\Big) = \xi(\lambda^{(i)} - \lambda_1)\big(1 + \mathcal{O}(\lambda^{(i)} - \lambda_1)\big)$$

其中包含一个略优的渐近项。与此相反，非渐近估计 $\lambda^{(i+1)} - \lambda_1 \le \xi(\lambda^{(i)} - \lambda_1)$ 通常并不成立，而且需要使用更大的收敛因子，而不是 ξ；参见 [20, 定理 2.1 和 6.2]，其中收敛因子依赖于 $\lambda^{(i)}$。若不借助 $\lambda^{(i)}$ 来构造合适的收敛因子 ξ，仍难以推导出类似 $\lambda^{(i+m)} - \lambda_1 \le \tilde{\xi}^m(\lambda^{(i)} - \lambda_1)$ 的多步估计。尽管如此，如 [22, 式 (5.9)] 所述，渐近估计 (3.5) 可以推广到重启广义 Davidson 方法。这对应于

$$\lambda^{(i+m)} - \lambda_1 \le \big(\mathcal{C}_m(\varphi)\big)^{-2}(\lambda^{(i)} - \lambda_1)\big(1 + \mathcal{O}(\sqrt{\lambda^{(i)} - \lambda_1})\big) \quad \text{for} \quad \varphi = \frac{\eta + 1}{\eta - 1} \tag{3.6}$$

采用式 (3.4) 的设置以及切比雪夫多项式 $\mathcal{C}_m$。需要指出的是，式 (3.6) 也要求假设 $\lambda^{(i)} < \lambda_2$，且不能直接基于启发式预条件共轭梯度法的类似估计 (2.4) 推导出来。此外，式 (3.4) 中用 $\big(\mathcal{C}_m(\varphi)\big)^{-2}$ 替代 ξ^m，对应于 [11] 中针对 LOPCG 的猜想估计。多步收敛因子 $\big(\mathcal{C}_m(\varphi)\big)^{-2}$ 也可以转换为

$$\big(\mathcal{C}_m(\varphi)\big)^{-2} = \Big(\frac{2\psi^m}{1 + \psi^{2m}}\Big)^2 \quad \text{for} \quad \psi = \frac{\sqrt{\eta} - 1}{\sqrt{\eta} + 1} \tag{3.7}$$

随后通过平均单步收敛因子 ψ^2进行解释。然而，这并不意味着在（3.5）中用 ψ^2 替换 ξ，就能为 LOPCG 提供合适的单步估计。图 3.1 比较了 LOPCG、PSD 和非重启广义 Davidson 方法（GD）在表 2.2 中第一、第四和第五种情形下的表现。上排展示了 Ritz 值误差 $\lambda^{(i)} - \lambda_1$，其中两条水平线分别表示 $\lambda_2 - \lambda_1$ 和（$\lambda_2 - \lambda_1$）/2。下排展示了相应的收敛因子（$\lambda^{(i+1)} - \lambda_1$）/（$\lambda^{(i)} - \lambda_1$），以及两条水平线，分别对应于（3.4）和（3.7）中定义的 ξ 和 ψ^2。显然，一旦 $\lambda^{(i)}$ 达到 λ_2，每种方法的速度都会变慢。此时，即使对于 GD，收敛因子也会“重置”为ξ，从而导致停滞。在 LOPCG 后续的停滞过程中，这种现象可能反复出现，因此其收敛速度无法简单地由 ψ^2界定。另一方面，对于 GD 或 LOPCG 脱离停滞后的收敛加速，ψ^2 并不能给出恰当的估计；对于精度较低的预条件子和聚集特征值，尤其如此，因为此时 ψ^2 接近 ξ；参见图 3.1 左列和右列。

注3.4. 近期关于 LOPCG 收敛理论的研究 [27, 28] 给出了两种替代方法 EPIC 和 RAP 的估计。为简要讨论，我们将这些结果改写为采用 (3.4) 中的设定。EPIC 的估计 [27, 式 (5.15)]（更正一处笔误后）对应于

$$\lambda^{(i+m)} - \lambda_1 \le 2\big(1 - \sqrt{\tilde{\eta}^{-1}}\big)^m(\lambda^{(i)} - \lambda_1) \quad \text{for} \quad \tilde{\eta} = \eta + \mathcal{O}\big(\lambda(q) - \lambda_1\big) \tag{3.8}$$

取决于辅助向量 q，而 RAP 的估计 [28, 式 (3)] 可写为

$$\lambda^{(i+m)} - \lambda_1 \le 2\Big(1 - \frac{1}{2}\sqrt{\tilde{\eta}^{-1}}\Big)^m(\lambda^{(i)} - \lambda_1) \quad \text{for} \quad \tilde{\eta} \approx \kappa\,\frac{\lambda_2 - \sigma}{\lambda_2 - \lambda_1} \tag{3.9}$$

其中，$\tilde{\eta}$ 的精确定义要求在 λ_1 和$(\lambda_1 + \lambda_2)/2$之间引入一个辅助标量 ρ。利用变换(3.7)并忽略辅助项，可以轻松地将估计式(3.8)和(3.9)与(3.6)进行比较。(3.6)中的收敛因子显然小于那些

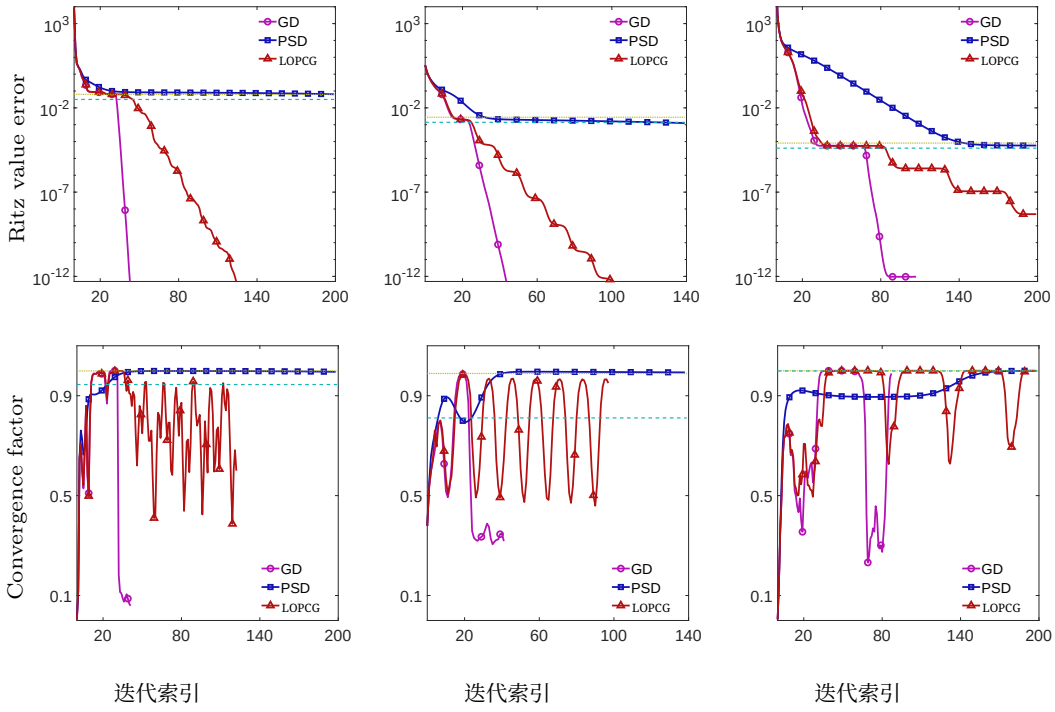


FIGURE 3.1. 类共轭梯度特征值求解器中超出已有估计范围的收敛现象（注 3.3）。上排：$\lambda^{(i)}-\lambda_1$ 的收敛历史，与 $\lambda_2-\lambda_1$ 和 $(\lambda_2-\lambda_1)/2$ （水平线）进行比较；这些水平线大致标示出最终阶段的开始。下排：单步收敛因子 $(\lambda^{(i+1)}-\lambda_1)\,/\,(\lambda^{(i)}-\lambda_1)$，与来自式 (3.4) 和 (3.7) 的估计值 ξ 和 ψ^2 （水平线）进行比较。

出现在(3.8)和(3.9)中的收敛因子，甚至小于相应的平方值。与(3.8)相比，(3.9)中新增的项$_{1\ 2}$ 存在问题，因为单步收敛因子至少为$_{1\ 2}$，这会导致对特征值间隔较大、预条件子更精确的情形作出过高估计。此外，(3.8)和(3.9)是较弱的渐近估计，因为其中的辅助项无法像 $\mathcal{O}(\sqrt{\lambda^{(i)}-\lambda_1})$那样解释为单调递减；例如，由于图 2]，EPIC的最后一个辅助向量 q 大多在最终阶段之前就已选定。

注 3.5。 注 3.4 中的比较，尤其是收敛因子之间的指数差异，表明对于更一般的问题设置，无法通过比较方法对 LOPCG 进行彻底研究。我们还注意到，将非二次共轭梯度最小化理论应用于类共轭梯度特征值求解器的一些潜在缺点已在 [22, 第 4], 节中指出，随后基于渐近等价的双项共轭梯度格式，推导出了 LOPCG 的直接两步估计。该显式估计实质上与 $m=2$ 的 (3.6) 特殊形式一致，实际上可以通过考虑减速形式来重构

$$x^{(i+1)} \xleftarrow{\text{RRw}} x^{(i)}+\operatorname{span}\{Tr^{(i)},p^{(i)}\};\quad x^{(i+2)} \xleftarrow{\text{RRw}} x^{(i+1)}+\operatorname{span}\{Tr^{(i+1)}\}; \tag{3.10}$$

LOPCG每隔一步的迭代都对应于广义 Davidson 方法的两步重启版本。此外， [22, 定理 4.1] 为 LOPCG 提供了一个显著的渐近估计：在满足 $\lambda^{(i)}<(\lambda_1+\lambda_2)/2$的最终阶段，有

$$(\lambda^{(i)}-\lambda^{(i+1)})^{-1}+(\lambda^{(i+1)}-\lambda^{(i+2)})^{-1}=(\lambda^{(i+1)}-\tilde{\lambda}^{(i+2)})^{-1}\big(1+\mathcal{O}(\sqrt{\lambda^{(i)}-\lambda_1})\big) \tag{3.11}$$

其中近似特征值 $\tilde{\lambda}^{(i+2)}$ 由对 $x^{(i+1)}$执行一次 PSD步得到。 [22,定理 4.2] 中的显式估计基于

$$\frac{\lambda^{(i+1)}-\lambda_1}{\lambda^{(i)}-\lambda^{(i+1)}}+\frac{\lambda^{(i+2)}-\lambda_1}{\lambda^{(i+1)}-\lambda^{(i+2)}}=\frac{\tilde{\lambda}^{(i+2)}-\lambda_1}{\lambda^{(i+1)}-\tilde{\lambda}^{(i+2)}}+\mathcal{O}(\sqrt{\lambda^{(i)}-\lambda_1}) \tag{3.12}$$

这是对 (3.11) 的一种变换，即将 (3.11) 两边乘以 $\lambda^{(i+1)}-\lambda_1$，然后减去 1。然而，对于聚集特征值，这种方法存在一些问题。在图3.2中，我们再次考虑表 2.2 中的第一、第四和第五种情形，以及与上述估计相关的形式 $\delta=\mathcal{O}(\sqrt{\lambda^{(i)}-\lambda_1})$。我们将 (3.11) 对应的 δ_1 和 (3.12) 对应的 δ_2 与 $\sqrt{\lambda^{(i)}-\lambda_1}$ 进行比较。与特征值间隔较大的左侧子图相比，另外两个子图表明，在聚集特征值导致数次停滞之后， δ_1 和 δ_2 可以明显超过 $\sqrt{\lambda^{(i)}-\lambda_1}$。此外，我们注意到， [22] 中的方法可以通过使用

$$\delta_3=\left((\lambda^{(i)}-\lambda_1)^{-1/2}+(\lambda^{(i+2)}-\lambda_1)^{-1/2}-2(\tilde{\lambda}^{(i+2)}-\lambda_1)^{-1/2}\right)^{-1}$$

通过切比雪夫多项式的三项递推实现形式 (3.6)。然而，如图 3.2 所示，对于聚集特征值， δ_3 与$\sqrt{\lambda^{(i)}-\lambda_1}$ 也不可比较。

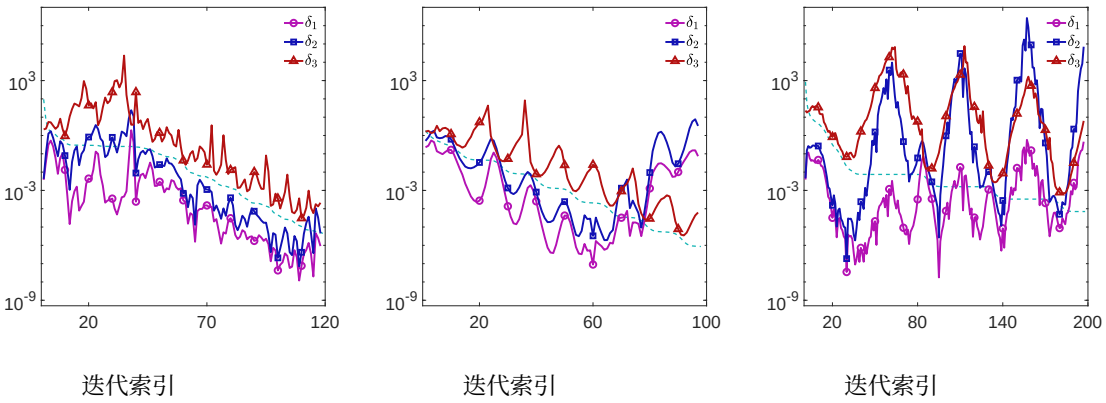


FIGURE 3.2. LOPCG 渐近估计的局限性（注 3.5）。对于聚集特征值，式 (3.11)、(3.12)、(3.6) 的渐近项 δ_1、 δ_2、 δ_3 可能与估计值 $\mathcal{O}\left(\sqrt{\lambda^{(i)}-\lambda_1}\right)$ 的大小相差甚远（中间和右侧子图）。

综上，对于非聚集特征值，现有估计对 LOPCG 有意义；然而，对于$\lambda_1\approx\lambda_2$ 情形，LOPCG 的阶梯状收敛行为仍无法得到充分解释，这是因为 $\left(\mathcal{C}_m\left(\varphi\right)\right)^{-2}\approx 1$。特别是，两次延迟之间的加速仍需在未来研究中通过其他方法加以考察。

4. 两项预条件共轭梯度法（TPCG）方案

现在我们将注意力转向 LOPCG 的两项替代方案。 [21, 22] 这些方案可通过直接修改算法 2.2 得到，如算法 4.1 所示。

其中，关键变化在于，搜索方向 $p^{(i)}$ 的更新不依赖于 Rayleigh-Ritz 过程。此外，根据我们的数值测试，加权形式$x^{(i+1)}=x^{(i)}+\cdots$特别适用于无需频繁归一化的 TPCG，因为它可以避免 $\|x^{(i+1)}\|_M$ 溢出。算法 4.1 的各个变体仅在标量参数上有所不同

算法 4.1： 用于计算 (1.1) 中的 $(\lambda_1,\ x_1)$ 的 TPCG

1 生成 $x^{(0)}$；设置 $p^{(0)}=0$；2 **对于** $i=0,\ 1,\ \ldots\ldots$直到收敛**执行**3 $\lambda^{(i)}=\lambda(x^{(i)})$;
$r^{(i)}=Ax^{(i)}-\lambda^{(i)}Mx^{(i)}$；4 选择 $\iota^{(i)},\ \tau^{(i)}$;
$p^{(i+1)}=\iota^{(i)}Tr^{(i)}+\tau_{(i)}p^{(i)}$；5 $x^{(i+1)}$ RRw
$\longleftarrow$ $x^{(i)}$ +张成$\{p^{(i+1)}\}$，如 (2.10) 中所示;

$\iota^{(i)}$ 和 $\tau^{(i)}$。例如，给定一个辅助 $\gamma^{(-1)}=1$，可以定义

$$
\begin{aligned}
&\text{(a)}\quad \iota^{(i)}=2\,\|x^{(i)}\|_M^{-2}\quad\text{and}\quad \tau^{(i)}=\gamma^{(i)}/\gamma^{(i-1)}\quad\text{for}\quad \gamma^{(i)}=(\iota^{(i)})^2\,\|r^{(i)}\|_T^2,\\
&\text{(b)}\quad \text{alternative for (a)}\quad\text{with}\quad \tau^{(i)}=\big(\gamma^{(i)}-(\iota^{(i)}r^{(i)})^*T(\iota^{(i-1)}r^{(i-1)})\big)/\gamma^{(i-1)},\\
&\text{(c)}\quad \iota^{(i)}=1,\quad \tau^{(0)}=1\quad\text{and}\quad \tau^{(i)}=-\big(p^{(i)*}BTr^{(i)}\big)/\big(p^{(i)*}Bp^{(i)}\big)\quad\text{for}\quad i>0\\
&\qquad\text{with}\quad B=Q_\alpha^*(A-\beta M)Q_\alpha,\quad Q_\alpha=I-\alpha\|x^{(i)}\|_M^{-2}\,x^{(i)}x^{(i)*}M\quad\text{and}\quad \beta\le\lambda^{(i)}.
\end{aligned}
\tag{4.1}
$$

特别地，(4.1, a) 可视为对启发式预条件共轭梯度法算法 2.1 的一种修改：将二次函数 $f(x)=\frac{1}{2}(x^*A_{\lambda_1}x)$ 的梯度替换为瑞利商 (2.1) 的梯度，即 $\iota^{(i)}r^{(i)}$。此外，(4.1, a) 和 (4.1, b) 分别对应于 [22, 第 3], 节中提到的 Bradbury-Fletcher 和 Polak-Ribi´ere 方案，而 (4.1, c) 则与其中的 Daniel（使用 $\alpha=2,\ \beta=\lambda^{(i)}$）、Perdon-Gambolati（使用 $\alpha=0,\ \beta<\lambda_1$）和 Jacobi（使用 $\alpha=1,\beta\le\lambda^{(i)}$）方案相对应，并通过 B-正交性直接定义相邻两个搜索方向之间的共轭关系。在 Daniel 方案中，矩阵 B 是瑞利商 (2.1) 的缩放海森矩阵；在 Perdon-Gambolati 方案中，它是二次函数 $f(x)=\frac{1}{2}(x^*A_{\lambda_1}x)$ 的近似海森矩阵。

Jacobi 方案与目标函数的海森矩阵关系较小，而是与 Jacobi 正交补校正（JOCC）方程 [30] 相关。我们简要介绍 JOCC 方程，并采用算法 4.1 中的设置：忽略这样一个平凡情形，即 $x^{(i)}$ 与目标特征向量 x_1 共线或与其 M 正交；任意非零向量 $y\in\mathrm{span}\{x_1,x^{(i)}\}\cap\mathrm{span}\{x^{(i)}\}^{\perp_M}$（其中上标 $^{\perp_M}$ 表示 M 正交补）均可缩放为 $\tilde{x}^{(i)}=-x^{(i)}+\tau x_1$，使得

$$(A-\lambda_1M)\tilde{x}^{(i)}=-(A-\lambda_1M)x^{(i)}+\tau(A-\lambda_1M)x_1=-r^{(i)}-(\lambda^{(i)}-\lambda_1)Mx^{(i)}.$$

利用 $\mathrm{span}\{x^{(i)}\}^{\perp_M}$ 的投影矩阵，即 (4.1, c) 中的 $Q^{(i)}=Q_1$，以及性质 $\tilde{x}^{(i)}=Q^{(i)}\tilde{x}^{(i)}$，可得 JOCC 方程

$$J_{\lambda_1}^{(i)}\tilde{x}^{(i)}=-r^{(i)}\quad\text{for}\quad J_{\lambda_1}^{(i)}=Q^{(i)*}(A-\lambda_1M)Q^{(i)},\quad Q^{(i)}=I-\|x^{(i)}\|_M^{-2}\,x^{(i)}x^{(i)*}M. \tag{4.2}$$

(4.2) 中的 JOCC 矩阵 $J_{\lambda_1}^{(i)}$ 显然是厄米半正定的，因此可以通过 PCG 迭代启发式地求解 (4.2), 以确定特征向量 $x_1=\tau^{-1}(x^{(i)}+\tilde{x}^{(i)})$。实际应用中，未知特征值 λ_1 会由一个猜测值 $\beta\le\lambda^{(i)}$ 替代，使得搜索方向关于 $J_\beta^{(i)}=Q^{(i)*}(A-\beta M)Q^{(i)}$ 共轭，即 (4.1, c) 中 B 的一种特殊形式。求解 (4.2) 的这一实际版本对应于 Jacobi-Davidson 共轭梯度法（JDCG）方案 [19] 的内层步。从 [22] 得到的 Jacobi 方案大致可视为 JDCG 的一种替代方案，其内层步退化为每个 $J_\beta^{(i)}$ 仅使用一次，并充当两个外层步之间的连接。

此外，[22, 第 3] 节中引人注目的分析表明，除 Perdon-Gambolati 方案外，TPCG 方案在最终阶段都渐近等价于 LOPCG；另见 [21]中对它们的块版本进行的数值比较。因此，可以通过使用 TPCG 来缩短 LOPCG 的总计算时间。我们倾向于采用 Jacobi 方案，因为其构造与 LOPCG 密切相关，因此将其扩展为

使用三维而非四维试探子空间的 LOPCGa 替代方案是合理的。不过，我们注意到， [21]中 Jacobi 方案采用另一种方法更新搜索方向。对于算法 4.1，这意味着 $p^{(i+1)}$ 由 $Tr^{(i)}$ 与上一步中的一个非目标 Ritz 向量的线性组合确定，而不是由 $p^{(i)}$确定。这一区别促使我们给出一种紧凑的推导，并考察直接使用 $p^{(i)}$ 进行更新是否会妨碍性能。

下面对 Jacobi 方案的推导，是基于 [22, 第 3 节。2] 对 LOPCG 进行的一种启发式重构。在算法 2.2 中，前两个试探子空间可表示为

$$\mathcal{V} = \operatorname{span}\{x^{(0)}, x^{(1)}\} \quad \text{and} \quad \mathcal{W} = \operatorname{span}\{x^{(0)}, x^{(1)}, Tr^{(1)}\} = \operatorname{span}\{x^{(0)}, x^{(1)}, x^{(2)}\}.$$

利用 $Q^{(1)}$ 在 (4.2) 中定义的投影矩阵以及搜索方向 $p^{(1)} = x^{(1)} - x^{(0)} \in \mathcal{V}$，我们考虑 $q^{(1)} = Q^{(1)}p^{(1)} \in \mathcal{V} \subset \mathcal{W}$。由于 $(\lambda^{(i)},\ x^{(i)})$ 按构造即为一个 Ritz 对，因此 Ritz 向量残差与相应子空间正交，这意味着

$$0 = q^{(1)*}(A-\lambda^{(1)}M)x^{(1)} = q^{(1)*}(A-\lambda^{(2)}M)x^{(1)} - (\lambda^{(1)}-\lambda^{(2)})q^{(1)*}Mx^{(1)} = q^{(1)*}(A-\lambda^{(2)}M)x^{(1)}$$

且 $0 = q^{(1)*}(A-\lambda^{(2)}M)x^{(2)}$。因此，$(A-\lambda^{(2)}M)q^{(1)}$ 与$x^{(1)}$ 和 $x^{(2)}$的任意线性组合正交，例如 $Q^{(1)}x^{(2)}$。换言之，我们得到

$$0 = (Q^{(1)}x^{(2)})^*(A-\lambda^{(2)}M)q^{(1)} = x^{(2)*}y \quad \text{for} \quad y = Q^{(1)*}(A-\lambda^{(2)}M)Q^{(1)}p^{(1)}$$

因此， $x^{(2)}$ 属于 $\mathcal{U} = \mathcal{W} \cap \operatorname{span}\{y\}^{\perp}$。除平凡情形外， $\mathcal{U}$ 的维数为 2，并且由于 $Q^{(1)}x^{(1)} = 0$而包含 $x^{(1)}$ 。通过将$Tr^{(1)} \in \mathcal{W}$ 对 y 进行正交化，可以在 $\mathcal{U}$ 中得到另一个向量，该向量充当下一个搜索方向 $p^{(2)}$，即，

$$p^{(2)} = Tr^{(1)} - \frac{y^*Tr^{(1)}}{y^*p^{(1)}}\,p^{(1)} = Tr^{(1)} - \frac{p^{(1)*}Q^{(1)*}(A-\lambda^{(2)}M)Q^{(1)}Tr^{(1)}}{p^{(1)*}Q^{(1)*}(A-\lambda^{(2)}M)Q^{(1)}p^{(1)}}\,p^{(1)}. \tag{4.3}$$

然后，对 $\operatorname{span}\{x^{(1)}, p^{(2)}\}$ 应用 Rayleigh-Ritz 过程，得到 $x^{(2)}$；约化试探子空间 $\operatorname{span}\{x^{(1)}, p^{(2)}\} = \operatorname{span}\{x^{(1)}, x^{(2)}\}$ 可用于以相同方式重新表述 LOPCG 的下一步。经过这一系列重新表述，便得到算法 4.1，其中第（i+1）步的（4.3）式对应形式变为更新公式（4.1，c），其中用 $\alpha = 1$ 和位移 $\beta \le \lambda^{(i)}$代替 $\lambda^{(i+1)}$。

此外，由于约化试探子空间是二维的， $q^{(1)} = Q^{(1)}p^{(1)}$ 及其在后续步数中的对应向量 $q^{(i)} = Q^{(i)}p^{(i)}$ 实际上是实现中使用的非目标 Ritz 向量，如 [21]所示。使用这些 Ritz 向量可能是为了避免在 $Q^{(i)}p^{(i)}$中进行正交化。不过，我们注意到， $Q^{(i)}p^{(i)}$ 可以用一种等价形式计算，并且所需标量已可直接使用，因此直接使用 $p^{(i)}$ 进行更新仍然高效。更确切地说，对于 $Q^{(i)}p^{(i)} = p^{(i)} - \|x^{(i)}\|_M^{-2}\,x^{(i)}x^{(i)*}Mp^{(i)}$，可以计算点积

$$x^{(i)*}Mp^{(i)} = (x^{(i-1)} + \delta p^{(i)})^*Mp^{(i)} = x^{(i-1)*}Mp^{(i)} + \delta p^{(i)*}Mp^{(i)}$$

利用 $U = [x^{(i-1)},\, p^{(i)}]$对应的格拉姆矩阵 U^*MU的分量，以及前一步的步长 δ。因此，$Q^{(i)}p^{(i)}$ 可以通过简单的向量更新得到。

作为实际实现中的另一个重要因素，必须指定位移 $\beta \le \lambda^{(i)}$ 。我们注意到， [21] 中的选择 $\beta = \lambda^{(i)}$ 并不总是接近启发式值 $\lambda^{(i+1)}$，尤其是在初始阶段。从这个意义上说，设置 $\beta < \lambda^{(i)}$ 更有意义，例如，

$$\beta = \max\left\{(\sigma + \lambda^{(i)})/2,\ \ 2\lambda^{(i)} - \lambda^{(i-1)}\right\} \tag{4.4}$$

预条件子 (2.2) 的估计下界为 $\sigma < \lambda_1$ 。有趣的是，这个位移也适用于采用 $Q^{(i-1)}p^{(i)}$ 的略微修改后的搜索方向更新（其灵感来自高斯-赛德尔等类似的算法简化）。这种组合不仅不会降低性能，反而略微提升了整体性能。

算法 4.2： 用于计算 (1.1) 中的 $(\lambda_1,\ x_1)$ 的两种指定 TPCG 方案

1 生成 $x^{(0)}$;
2 **对于** $i = 0,\ 1,\ \ldots\ldots$ 直至收敛**执行**
3 $\lambda^{(i)} = \lambda(x^{(i)})$; $r^{(i)} = Ax^{(i)} - \lambda^{(i)}Mx^{(i)}$;
4 **如果** $i > 0$，**那么**
5 $v = Q^{(i)}p^{(i)}$ or $Q^{(i-1)}p^{(i)}$;
6 $w = Av - \beta Mv$，其中 β 定义于 (4.4);
7 $\tau^{(i)} = -(w^*Tr^{(i)})/(w^*v)$;
8 $p^{(i+1)} = Tr^{(i)} + \tau^{(i)}p^{(i)}$ or $Tr^{(i)} + \tau^{(i)}v$;
9 **else**
10 $p^{(i+1)} = Tr^{(i)}$;
11 $x^{(i+1)} \xleftarrow{\text{RRw}} x^{(i)}$ +张成$\{p^{(i+1)}\}$ 或如 (2.10) 中所示， $x^{(i)}$ +张成$\{Q^{(i)}p^{(i+1)}\}$;

上述讨论引出了算法 4.2 中两个具体的 TPCG 方案，它们的区别见第 5、8 和 11 行。对于第一个（也是标准）版本，需要注意，第 7 行中的项 $w^*Tr^{(i)}$ 与 $w^*Q^{(i)}Tr^{(i)}$ 相同，因此并不与 (4.1, c) 中的描述矛盾。这种一致性可直接由以下内容得出：

$$w^*x^{(i)} = p^{(i)*}Q^{(i)*}(A - \beta M)x^{(i)} = p^{(i)*}Q^{(i)*}(A - \lambda^{(i)}M)x^{(i)} = p^{(i)*}r^{(i)} = 0$$

其中，最后一个等式利用了以下事实： $r^{(i)}$ 是先前试探子空间（包含 $p^{(i)}$）的 Ritz 向量残差。关于算法 4.2 的性能，在约化试探子空间上应用 Rayleigh-Ritz 过程只是一项基本计算，因此每步开销仅略高于启发式预条件共轭梯度法，且低于 LOPCG。具体差异取决于三个必不可少的显式矩阵-向量乘法（分别与 A、 M 和 T相乘）在每步开销中所占的比重（等价地，取决于这些矩阵的非零分量密度）。例如，将算法 4.2 的第二个版本应用于表 2.2 中的各个案例，并以启发式预条件共轭梯度法的开销为衡量单位，则 LOPCG 或算法 4.2 的开销分别介于 1.18 至 2.31 之间，或介于 1.06 至 1.39 之间。其中，LOPCG 使用 [12, lobpcg.m], 进行测试；不过，跳过某些对单向量迭代并非必要的代码行，可以实现略微加速。

算法 4.2 可以轻松升级为一种方案，在第2.3节的数值比较中，其效率优于表现最佳的 LOPCGa。试探子空间同样通过辅助向量 a 进行增广，但在 LOPCGa 中用于更新 a 的简单策略 $|\cos\angle_M(a, x^{(i)})| < 0.7$，则通过观察残差范数加以改进，这一做法由下述测试启发。在该测试中，算法 4.2 的第二个版本记为 TPCG，并针对表 2.2 中的第五种情形（包含聚集特征值），将其与 LOPCG 和 LOPCGa 进行比较。如图4.1所示，它们的收敛历史以.....为指标展示

$$\begin{gathered}\lambda:\ \lambda^{(i)} - \lambda_1,\quad \nu:\ \|r^{(i)}\|_2/\|x^{(i)}\|_M,\quad \phi:\ \phi^{(i)} = |\cos\angle_M(a, x^{(i)})| \\ \Delta\lambda:\ |\lambda^{(i)}/\lambda^{(i-1)} - 1|^{1/2},\quad \Delta\phi:\ |\phi^{(i)}/\phi^{(i-1)} - 1|.\end{gathered} \tag{4.5}$$

这一比较清楚地反映了 LOPCG（左图）与 TPCG（中间图）之间的渐近等价。此外，由于图 2.3 中的前一项测试表明，LOPCG 的初始阶段无法显著加速，因此我们只需观察首次延迟之前的数据。根据针对 LOPCGa（右图）的策略 $\phi^{(i)} = |\cos\angle_M(a, x^{(i)})| < 0.7$，我们注意到，$\phi^{(i)}$ for $a = x^{(0)}$ 的显著变化几乎与 ν的峰值同时发生。观察 $\Delta\phi$可以更清楚地看到这种关系。因此，更合理的增广策略是检测残差峰值。另一个显著的现象是， $\Delta\lambda$几乎

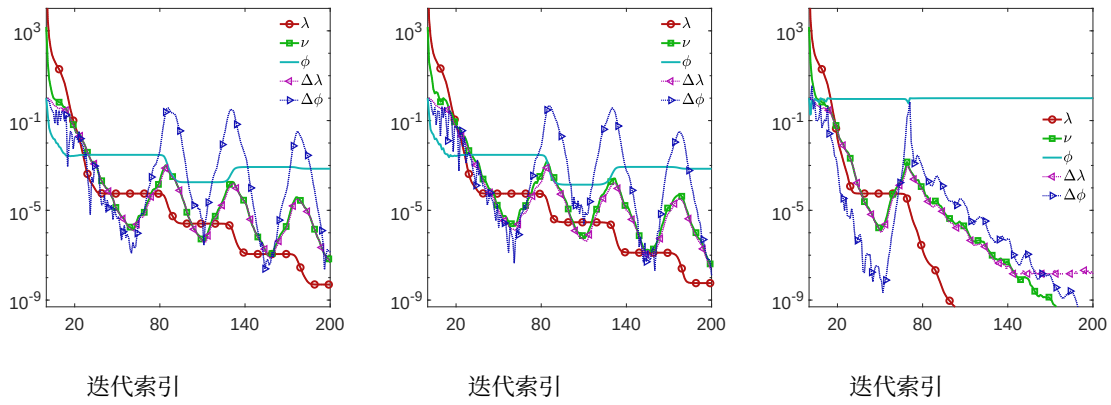


FIGURE 4.1. 根据（4.5），比较 LOPCG 和 TPCG 与 LOPCGa，以确定适时增广的时机。残差范数 ν 的变化可提供良好的时机判断。

与 ν吻合，这可能有助于未来对 LOPCG 和 TPCG 开展研究，提出一种新方法。

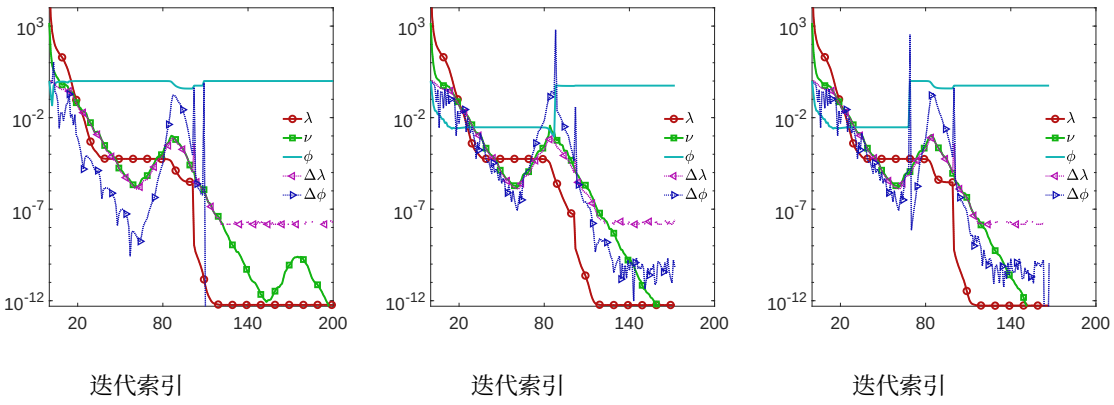


FIGURE 4.2. 比较采用基于角度的判据 $\phi^{(i)} = |\cos \angle_M(a, \ x^{(i)})| < \tau$或基于残差的判据的 TPCG 增广版本。左：采用 $\tau = 0.99$。中间：采用 $\tau = 0.001$。右：残差范数 ν 大幅增加后再次下降时进行更新。

我们进一步指出，采用 $\phi^{(i)} < \tau$ 和 $\tau = 0.7$ 的简单策略并不适合用于 TPCG 的增广，甚至会导致速度下降。图 4.2 左侧和中间的子图展示了采用另外两个阈值 $\tau = 0.99$ 和 $\tau = 0.001$时可接受的性能。后者比前者更有效，表现为初始阶段的更新次数更少，且最终阶段不再出现延迟。此外， a 由 $x^{(0)}$ 初始化，并通过与（已计算的）残差范数 ν的最小值相对应的迭代进行更新。正如我们通过额外测试得出的结论，这主要改善了平凡的更新设置 $a = x^{(i)}$ 。接下来，右侧子图展示了一种便捷的基于残差的策略，可以减少对阈值选择的依赖。在相关尝试中，我们测试了以下三个时间点来更新增广：（i）残差范数 ν显著增加（形成谷值）后立即更新；（ii）当ν 再次下降时更新；（iii）当 ν 小于其谷值时更新。方法（i）和（ii）都能很好地避免未增广 TPCG 或 LOPCG 所产生的延迟。然而，方法（i）会使 ν 谷值之后的上升步骤略微变慢，因此方法（ii）的时机更佳。方法（iii）则不太合适，因为更新后还会进一步产生一次（但仅一次）延迟。因此，图 4.2（右）展示了方法（ii），其中 ν 的无障碍下降路径与其相当

在图 4.2（中）中使用阈值 $\tau = 0.001$ 得到的结果相当。此外， ν 在第 160 步之前降至 1e-12 以下，这意味着相较于图 2.3（中）中观察到的 LOPCG 在第 225 步达到的结果，有了显著改进。

综上所述，我们建议通过定义 $\nu^{(i)} = \|r^{(i)}\|_2/\|x^{(i)}\|_M$并向算法 4.2 中添加以下内容，对其进行修改

第 3 行：存储对应于 $\nu_{\min} = \min_{j\le i}\nu^{(j)}$ 的迭代 $\check{}x$ ，

第 11 行：如果 $\nu^{(i)}$ 达到峰值，则用 $\check{}x$ 增广试探子空间

其中，可以通过从以下位置获取标志 flag=2 来检测峰值

$$\begin{aligned} &\text{if } \texttt{flag==0} \text{ and } \nu^{(i)} > 1.5\,\nu_{\min}, \quad \text{set } \texttt{flag=1},\\ &\text{if } \texttt{flag==1} \text{ and } \nu^{(i)} < \nu^{(i-1)}, \quad \text{set } \texttt{flag=2}. \end{aligned} \tag{4.6}$$

由于增广后的试探子空间维数为 3，因此所得 “TPCGa” 方案每步的开销与 LOPCG 相当，总计算时间的减少直接体现在所需步数上。此外，我们注意到，式 (4.6) 特别适用于计算聚集特征值。在其他情况下，例如图 2.2 中的示例，残差范数可能会频繁振荡，因此难以准确预测最佳时机。此时，检查 ν 是否在多步（例如 3 步）内下降，比检查单步下降 $\nu^{(i)} < \nu^{(i-1)}$ 更有效。最后但同样重要的是，基于残差的判据允许我们在初始阶段省去增广，从而进一步降低总开销。参照图 2.2 和图 2.3，我们在图 4.3 中将这一最终版本的 TPCGa 与 LOPCGa 进行了比较。其中，λ_a 和 ν_a 表示将式 (4.5) 中的 λ 和 ν 项应用于 LOPCGa 后得到的项。

结论

通过模仿或改进用于求解线性方程组的传统共轭梯度迭代，可以构造出具有竞争力的特征值求解器。然而，原始搜索方向的最优性并未得到保留；在计算聚集特征值时，正如 LOPCG 及其近期替代方法呈现的阶梯状收敛历史所示，这种最优性有时还会受到严重干扰。我们的单向量方案 TPCGa 采用增广的二项递推，通过检测残差范数的峰值并及时更新增广，克服了这一缺点。TPCGa 的试探子空间维数较低（为二或三），因此其每步开销甚至低于 LOPCG，从而在总计算时间方面也展现出显著更好的性能（并非仅限于聚集特征值的情形）。这一方法也引出了有关分块实现和收敛性分析的未来研究课题。

附录 A. 典型广义特征值问题

在 [11] 中介绍 LOPCG 时，所关注的是计算最大特征值 o f

$$Mx = \mu Ax, \quad M, A \in \mathbb{C}^{n\times n} \text{ Hermitian}, \quad A \text{ positive definite}. \tag{A.1}$$

这涵盖了特征值问题的两种情形

$$Lu = \lambda Su, \quad L, S \in \mathbb{C}^{n\times n} \text{ Hermitian}, \quad S \text{ positive definite} \tag{A.2}$$

源自对自伴椭圆偏微分算子的离散化。

第一种情形与第1节中介绍的情形一致（令 $L = A$, $S = M$），即，我们关注矩阵对 (L, S) 的最小特征值 λ_1 。为使其符合 ˜(A.1) 的形式，可采用位移 $\sigma < \lambda_1$ ，使得 (A.2) 可改写为 $Lu = (\lambda - \sigma)Su$ ，其中˜移位矩阵 $L = L - \sigma S$ 显然是正定的。于是 (A.1) 对应于

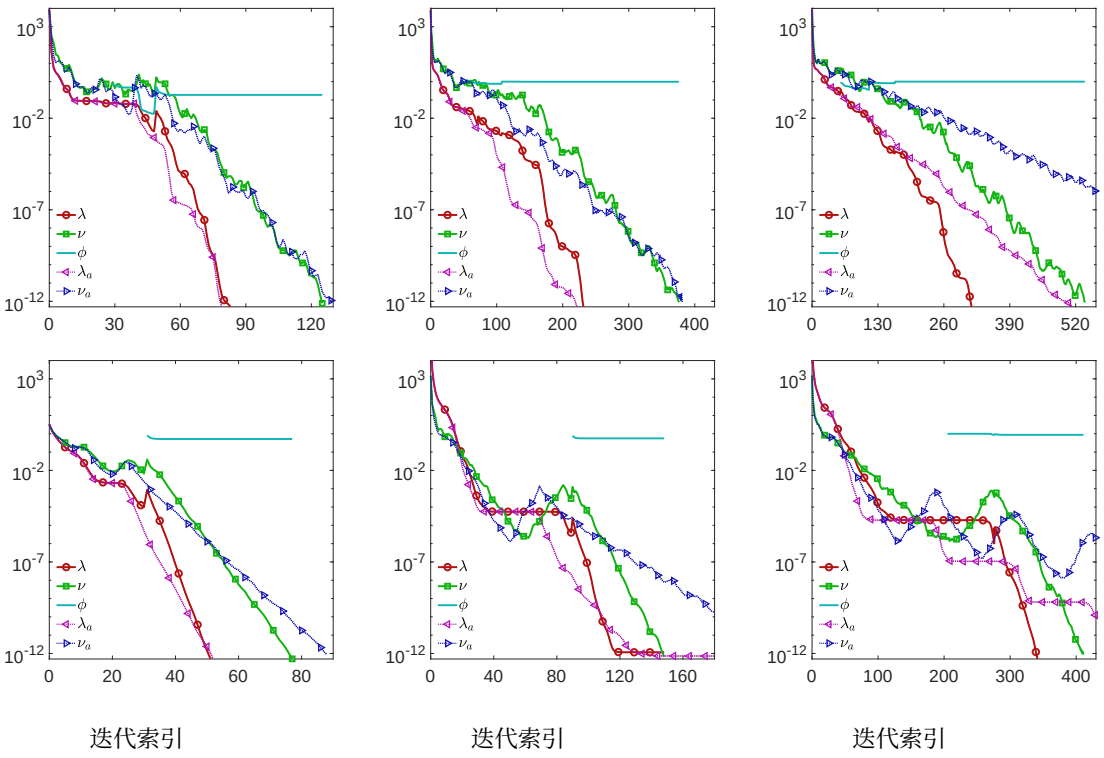


FIGURE 4.3. 比较 TPCGa 与 LOPCGa 在式 (4.5) 下的表现，其中 λ_a 和ν_a 分别表示应用于 LOPCGa 的 λ （Ritz 值误差）和 ν （残差范数）。上/下行可分别与图 2.2/2.3 中的比较结合起来。对于精度较低的预条件子和聚集特征值，TPCGa 的优势十分显著。

令 $M = S$, $A = \tilde{L}$ 以及 $\mu = (\lambda - \sigma)^{-1}$，即可得到 (A.2)。此外，在采用˜预条件特征值求解器进行实际处理时，需要近似求解形如 $Lv = w$ 的线性方程组。

第二种情形讨论位移 σ 接近但不等于某个内部特征值的情况，即$\tilde{L}$是不定且可逆的。此时，式 (A.2) 可改写为 $(\tilde{L}S^{-1}\tilde{L})u = (\lambda - \sigma)\,\tilde{L}u$，并与式 (A.1) 相对应，其中 $M = \pm\tilde{L}$、$A = \tilde{L}S^{-1}\tilde{L}$和 $\mu = \pm(\lambda - \sigma)^{-1}$。其中，符号 $\pm$表示确定大于或小于 σ的特征值。另一种表示形式使用$M = S$、 $A = \tilde{L}S^{-1}\tilde{L}$ 和 $\mu = (\lambda - \sigma)^{-2}$。将预条件特征值求解器应用于这种情形，需要近似求解形如 $(\tilde{L}S^{-1}\tilde{L})v = w$的线性方程组；在实现中，这对应于矩阵 L 的˜两个线性方程组。

(A.1) 的另一种特殊形式涉及厄米定矩阵束 [14]

$$\text{(A.3)} \qquad Lu = \lambda Su, \quad L, S \in \mathbb{C}^{n\times n} \text{ Hermitian}, \quad \tilde{L} = L - \sigma S \text{ definite for a certain } \sigma \in \mathbb{R}.$$

˜若 L 为正定，则变换为 (A.1) 的方式与 (A.2) 第一种情形相同。对于负定的 $\tilde{L}$，则改为设置 $M = S$, $A = -\tilde{L}$ 和 $\mu = -(\lambda - \sigma)^{-1}$。此外，改变 M 和 μ 的符号（两种情形均如此），即可确定σ两侧的特征值。文献 [14, 的定理 2.1] 提出了一种实用的特征值排列方法，该方法考虑了可能为不定且奇异的矩阵 S 的惯性指数。

也可以使用 (A.1) 来描述线性响应特征值问题 [3]。其中，块反对角矩阵的最小正特征值

$$\begin{bmatrix} O & \tilde{L} \\ \tilde{S} & O \end{bmatrix}, \quad \tilde{L}, \tilde{S} \in \mathbb{C}^{\tilde{n}\times\tilde{n}} \text{ Hermitian positive semidefinite, one of them definite} \tag{A.4}$$

需要计算。一个简单的改写利用了这样一个事实：目标特征值是矩阵对 $(\tilde{S},\ \tilde{L}^{-1})$ 或 $(\tilde{L},\ \tilde{S}^{-1})$ 中最小非零特征值的平方根，前提是 $\tilde{L}$ 或 $\tilde{S}$ 为定矩阵。显然，这包含在 (A.2) 的第一种情形中。若 $\tilde{L}$ 和 $\tilde{S}$ 均为定矩阵，则 [14] 中提出的一种改写形式为

$$Lu = \lambda S u \quad \text{with} \quad L = \begin{bmatrix} \tilde{L} & O \\ O & \tilde{S} \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} O & \tilde{I} \\ \tilde{I} & O \end{bmatrix} \tag{A.5}$$

其中 I 表示 $\tilde{n}\times\tilde{n}$ 单位矩阵。这与 (A.3) 相符，其中 $\sigma = 0$。将 (A.2) 和 (A.3) 转换为 (A.1)，即可基于求解 (A.1) 的方法构造合适的特征值求解器。

参考文献

[1] H.M. Aktulga, M. Afibuzzaman, S. Williams, A. Buluc, M. Shao, C. Yang, E.G. Ng, P. Maris, and J.P. Vary, *A high performance block eigensolver for nuclear configuration interaction calculations*, IEEE TPDS 28 (2017), 1550–1563. https://doi.org/10.1109/TPDS.2016.2630699 [2] P. Arbenz, U.L. Hetmaniuk, R.B. Lehoucq, and R.S. Tuminaro, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Int. J. Numer. Meth. Engng 64 (2005), 204–236. https://doi.org/10.1002/nme.1365 [3] Z. Bai and R.C. Li, *Minimization principles for the linear response eigenvalue problem I: Theory*, SIAM J. Matrix Anal. Appl. 33 (2012), 1075–1100. https://doi.org/10.1137/110838960 [4] Z. Bai and R.C. Li, *Minimization principles for the linear response eigenvalue problem II: Computation*, SIAM J. Matrix Anal. Appl. 34 (2013), 392–416. https://doi.org/10.1137/110838972 [5] Y. Cai, Z. Bai, J.E. Pask, and N. Sukumar, *Convergence analysis of a locally accelerated preconditioned steepest descent method for Hermitian-definite generalized eigenvalue problems*, J. Comp. Math. 36 (2018), 739–760. https://doi.org/10.4208/jcm.1703-m2016-0580 [6] J.K. Cullumand R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol.I: Theory*, SIAM, 费城, 2002. https://doi.org/10.1137/1.9780898719192 [7] J.A. Duersch, M. Shao, C. Yang, and M. Gu, *A robust and efficient implementation of LOBPCG*, SIAM J. Sci. Comput. 40 (2018), C655–C676. https://doi.org/10.1137/17M1129830 [8] U. Hetmaniuk and R. Lehoucq, *Basis selection in LOBPCG*, J. Comput. Phys. 218 (2006), 324–332. https://doi.org/10.1016/j.jcp.2006.02.007 [9] A.V. Knyazev, *Convergence rate estimates for iterative methods for a mesh symmetric eigenvalue problem*, Russian J. Numer. Anal.Math. Modelling 2 (1987), 371–396. https://doi.org/10.1515/rnam.1987.2.5.371 [10] A.V. Knyazev, *Preconditioned eigensolvers—an oxymoron?* 《数值分析电子学报》 7 (1998), 104– 123. https://etna.ricam.oeaw.ac.at/volumes/1993-2000/vol7 [11] A.V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conju-gate gradient method*, SIAM J. Sci. Comput. 23 (2001), 517–541. https://doi.org/10.1137/S1064827500366124 [12] A.V. Knyazev, M. E. Argentati, I. Lashuk, and E.E. Ovtchinnikov *Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in hypre and PETSc*, SIAM J. Sci. Comput. 29 (2007), 2224–2239. https://doi.org/10.1137/060661624, https://github.com/lobpcg/blopex [13] A.V. Knyazev and K. Neymeyr, *A geometric theory for preconditioned inverse iteration III: A short and sharp convergence estimate for generalized eigenvalue problems*, Linear Algebra Appl. 358 (2003), 95–114. https://doi.org/10.1016/S0024-3795(01)00461-X [14] D. Kressner, M.M. Pandur, and M. Shao, *An indefinite variant of LOBPCG for definite matrix pencils*, Numer. Algor. 66 (2014), 681–703. https://doi.org/10.1007/s11075-013-9754-3 [15] L. Lin, S. Shao, and W. E, *Efficient iterative method for solving the Dirac-Kohn-Sham density functionaltheory*, J. Comput. Phys. 245 (2013), 205–217. https://doi.org/10.1016/j.jcp.2013.03.030 [16] A. Levitt and M. Torrent, *Parallel eigensolvers in plane-wave density functional theory*, Comput. Phys. Commun. 187 (2015), 98–105. https://doi.org/10.1016/j.cpc.2014.10.015 [17] K. Neymeyr, *On preconditioned eigensolvers and Invert-Lanczos processes*, Linear Algebra Appl. 430 (2009), 1039–1056. https://doi.org/10.1016/j.laa.2008.10.016

[18] K. Neymeyr, *A geometric convergence theory for the preconditioned steepest descent iteration*, SIAM J. Numer. Anal. 50 (2012), 3188–3207. https://doi.org/10.1137/11084488X[19] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl. 9 (2002), 21–44. https://doi.org/10.1002/nla.246[20] E.E. Ovtchinnikov, *Sharp convergence estimates for the preconditioned steepest descent method for Hermit-ian eigenvalue problems*, SIAM J. Numer. Anal. 43 (2006), 2668–2689. https://doi.org/10.1137/040620643[21] E.E. Ovtchinnikov, *Computing several eigenpairs of Hermitian problems by conjugate gradient iterations*, J. Comput. Phys. 227 (2008), 9477–9497. https://doi.org/10.1016/j.jcp.2008.06.038[22] E.E. Ovtchinnikov, *Jacobi correction equation, line search, and conjugate gradients in Hermitian eigenvalue computation I: Computing an extreme eigenvalue*, SIAM J. Numer. Anal. 46 (2008), 2567–2592. https://doi.org/10.1137/070688742[23] Y. Saad, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal. 17 (1980), 687–706. https://doi.org/10.1137/0717059[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 费城, 2003.https://doi.org/10.1137/1.9780898718003[25] B.A. Samokish, *The steepest descent method for an eigenvalue problem with semi-bounded operators*, Izv. Vyssh. Uchebn. Zaved. Mat. 5 (1958), 105–114 (俄文). https://www.mathnet.ru/eng/ivm2983[26] M. Shao, H.M. Aktulga, C. Yang, E.G. Ng, P. Maris, and J.P. Vary, *Accelerating nuclear configuration interaction calculations through a preconditioned block iterative eigensolver*, Comput. Phys. Commun. 222 (2018), 1–13. https://doi.org/10.1016/j.cpc.2017.09.004[27] N. Shao, W. Chen, and Z. Bai, *EPIC: A provable accelerated eigensolver based on preconditioning andimplicit convexity*, SIAM J. Matrix Anal. Appl. 46 (2025), 45–73. https://doi.org/10.1137/24M1641440[28] N. Shao and W. Chen, *Riemannian acceleration with preconditioning for symmetric eigenvalue problems*, Numer. Math. 157 (2025), 307–354. https://doi.org/10.1007/s00211-025-01451-0[29] A.Stathopoulos, *Nearly optimal preconditioned methods for Hermitian eigenproblems under limited memory. Part I: Seeking one eigenvalue*, SIAM J. Sci. Comput. 29 (2007), 481–514.https://doi.org/10.1137/050631574 [30] G.L.G. Sleijpen and H.A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. 17 (1996), 401–425. https://doi.org/10.1137/S0895479894270427[31] L. Wu, F. Xue, and A. Stathopoulos, *TRPL+K: Thick-restart preconditioned Lanczos+K method for large symmetric eigenvalue problems*, SIAM J. Sci. Comput. 41 (2019), A1013–A1040. https://doi.org/10.1137/17M1157568[32] M. Zhou, Z. Bai, Y. Cai, and K. Neymeyr, *Convergence analysis of a block preconditioned steepest descenteigensolver with implicit deflation*, Numer. Linear Algebra Appl. 30 (2023), e2498.https://doi.org/10.1002/nla.2498

Universität Rostock, Institut für Mathematik, Ulmenstrasse 69, 18055 Rostock, Germany

E-mail address: ming.zhou at uni-rostock (dot) de

Universität Rostock, Institut für Mathematik, Ulmenstrasse 69, 18055 Rostock, Germany

E-mail address: klaus.neymeyr at uni-rostock (dot) de