



广义二次 Arnoldi 方法的隐式重新启动位移策略

龚方徽, 孙玉泉*

北京航空航天大学数学与系统科学学院数学信息与行为教育部重点实验室, 北京 100191
E-mail: fanghui.gong@buaa.edu.cn, sunyq@buaa.edu.cn

收稿日期: 2015-11-08; 接受日期: 2016-04-05; 网络出版日期: 2016-11-21; * 通信作者
国家自然科学基金 (批准号: 11201020) 资助项目

摘要 在隐式重新启动的广义二次 Arnoldi 方法中, 将二次特征值问题显式投影到 m 维子空间中可得到 $2m$ 个近似特征对, 在进行隐式重新启动时会存在位移个数与子空间维数不匹配的问题. 针对此困难, 本文给出一种新的可使用全部位移信息的位移策略, 证明该方法既能保持原方法的特殊结构, 也能充分利用位移信息提高算法的效率. 数值算例验证了新的位移策略通过提高每一次重新启动的效率, 有效地提高了算法的整体效率.

关键词 二次特征值问题 GSOAR (generalized second-order Arnoldi) 方法 隐式重启 位移

MSC (2010) 主题分类 65F15, 65Y20

1 引言

二次特征值问题 (quadratic eigenvalue problem, QEP)

$$Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = 0 \quad (1.1)$$

在机械振动、声学的动态分析、流体力学、微电子震荡模拟和结构力学等许多领域都有广泛应用 (参见文献 [1-3]). 求解大规模二次特征值问题的方法可分为两大类: 一是将二次特征值问题化为等价的线性特征值问题:

$$\begin{bmatrix} -C & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix}, \quad (1.2)$$

然后利用求解线性特征值问题的方法进行求解. 该方法的优点是求解广义特征值问题已有成熟的理论和算法 (参见文献 [4, 5]). 但该方法不能保持原二次特征值问题的相关结构和性质, 并会使原问题的规模翻倍, 且可能遇到稳定性的问题 (参见文献 [6]). 第二类方法是对二次特征值问题使用直接投影法, 将原问题投影为中小规模的二次特征值问题进行求解. 这类方法具有较好的数值稳定性, 而且能够保持原问题的结构和物理性质等, 但缺少完备的理论分析和实用的算法.

Krylov 子空间方法是当前求解大规模特征值问题的最有效方法之一. Bai 和 Su^[7] 将 Krylov 子空间的定义推广至二阶, 并提出了求解大型稀疏二次特征值问题的二次 Arnoldi 方法 (second-order

英文引用格式: Gong F H, Sun Y Q. A new shift strategy for the implicitly restarted generalized second-order Arnoldi method (in Chinese). *Sci Sin Math*, 2017, 47: 635-650, doi: 10.1360/012016-22

Arnoldi method, SOAR). 该方法用 SOAR 过程产生二阶 Krylov 子空间的一组标准正交基, 将二次特征值问题 (1.1) 投影至该子空间上, 通过计算得到的 Ritz 对来近似 (1.1) 的特征对. 文献 [8] 验证了 Rayleigh-Ritz 正交投影技术一致整体收敛的性质, 并且改进了作用于二次特征值问题的 Rayleigh-Ritz 正交投影技术.

大型矩阵计算的投影方法中, 由于受存储空间和计算量的限制, 算法需要进行重新启动. 隐式重新启动是一个强大并且应用广泛的重启方式 (参见文献 [9]). 但由于 SOAR 方法中初始向量的特殊结构, SOAR 过程无法直接进行隐式重新启动 (参见文献 [10, 11]). 故在 Bai 和 Su 的 SOAR 方法基础上, 众多学者针对其隐式重启作出研究. Otto^[11] 提出了用普通初始向量代替原有特殊的初始向量, 得到了广义 SOAR (generalized second-order Arnoldi, GSOAR) 方法, 在没有收缩这一较强的假设前提下, 此方法可以选择准确位移进行隐式重新启动. Zhou^[12] 作了一个相似的改进, 提出了显示重新启动的广义 Krylov 子空间方法. Huang^[13] 提出了半正交广义 Arnoldi 过程 (semiorthogonal generalized Arnoldi, SGA) 过程, 并相应地提出了隐式重新启动的 SGA 方法. 这些方法的主要思路是, 利用二次特征值问题的特殊结构, 在不增加存储量的条件下, 将 SOAR 方法变为正交投影方法. Beeumen 等^[14] 对此提出了一个统一的框架, 对方法的执行和重启等进行了全面的总结. 根据这类方法的特点, 从数学原理上来讲, 这些方法都属于线性化方法的范畴.

在各种改进的 SOAR 方法中, Jia 和 Sun^[15] 探讨了 GSOAR 方法的更多性质, 并提出了一种精化的广义二次 Arnoldi (refined generalized second-order Arnoldi, RGSOAR) 方法, 这类方法与原始 SOAR 方法类似, 从数学上属于直接投影方法的范畴. 该方法在进行隐式重新启动时存在候选位移个数与子空间维数不匹配的问题, 需要使用恰当的位移选择策略, Jia 和 Sun^[15] 给出了一种计算准确候选位移与精化候选位移的可靠高效算法. 从学者们的研究可以看出, 选择好的位移对算法的收敛至关重要 (也可参见文献 [11]), 与线性特征值问题的隐式重新启动算法不同, GSOAR 方法在重新启动中若不能使用全部的候选位移, 必然会影响算法的整体效率.

在 Jia 和 Sun 的工作基础上, 本文提出一种可使用全部候选位移的隐式重新启动 GSOAR 和 RGSOAR 位移策略. 在隐式重启过程中, 我们通过使用恰当的位移, 把不需要的特征向量的信息从已有的子空间中剔除, 保留需要的相关信息进行重新启动. 一般而言, 剔除的不需要的信息越多, 更新后得到的子空间就越好. 对于二次特征值问题, 一个特征向量可对应两个不同的特征值, 若一个特征向量对应的特征值都没有被选入位移, 则该信息将不会被剔除, 而使用全部候选位移可以避免这种现象. 但在实际执行时, 这会破坏特征值问题原有的结构, 使算法无法进行重新启动. 本文运用一个巧妙的变换使被破坏的结构重新变为 Hessenberg 形式. 使用全部位移, 不需要的信息能被更彻底地移除, 因此, 每次重启可得到含有更多有用信息的子空间. 由于文献 [15] 中的隐式重启方案并无约束条件, 故本文提出的算法本质上无新加约束条件.

本文结构如下: 第 2 节介绍 SOAR 方法、GSOAR 方法、隐式重新启动和准确位移与精化位移的选择; 第 3 节介绍使用全部候选位移的重新启动策略; 第 4 节使用数值算例验证本文提出的隐式重启位移策略算法的有效性.

2 隐式重新启动 GSOAR 方法与 RGSOAR 方法

SOAR 方法是由 Bai 和 Su^[7] 提出的基于二阶 Krylov 子空间的求解大型稀疏二次特征值问题的方法.

定义 1 存在矩阵 $A, B \in \mathbb{C}^{n \times n}$, 非零向量 $u \in \mathbb{C}^n$, 定义

$$\begin{aligned} r_0 &= u, \\ r_1 &= Au, \\ r_j &= Ar_{j-1} + Br_{j-2}, \quad j \geq 2, \end{aligned}$$

称向量序列 $r_0, r_1, r_2, \dots, r_{m-1}$ 为基于矩阵 A, B 和向量 u 的二阶 Krylov 子空间序列; 将空间

$$\mathcal{G}_m(A, B; u) = \text{span}\{r_0, r_1, r_2, \dots, r_{m-1}\} \quad (2.1)$$

称为基于矩阵 A 和 B 以及向量 u 的 m 维二阶 Krylov 子空间.

二阶 Krylov 子空间 $\mathcal{G}_j(A, B; u)$ 可作为 QEP 的投影子空间, 只需要构造一个标准正交基 $\{q_j\}$, $j = 1, \dots, m$ 即可:

$$\mathcal{G}_j(A, B; u) = \text{span}\{q_1, q_2, \dots, q_j\}, \quad j \geq 1. \quad (2.2)$$

Bai 和 Su 提出了一个用来构造二阶 Krylov 子空间的一组标准正交基序列 $\mathcal{G}_j(A, B; u)$ 的方法.

在此基础上, 广义的 SOAR (GSOAR) 过程和对应的广义二阶 Krylov 序列以及相应的子空间定义如下:

定义 2 ^[15] 存在矩阵 $A, B \in \mathbb{C}^{n \times n}$, 非零向量 $u_1, u_2 \in \mathbb{C}^n$, 定义

$$\begin{aligned} r_0 &= u_1, \\ r_1 &= Ar_0 + Bu_2, \\ r_j &= Ar_{j-1} + Br_{j-2}, \quad j \geq 2, \end{aligned}$$

称向量序列 $r_0, r_1, r_2, \dots, r_{m-1}$ 为基于 A, B, u_1 和 u_2 的广义二阶 Krylov 序列; 将空间

$$\mathcal{G}_m(A, B; u_1, u_2) = \text{span}\{r_0, r_1, r_2, \dots, r_{m-1}\}$$

称为基于矩阵 A 和 B 以及向量 u_1 和 u_2 的 m 维广义二阶 Krylov 子空间.

由 GSOAR 过程可得到算法如下:

算法 1 克服收缩的 GSOAR 过程:

```

 $q_1 = u_1 / \|u_1\|_2$ 
 $p_1 = u_2 / \|u_2\|_2$ 
for  $j = 1, 2, \dots, k$  do
   $r = Aq_j + Bp_j$ 
   $s = q_j$ 
  for  $i = 1, 2, \dots, j$ 
     $t_{ij} = q_i^* r$ 
     $r = r - t_{ij} q_i$ 
     $s = s - t_{ij} p_i$ 
  end for
   $t_{j+1j} = \|r\|$ 
  if  $t_{j+1j} = 0$ 
    if  $s \in \text{span}\{p_i \mid i : q_i = 0, 1 \leq i \leq j\}$ 
      break

```

```

else deflation
reset  $t_{j+1j} = 1$ 
 $q_{j+1} = 0$ 
 $p_{j+1} = s$ 
end if
else
 $q_{j+1} = r/t_{j+1j}$ 
 $p_{j+1} = s/t_{j+1j}$ 
end if
end for
    
```

在算法中若取 $u_2 = 0$, 则该算法就是标准的 SOAR 过程. 算法运行到第 j 步时, 若 $\{r_i\}$ ($i = 0, 1, \dots, j$) 线性相关, 而 $\{[r_i^T, r_{i-1}^T]^T\}$ ($i = 0, \dots, j$) 线性无关, 其中 $r_{-1} = 0$, 这种情形称为“收缩”; 若 $\{[r_i^T, r_{i-1}^T]^T\}$ 与 $\{r_i\}$ 均线性相关, 则称为“中断”. 若在第 j 步发生“收缩”这种情形时, 二阶 Krylov 子空间 $\mathcal{G}_j(A, B; u)$ 并不含有二次特征值问题 (1.1) 的任一精确特征向量 (参见文献 [15]). 故对“收缩”现象采取的补救措施是将 t_{j+1j} 重置为一个非零值 (例如 1), 令 $q_{j+1} = 0$, 继续执行算法. 向量序列 $\{q_j\}$ 中的非零向量仍然正交, 并且构成子空间 $G_j(A, B; u)$ 的一组标准正交基.

令正交矩阵 $Q_m = [q_1, q_2, \dots, q_m] \in \mathcal{C}^{n \times m}$, $P_m = [p_1, p_2, \dots, p_m] \in \mathcal{C}^{n \times m}$, 不可约上 Hessenberg 矩阵

$$\hat{T}_m = [t_{ij}] = \begin{bmatrix} T_m \\ t_{m+1,m} e_m^T \end{bmatrix} \in \mathcal{C}^{(m+1) \times m},$$

得到 m 步 GSOAR 分解:

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_{m+1} \\ P_{m+1} \end{bmatrix} \hat{T}_m, \quad (2.3)$$

其中 $Q_{m+1} = [Q_m, q_{m+1}]$, $P_{m+1} = [P_m, p_{m+1}]$, $H = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$.

在 (1.2) 两端左乘 M^{-1} 可将其化为一个标准特征值问题, 并令 $A = -M^{-1}C$, $B = -M^{-1}K$, 则可由算法 1 产生广义二阶 Krylov 子空间. 在实际应用中常需要求解某一特定数值 σ 附近的特征值 λ , 此特定数值称为目标点. 我们可通过平移逆变换 $\rho = \frac{1}{\lambda - \sigma}$ 将 QEP (1.1) 转化为新的 QEP:

$$Q(\rho)x = (\rho^2 M_\sigma + \rho C_\sigma + K_\sigma)x = 0, \quad (2.4)$$

其中 $M_\sigma = \sigma^2 M + \sigma C + K$, $C_\sigma = C + 2\sigma M$, $K_\sigma = M$. 令 $A = -(\sigma^2 M + \sigma C + K)^{-1}(C + 2\sigma M)$, $B = -(\sigma^2 M + \sigma C + K)^{-1}M$, 可由算法 1 产生关于 (2.4) 的广义二阶 Krylov 子空间. 显然, $\lambda = \frac{1}{\rho} + \sigma$, 且对于给定的目标点, 算法也仅需要一次矩阵求逆.

将原 QEP (1.1) 显式投影至广义二阶 Krylov 子空间 $\mathcal{G}_m(A, B; u_1, u_2)$ 上, 可得到一个小规模 QEP:

$$(\theta^2 M_m + \theta C_m + K_m)g = 0, \quad (2.5)$$

其中 $M_m = Q_m^T M Q_m$, $C_m = Q_m^T C Q_m$, $K_m = Q_m^T K Q_m$. 通过计算 (2.5) 的特征对, 可得到 (1.1) 的 Ritz 值 θ 和 Ritz 向量 $y = Q_m g$, 由此我们得到 GSOAR 方法. 假设已通过 GSOAR 方法得到 Ritz 值 θ ,

对于每一个 θ , 找一个单位向量 $\tilde{u} \in \mathcal{G}_m(A, B; u_1, u_2)$ 满足如下最优条件:

$$\|(\theta^2 M + \theta C + K)\tilde{u}\| = \min_{\substack{u \in \mathcal{G}_m(A, B; u_1, u_2) \\ \|u\|=1}} \|(\theta^2 M + \theta C + K)u\|, \quad (2.6)$$

并用它作为近似特征向量, 称为精化 Ritz 向量, 同时称 (θ, \tilde{u}) 为精化 Rayleigh-Ritz 对, 将此方法称为精化广义二次 Arnoldi (refined generalized SOAR, RGSOAR) 方法. 精化向量可通过求解单位向量 $\tilde{z} \in \mathcal{C}^m$ 得到, $\tilde{u} = Q_m \tilde{z}$, 向量 \tilde{z} 满足

$$\tilde{z} = \arg \min_{\substack{z \in \mathcal{C}^m \\ \|z\|=1}} \|(\theta^2 M + \theta C + K)Q_m z\|. \quad (2.7)$$

对于上式, 可以通过用适中的计算成本计算小规模矩阵 $(\theta^2 M Q_m + \theta C Q_m + K Q_m)$ 的最小奇异值对应的右奇异向量得到 (参见文献 [15]).

2.1 隐式重新启动

在 SOAR 和 GSOAR 方法中, 为了保证近似特征对的精度和好的收敛性, 理论上 m 的取值应该尽可能大. 但是随着子空间维数 m 的不断变大, 对于计算量和存储空间的要求也逐渐变大, 故对于一个实际算法而言, 重新启动是必要的. 若 GSOAR 过程未发生收缩, 则可直接将隐式重新启动的策略用于该方法 (参见文献 [9]).

一个不可约 m 步 GSOAR 分解 (2.3), 可通过变换并截断为一个不可约 k ($k < m$) 步 GSOAR 分解, 实现隐式重新启动. 以 T_m 的一个特征值 μ 为位移, 在其隐式重新启动过程中, 首先对 T_m 运行带一个位移 μ 的 QR 分解, 得

$$T_m - \mu I = VR, \quad (2.8)$$

其中 R 为上三角矩阵, V 为上 Hessenberg 的正交矩阵, 由 (2.8) 整理可得

$$T_m^1 = RV + \mu I = V^T T_m V = \begin{bmatrix} T_{m-1}^1 & t^1 \\ t_{m,m-1}^1 e_{m-1}^T & \mu \end{bmatrix}. \quad (2.9)$$

T_{m-1}^1 仍然为上 Hessenberg 矩阵, 将 V 右乘到 (2.3) 两边, 得到

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V (V^T T_m V) + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T V. \quad (2.10)$$

由于 V 是上 Hessenberg 矩阵, 故 $e_m^T V = (0, \dots, 0, v_{m,m-1}, v_{m,m})$. 将 (2.10) 化为如下形式:

$$H \begin{bmatrix} \tilde{Q}_{m-1} & \tilde{q}_m \\ \tilde{P}_{m-1} & \tilde{p}_m \end{bmatrix} = \begin{bmatrix} \tilde{Q}_{m-1} & \tilde{q}_m \\ \tilde{P}_{m-1} & \tilde{p}_m \end{bmatrix} \begin{bmatrix} T_{m-1}^1 & t^1 \\ t_{m,m-1}^1 e_{m-1}^T & \mu \end{bmatrix} + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} [v_{m,m-1} e_{m-1}^T \quad v_{m,m}]. \quad (2.11)$$

取 (2.11) 的前 $m-1$ 列, 我们可得到 $m-1$ 步 GSOAR 分解.

给定 l 个位移 $\mu_1, \mu_2, \dots, \mu_l$, 对 (2.3) 中的矩阵 T_m 运用 l 次带位移的 QR 分解:

$$(T_m - \mu_1 I) \cdots (T_m - \mu_l I) = V_m R, \quad (2.12)$$

其中 V_m 是大小为 $m \times m$ 的正交矩阵, R 为 $m \times m$ 的上三角矩阵. 根据 (2.3), 得到

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m (V_m^T T_m V_m) + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T V_m. \quad (2.13)$$

经简单推导可知, 矩阵 $V_m^T T_m V_m$ 仍是上 Hessenberg 矩阵. 若令

$$\begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m, \quad V_m^T T_m V_m = \tilde{T}_m,$$

则有

$$H \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} = \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} \tilde{T}_m + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} b_m^T, \quad (2.14)$$

其中 $b_m^T = (0, \dots, 0, b_{m-l}, \dots, b_{m-1}, b_m)$, b_m^T 至少含有 $m-l-1$ 个零元素.

对一个 m 步 GSOAR 分解 (2.14), 可截断为一个 k 步的 GSOAR 分解

$$H \begin{bmatrix} \tilde{Q}_k \\ \tilde{P}_k \end{bmatrix} = \begin{bmatrix} \tilde{Q}_k \\ \tilde{P}_k \end{bmatrix} \tilde{T}_k + t_{k+1,k} \begin{bmatrix} \tilde{q}_{k+1} \\ \tilde{p}_{k+1} \end{bmatrix} e_k^T, \quad (2.15)$$

其中 $k \leq m-l$, 得到以 $\{\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_k\}$ 为标准正交基的 k 维重启子空间 $G_k(A, B; \tilde{q}_1, \tilde{p}_1)$.

隐式重新启动的 GSOAR 算法对于重启子空间的维数 k 有一定的要求, 仅当 $k \leq m-l$ 时, 可以将 (2.14) 直接截断成为新的 k 步 GSOAR 分解, 式子右半部分尾项能约化为只带有 e_k^T 的形式, 新得到的 (2.15) 具有与 (2.3) 相同的格式. 可以继续将新的 k 步 GSOAR 分解 (2.15) 运用算法 1 重新扩展为新的 m 步 GSOAR 分解. 计算新的 Ritz 对以及 Ritz 对的误差, 若未得到满足收敛条件的 Ritz 对, 重复以上步骤, 直到得到满足收敛条件的近似特征对.

(2.15) 从数学上等价于一个 k 步 GSOAR 分解. 如果在 $m_1, m_2, \dots, m_j \leq k$ 步发生收缩, 则 \tilde{Q}_k 中相应的 \tilde{q}_{m_j} 为零向量. Jia 和 Sun^[15] 给出了一种可克服该问题的方法, 使得隐式重启可无条件进行.

2.2 准确位移与精化位移

根据文献 [9] 中对隐式位移的分析可知, 使用一次隐式位移 μ 等价于对重启后的初始向量作了一次如下的更新:

$$\begin{bmatrix} q_1 \\ p_1 \end{bmatrix} = (H - \mu I) \begin{bmatrix} q_1 \\ p_1 \end{bmatrix}.$$

若 μ 和 u 为 H 的一对在求解过程中得到的不需要的特征对, 则在使用 μ 作为隐式位移后, 重启的初始向量中将不含有关于 u 的信息. 设 $\{\lambda, \mu_1, \dots, \mu_{n-1}\}$ 为 H 的全部特征值, 且 $\lambda \neq \mu_i, i = 1, \dots, n-1$. 设 x 为 λ 对应的特征向量, v 是一个与 x 不正交的任意向量. 根据隐式位移的性质, 若将全部 μ_i 用作位移, 则可由 v 得到 x . 在实际计算中显然不可能得到这些位移, 但是这个原理说明在重新启动的过程中选取的位移越好, 使用的位移越多, 重新启动的效果也就越好.

在使用二次特征值问题的显式投影过程中, 我们会遇到一个特征向量 u_1 对应两个特征值 λ 和 μ 的情形. 我们在重新启动时会面临这样的隐患, 若 λ 是我们想求的特征值, μ 是不需要的特征值, 此时,

如果我们使用 μ 作为位移, 则会将我们想求的特征向量信息删除, 因此在选择位移时必须避免这种情形发生. 下面介绍如何选取 GSOAR 方法和 RGSOAR 方法中的位移. 准确位移和精化位移的定义可参见文献 [15]. 首先以精化位移为例, 介绍计算精化位移的方法.

通过 RGSOAR 方法计算得到的精化 Ritz 向量 \tilde{u}_i 比 Ritz 向量 y_i 更加准确. 对于求解线性特征值问题的精化 Arnoldi 方法和精化调和 Arnoldi 方法, 均有相应的精化位移 (参见文献 [5, 16]). 同理, 对于 RGSOAR 方法, 也存在其相应的精化位移.

令 $\text{span}\{\hat{u}_1, \dots, \hat{u}_l\}$ 表示子空间 $\mathcal{G}_k(A, B; q_1, p_1)$ 中关于精化 Ritz 向量 \tilde{u}_i ($i = 1, 2, \dots, k$) 的正交补空间. 将 QEP (1.1) 投影至空间 $\text{span}\{\hat{u}_1, \dots, \hat{u}_l\}$ 上, 从而得到一个 $l = m - k$ 维的 QEP, 该 QEP 具有 $2l$ 个特征值. 将此 $2l$ 个特征值作为不需要的近似特征值, 即作为精化候选位移. 根据前面的分析知, 如果使用多于 l 个的位移, (2.14) 无法将前 k 列截断为 (2.15). 故需要从 $2l$ 个精化候选位移中选取 l 个值作为重新启动时使用的位移. 一个合理的选择是从 $2l$ 个候选位移中选取 l 个离目标点 σ 最远的值作为精化位移. 对于 GSOAR 方法, 类似地可以用 k 个 Ritz 向量对应的正交补空间来求得 $2(m - k)$ 个候选位移, 并使用相同的策略来选择位移.

在两种方法中, 精化向量 $\tilde{u}_1, \dots, \tilde{u}_k$ 和 Ritz 向量 y_1, \dots, y_k 均为想求的特征向量 x_1, \dots, x_k 的近似向量. 故准确候选位移和精化候选位移均为 QEP (1.1) 部分不需要特征值的近似值. 采用 RGSOAR 方法和 GSOAR 方法对 (2.3) 进行重新启动, 这两种方法均是通过位移移除不必要信息的方式来达到优化重启子空间的目的. 如前所述, 若从候选位移中选择了两个对应同一个特征向量的位移, 则至少有一个需要移除的特征向量无法从子空间 $\mathcal{G}_m(A, B; q_1, p_1)$ 中移除. 此时重启效果无法保证, 为了达到重新启动的效果, 应该使用全部候选位移来避免这种情形. 而对于 Arnoldi 分解, 在隐式重新启动中使用过多的位移必然会破坏其结构, 使得算法无法重启, 下面给出这个问题的解决方案.

3 隐式重新启动的 GSOAR 与 RGSOAR 方法的位移策略

本文重点研究当 $l > m - k$ 时对 GSOAR 算法进行隐式重新启动的解决方案, 即提出可使用全部位移信息使 GSOAR 算法可隐式重新启动的方案. 在隐式重新启动中, 位移个数与 (2.14) 中 $b_m^T = (0, \dots, 0, b_{m-l}, \dots, b_{m-1}, b_m)$ 非零元素个数的关系如图 1 所示. 若位移的个数 $l > m - k$, 取 (2.14) 的前 k 列将不会是一个 k 维的 GSOAR 分解, 所以无法直接进行隐式重启.

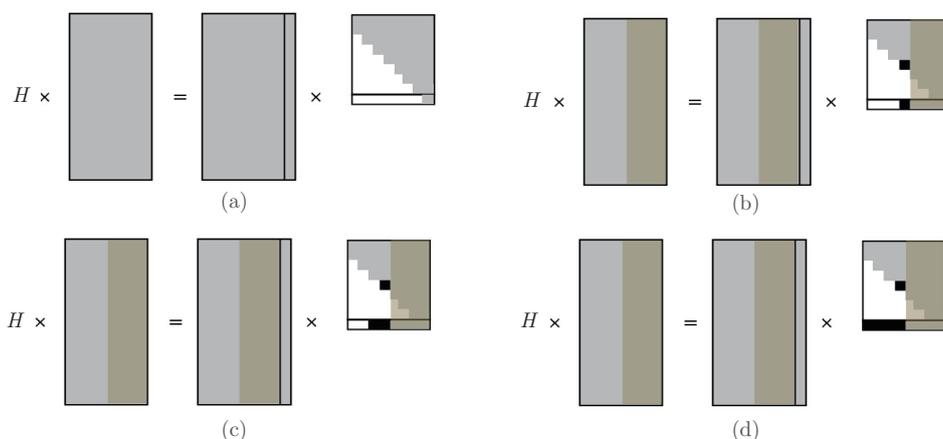


图 1 隐式重新启动 GSOAR 算法示意图. (a) GSOAR 分解; (b) $l = m - k$; (c) $l > m - k$; (d) $l > m$

从文献 [17] 中可知, Krylov 分解与 Arnoldi 分解是等价的. 将图 1(d) 看作广义 Krylov 分解, 则通过正交变换, 可以将它转换为 GSOAR 分解. 对 (2.14), 可以构造一个正交矩阵 W , 使得 b_m^T 变为 αe_m^T (α 为非零常数), 并且使 $W^T \tilde{T}_m W$ 保持上 Hessenberg 矩阵形式.

首先通过 Householder 变换将 b_m 变为 e_m 形式, 求解 Householder 变换矩阵 W_1 使其满足

$$W_1 b_m = \alpha e_m, \quad (3.1)$$

其中 α 为非零常数. 由于 Householder 变换阵具有非奇异和对称正交等性质, 故 $W_1 = W_1^T$, 所以有

$$b_m^T W_1 = \alpha e_m^T \quad (3.2)$$

成立. 将矩阵 W_1 右乘 (2.14) 可得

$$H \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1 = \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1 (W_1^T \tilde{T}_m W_1) + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} b_m^T W_1. \quad (3.3)$$

将 $\begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1$ 视为新的矩阵 $\begin{bmatrix} Q_m \\ P_m \end{bmatrix}$, 将 $W_1^T \tilde{T}_m W_1$ 记为 B_m , 此时的矩阵 B_m 已失去上 Hessenberg 矩阵形式. 综合 (3.2) 和 (3.3), 得到

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} B_m + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T, \quad (3.4)$$

其中 $b_{m+1,m} = t_{m+1,m} \alpha$. 此时, 只需将矩阵 B_m 重新变为上 Hessenberg 矩阵形式, 并且不影响到尾项 e_m^T , 即可以进行隐式重新启动.

仍然采用 Householder 变换将矩阵 B_m 转化为上 Hessenberg 矩阵, 为了保证下一步的变换不对上一步骤的结果产生影响, 从矩阵 B_m 底部开始逐行往上分别对每一行进行 Householder 变换.

当 $i = 1$ 时, 将矩阵 B_m 写作矩阵块形式:

$$B_m = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ c_1 & b_{mm} \end{bmatrix}, \quad (3.5)$$

其中 $c_1 = (b_{m,1}, \dots, b_{m,m-1})$, $B_{11} = B_m(1:m-1, 1:m-1)$ 是一个 $(m-1) \times (m-1)$ 大小的矩阵, $B_{12} = B_m(1:m-1, m)$ 是长度为 $m-1$ 的列向量.

构造 $m-1$ 维 Householder 变换阵 R_1 , 满足

$$c_1 R_1 = \alpha_1 e_{m-1}^T, \quad (3.6)$$

其中 α_1 是非零常数. 令 $U_1 = \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix}$, 对矩阵 B_m 进行正交变换:

$$B_m^{(1)} = U_1^T B_m U_1 = U_1^T \begin{bmatrix} B_{11} & B_{12} \\ c_1 & b_{mm} \end{bmatrix} U_1 = \left[\begin{array}{c|c} B_{11}^{(1)} & B_{12}^{(1)} \\ \hline c_2 & \\ \hline 0 & \cdots & 0 & B_{22}^{(1)} \end{array} \right], \quad (3.7)$$

其中 $B_{11}^{(1)}$ 是大小为 $(m-2) \times (m-2)$ 的矩阵, $B_{12}^{(1)}$ 是大小为 $(m-2) \times 2$ 的矩阵, $B_{22}^{(1)}$ 为 2×2 矩阵, $c_2 = (b_{m-1,1}, \dots, b_{m-1,m-2})$.

当 $i = k$ 时, 将矩阵 $B_m^{(k-1)}$ 写作矩阵块形式:

$$B_m^{(k-1)} = \left[\begin{array}{ccc|c} B_{11}^{(k-1)} & & & B_{12}^{(k-1)} \\ \hline & c_k & & \\ 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & B_{22}^{(k-1)} \\ 0 & \cdots & 0 & \end{array} \right], \quad (3.8)$$

其中 $B_{11}^{(k-1)}$ 为 $(m-k) \times (m-k)$ 矩阵, $B_{12}^{(k-1)}$ 为 $(m-k) \times k$ 矩阵, $B_{22}^{(k-1)}$ 为 $k \times k$ 上 Hessenberg 矩阵, $c_k = (b_{m-k+1,1}, \dots, b_{m-k+1,m-k})$. 构造大小为 $(m-k) \times (m-k)$ 的 Householder 矩阵 R_k , 使之满足

$$c_k R_k = \alpha_k e_{m-k}^T, \quad (3.9)$$

其中 α_k 为非零常数. 令

$$U_k = \begin{bmatrix} R_k & 0 \\ 0 & I_{k \times k} \end{bmatrix},$$

得到

$$B_m^{(k)} = U_k^T B_m^{(k-1)} U_k = \left[\begin{array}{ccc|c} \tilde{B}_{11}^{(k-1)} & & & \tilde{B}_{12}^{(k-1)} \\ \hline 0 & \cdots & \alpha_k & \\ \vdots & \ddots & \vdots & B_{22}^{(k-1)} \\ 0 & \cdots & 0 & \end{array} \right] = \left[\begin{array}{ccc|c} B_{11}^{(k)} & & & B_{12}^{(k)} \\ \hline & c_{k+1} & & \\ 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & B_{22}^{(k)} \\ 0 & \cdots & 0 & \end{array} \right], \quad (3.10)$$

其中 $B_{11}^{(k)}$ 为 $(m-k-1) \times (m-k-1)$ 矩阵, $B_{12}^{(k)}$ 为 $(m-k-1) \times (k+1)$ 矩阵, $B_{22}^{(k)}$ 为 $(k+1) \times (k+1)$ 上 Hessenberg 矩阵, $c_k = (b_{m-k,1}, \dots, b_{m-k,m-k-1})$.

显然, 当 $i = m-2$ 时, $B_m^{(m-2)}$ 是上 Hessenberg 矩阵.

令 $U = U_1 U_2 \cdots U_{m-2}$, 由于矩阵 U_1, U_2, \dots, U_{m-2} 均为正交矩阵, 故矩阵 U 也为正交矩阵. 将正交矩阵 U 右乘 (3.4) 可得

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} U = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} U (U^T B_m U) + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T U. \quad (3.11)$$

由于矩阵 U_1, U_2, \dots, U_{m-2} 的特殊形式, 矩阵 U 的最后一行只有最后一个元素为 1, 其他元素均为 0, 即矩阵 U 最后一行为 e_m^T , 所以有

$$e_m^T U = e_m^T. \quad (3.12)$$

令

$$\begin{bmatrix} Q_m \\ P_m \end{bmatrix} U = \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix},$$

$U^T B_m U = T'_m$, T'_m 为上 Hessenberg 矩阵, 根据 (2.3)、(3.11) 和 (3.12), 得到

$$H \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} = \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} T'_m + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T. \quad (3.13)$$

对比 (3.13) 与 (2.3), 我们找到正交矩阵 $W = W_1 U$ 使得 (2.14) 恢复为 GSOAR 分解形式.

将如上过程可归纳为算法 2.

算法 2 正交变换复原 GSOAR 算法:

Input: $\tilde{Q}_m, \tilde{P}_m, \tilde{T}_m, t_{m+1,m}, b_m^T$ in (2.14)

W_1 : $b_m^T W_1 = \alpha e_m^T$

$$\begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1$$

$$B_m = W_1^T \tilde{T}_m W_1$$

$$b_{m+1,m} = t_{m+1,m} \alpha$$

$$U = I$$

for $i = 1, 2, \dots, m-2$

$$c_i = (b_{m-i+1,1}, \dots, b_{m-i+1,m-i})$$

$$R_i$$
: $c_i R_i = \alpha_i e_{m-i}^T$

$$U_i = \begin{bmatrix} R_i & 0 \\ 0 & I_{i \times i} \end{bmatrix}$$

$$B_m = U_i^T B_m U_i$$

$$U = U U_i$$

end for

$$\begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} U$$

$$T'_m = B_m$$

$$W = W_1 U$$

Output: $Q'_m, P'_m, T'_m, b_{m+1,m}$ in (3.13), W

取 (3.13) 的前 k 列, 即可得到一个新的 k 步 GSOAR 分解

$$H \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} = \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} T'_k + t'_{k+1,k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^T, \quad (3.14)$$

并且得到以 $\{q'_1, q'_2, \dots, q'_k\}$ 为标准正交基的 k 维重启子空间.

在算法 2 中, 重启子空间维数理论上为 k , 但在实际数值计算中, 为了保护 k 个所需的特征值, 通常将重启子空间的维数设为 $k+a$, 其中 a 为一个很小的非负整数, 一般将 a 取为 3 (参见文献 [9]). 算法重启一次需要多进行 $m-1$ 次的 Householder 变换以及 $3m-6$ 次的矩阵乘法运算, 随着运算的进行, 进行变换的向量和矩阵的维数在不断减小, 因此, 运算总次数为 $O(m^3)$.

4 数值实验

本节将给出若干数值算例验证本文算法的有效性, 对带准确位移的隐式重新启动方案 IGSOAR 和带精化位移的隐式重新启动方案 IRGSOAR 使用全部位移信息. 此外, 将与文献 [15] 中算法 (带精化位移的隐式重启方法记为 IRGSOAR0 与带准确位移的隐式重启方法记为 IGSOAR0) 进行对比. 本节所有算例在同一个环境下运行: Win 7, 64 位操作系统, Intel(R) Core(TM) i5-2430M CPU 2.4GHz, RAM 4GB, 使用 MATLAB R2010a.

在下列进行的数值实验中, 投影子空间维数为正整数 m , 要求的特征值个数为 k , 抛弃子空间 (重启时收缩掉的子空间) 的维数为 f , 位移个数为 l ($l = 2f$). 在表格中 tol 代表收敛标准, TOTAL 表示算法的总运行时间, SOAR 表示形成 GSOAR 分解和显式投影所需时间之和, RESTART 表示算法中重启的总时间, FIND 表示求解投影二次特征值问题特征对的时间, 单位均为秒. 图像的横轴代表重新启动的次数, 纵轴代表误差 (相对残量范数).

例 1 本算例为文献 [13, 15] 中的算例, 其系数矩阵为

$$M = -4\pi^2 h^2 I_{q-1} \otimes \left(I_q - \frac{1}{2} e_q e_q^T \right), \quad C = 2\pi i \frac{h}{\xi} I_{q-1} \otimes (e_q e_q^T),$$

$$K = I_{q-1} \otimes D_q + T_{q-1} \otimes \left(-I_q + \frac{1}{2} e_q e_q^T \right),$$

$h = 1/90$ 代表网目尺寸, $q = \frac{1}{h}$, 符号 \otimes 代表 Kronecker 积, ξ 代表阻抗, $D_q = \text{tridiag}(-1, 4, -1) - 2e_q e_q^T$, $T_{q-1} = \text{tridiag}(1, 0, 1)$, 系数矩阵的大小均为 $8,010 \times 8,010$. 使用 12 维子空间, 选取 6 个最接近原点的近似特征值, 取 $f = 5$, 分别运用带全精化位移的隐式重新启动的广义二次 Arnoldi 方法 (IRGSOAR)、带全准确位移的隐式重新启动的广义二次 Arnoldi 方法 (IGSOAR)、隐式重新启动的广义精化二次 Arnoldi 方法 (IRGSOAR0) 和隐式重新启动的广义二次 Arnoldi 方法 (IGSOAR0) 进行计算. 表 1 和图 2 显示了运行结果.

从表 1 和图 2 可以看出, 本文提出的算法与原有算法相比, 总运行时间更短, 具有更快的收敛速度. 新算法只需要 3 次隐式重启就可达到原有 5 次隐式重启的效果. 在 IRGSOAR 和 IGSOAR 方法中, 共调用 114 次 Householder 变换, 用时 0.012 s, 但新算法总节省的运行时间远大于 0.012 s. 对比 IRGSOAR 方法与 IGSOAR 方法, 发现当位移个数相同时, 在总运行时间相差无几的情形下, IRGSOAR 方法的收敛性比 IGSOAR 方法更好.

例 2 (1) 本算例为文献 [15] 中的算例 4, 其系数矩阵分别为

$$M = I, \quad C = \tau \cdot \text{tridiag}(-1, 3, -1), \quad K = \kappa \cdot \text{tridiag}(-1, 3, -1),$$

系数矩阵的大小均为 $n = 5,000$, 并且取 $\tau = 10$, $\kappa = 5$. 选取 6 个最接近目标点 $\sigma = -13 + 0.4i$ 的近似

表 1 例 1 运行结果, tol = 10^{-10}

算法	m	f	重启次数	TOTAL	SOAR	RESTART	FIND
IRGSOAR	12	5	3	0.600	0.357	0.086	0.147
IGSOAR	12	5	3	0.521	0.324	0.076	0.109
IRGSOAR0	12	5	5	1.091	0.561	0.187	0.294
IGSOAR0	12	5	5	1.017	0.571	0.176	0.211

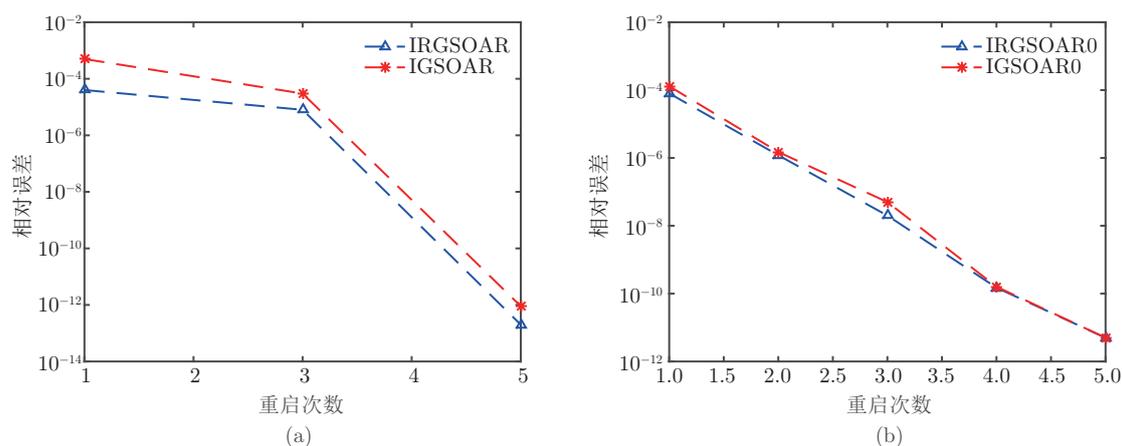


图 2 例 1 运行结果, $m = 12, f = 5$

特征值, 使用 40 维子空间, 取 $f = 28$. 分别运用 IRGSOAR、IGSOAR、IRGSOAR0 和 IGSOAR0 方法进行计算. 表 2 和图 3 显示了运行结果.

(2) 取 $f = 30$, 其他条件保持不变. 分别运用 IRGSOAR、IGSOAR、IRGSOAR0 和 IGSOAR0 方法进行计算. 表 3 和图 4 显示了运行结果.

(3) 将系数矩阵大小扩大至 $n = 20,000$, 使用 50 维子空间, 取 $f = 40$, 其他条件保持不变. 分别运用 IRGSOAR、IGSOAR、IRGSOAR0 和 IGSOAR0 方法进行计算. 表 4 和图 5 显示了运行结果.

表 2 例 2(1) 运行结果, $\text{tol} = 10^{-10}$

算法	m	f	重启次数	TOTAL	SOAR	RESTART	FIND
IRGSOAR	40	28	4	1.803	0.558	0.644	0.581
IGSOAR	40	28	6	2.567	0.700	1.174	0.673
IRGSOAR0	40	28	54	27.390	7.955	10.008	8.590
IGSOAR0	40	28	65	29.024	9.617	12.239	6.173

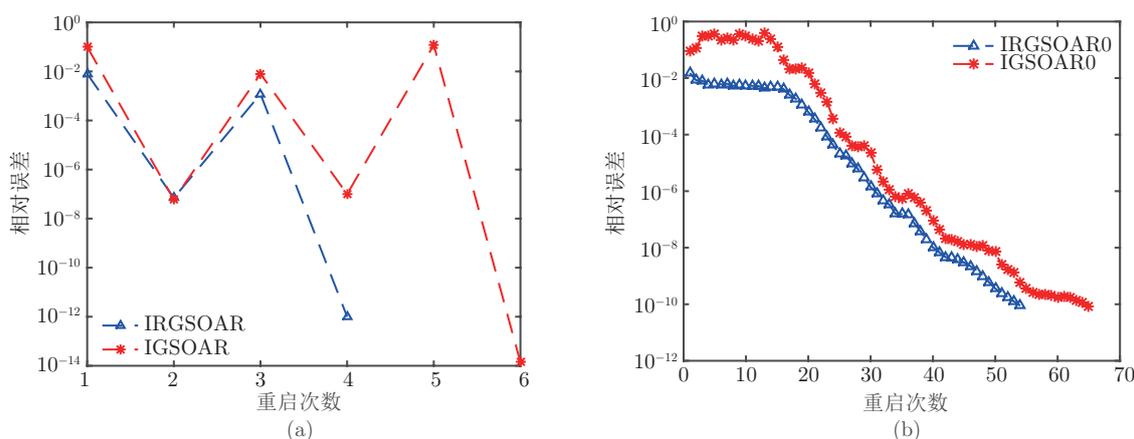


图 3 例 2(1) 运行结果, $m = 40, f = 28$

表 3 例 2(2) 运行结果, $\text{tol} = 10^{-10}$

算法	m	f	重启次数	TOTAL	SOAR	RESTART	FIND
IRGSOAR	40	30	5	2.341	0.616	0.979	0.730
IGSOAR	40	30	6	2.580	0.810	1.177	0.572
IRGSOAR0	40	30	54	26.880	8.154	10.013	7.877
IGSOAR0	40	30	59	25.508	8.514	10.665	5.416

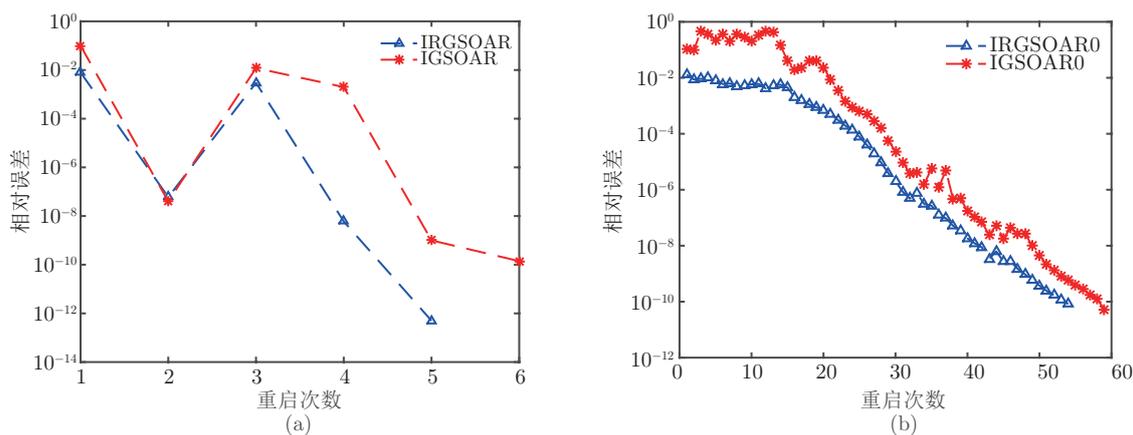


图 4 例 2(2) 运行结果, $m = 40, f = 30$

表 4 例 2(3) 运行结果, $\text{tol} = 10^{-10}$

算法	m	f	重启次数	TOTAL	SOAR	RESTART	FIND
IRGSOAR	50	40	5	37.618	10.174	11.554	15.799
IGSOAR	50	40	6	32.459	8.992	15.298	8.097
IRGSOAR0	50	40	100	143.670	37.808	46.238	56.421
IGSOAR0	50	40	100	122.562	37.421	49.038	32.787

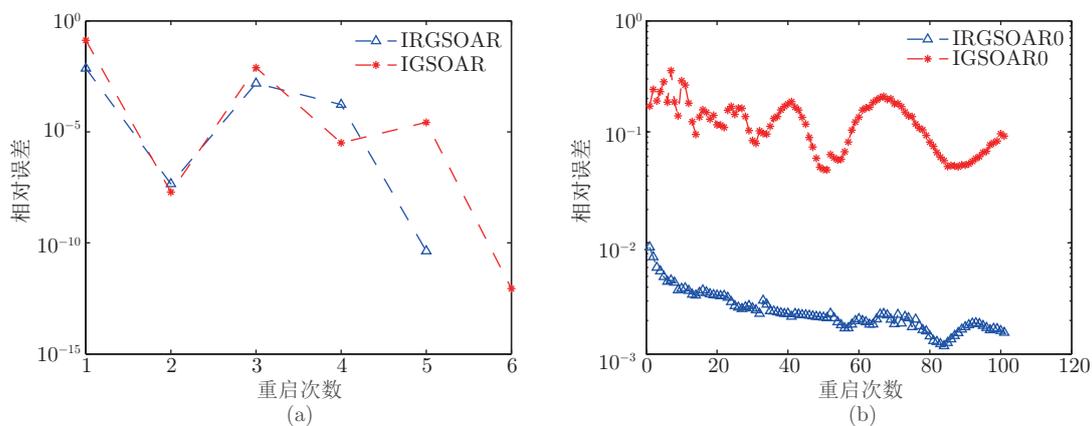


图 5 例 2(3) 运行结果, $m = 50, f = 40$

通过结果可以看出 IRGSOAR 和 IGSOAR 方法的显著优越性. 当 $f = 28$ 时, 原算法中需要 50 多次重启才能达到收敛要求, 而本文的 IRGSOAR 和 IGSOAR 方法分别使用 4 和 6 次隐式重启即可收敛. 在新算法中, 调用 Householder 变换耗时 0.028 s, 而总节省的运行时间约为 26 s. 当 $f = 30$ 时, 原算法中需要 50 多次重启才能达到收敛要求, 而 IRGSOAR 和 IGSOAR 方法分别使用 5 和 6 次隐式重启即可收敛. 虽然在 Householder 变换共耗时 0.033 s, 但是隐式重启部分的时间仍大大缩短. 当系数矩阵的维数为 20,000 时, 原算法重启 100 次也无法达到收敛条件, 而 IRGSOAR 和 IGSOAR 方法分别重启 5 和 6 次即可得到结果. 当位移个数相同时, IRGSOAR 方法的重启次数总比 IGSOAR 方法少. 这说明当位移个数较多时, 带全部精化位移 IRGSOAR 方法其位移所起的效果也更好.

例 3 二次特征值问题的系数矩阵如下:

$$M = I_n, \quad C = \begin{bmatrix} 8 & -4 & 0 & & \\ & 2 & 12 & \ddots & 0 \\ & 0 & \ddots & 12 & -4 \\ & & & 0 & 2 & 8 \end{bmatrix}, \quad K = \begin{bmatrix} 2 & 2 & 0 & & \\ & -1 & 3 & \ddots & 0 \\ & 0 & \ddots & 3 & 2 \\ & & & 0 & -1 & 2 \end{bmatrix},$$

系数矩阵阶为 $n = 5,000$. 选取 6 个最接近目标点 $\sigma = -10 - 0.8i$ 的近似特征值, 使用 26 维子空间, 取 $f = 15$, 分别运用 IRGSOAR、IGSOAR、IRGSOAR0 和 IGSOAR0 方法进行计算. 表 5 和图 6 显示了运行结果.

表 5 例 3 运行结果, $\text{tol} = 10^{-10}$

算法	m	f	重启次数	TOTAL	SOAR	RESTART	FIND
IRGSOAR	26	15	2	1.0140	0.102	0.101	0.391
IGSOAR	26	15	3	0.9360	0.194	0.175	0.129
IRGSOAR0	26	15	100	53.7110	6.523	12.169	8.312
IGSOAR0	26	15	100	47.8455	6.494	12.095	5.497

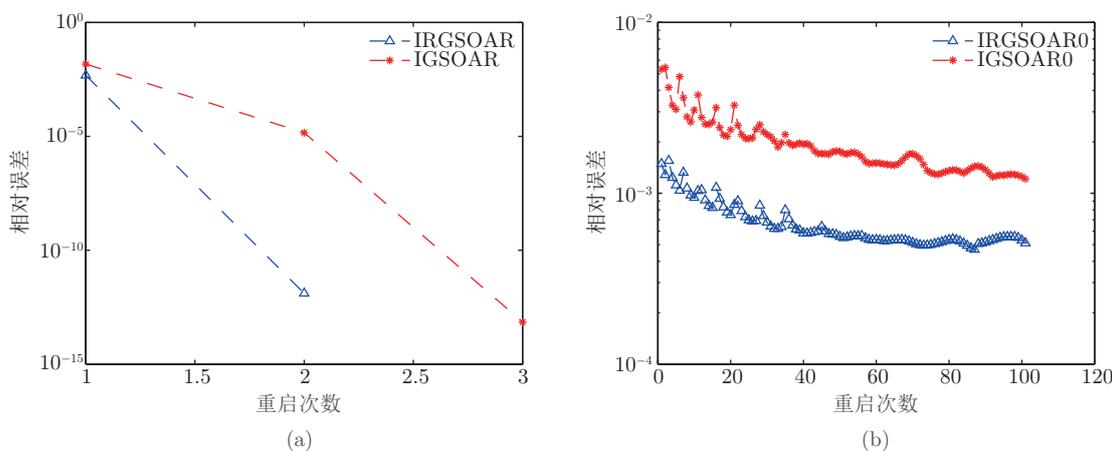


图 6 例 3 运行结果, $m = 26, f = 15$

从结果可看出, 即使经过 100 次的重启, 先前的算法也不能满足收敛条件. 然而, 本文提出的新算法在仅经过 2 或 3 次重启即可得到结果.

与线性特征值问题不同, 二次特征值问题可能会出现候选位移个数大于可使用的位移个数的情形. 为了解决这个问题, 我们提出了一个有效且稳定的可使用全部候选位移信息的算法. 数值算例表明, 新的位移策略提高了每一次重新启动的效率, 降低了重新启动的总次数, 从而提高了算法的整体效率.

致谢 感谢审稿专家的宝贵建议.

参考文献

- 1 Tisseur F, Meerbergen K. The quadratic eigenvalue problem. *SIAM Rev*, 2001, 43: 235–286
- 2 Betcke T, Higham N J, Mehrmann V, et al. NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans Math Software*, 2013, 39: 254–261
- 3 Ruhe A. Algorithms for the nonlinear eigenvalue problem. *SIAM J Numer Anal*, 1973, 10: 674–689
- 4 Bai Z J, Demmel J, Dongarra J, et al. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Philadelphia: SIAM, 2000
- 5 Jia Z X. Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm. *Linear Algebra Appl*, 1999, 287: 191–214
- 6 Higham N J, Li R C, Tisseur F. Backward error of polynomial eigenproblems solved by linearization. *SIAM J Matrix Anal Appl*, 2007, 29: 1218–1241
- 7 Bai Z J, Su Y F. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J Matrix Anal Appl*, 2003, 26: 640–659
- 8 Huang H M, Jia Z X, Lin W W. On the convergence of Ritz pairs and refined Ritz vectors for quadratic eigenvalue problems. *BIT*, 2013, 53: 941–958
- 9 Sorensen D C. Implicit application of polynomial filters in a k -step Arnoldi method. *SIAM J Matrix Anal Appl*, 1992, 13: 357–385
- 10 Meerbergen K. The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J Matrix Anal Appl*, 2008, 30: 1463–1482
- 11 Otto C. Arnoldi and Jacobi-Davidson methods for quadratic eigenvalue problems. PhD Thesis. Berlin: Technische Universität Berlin, 2004
- 12 Zhou L P, Bao L, Lin Y Q, et al. Restarted generalized Krylov subspace methods for solving quadratic eigenvalue problems. *Int J Comput Math Sci*, 2000, 4: 148–155
- 13 Huang W Q, Li T, Li Y T, et al. A semiorthogonal generalized Arnoldi method and its variations for quadratic eigenvalue problems. *Numer Linear Algebra Appl*, 2013, 20: 259–280
- 14 Beeumen R L, Meerbergen K, Michiels W. Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM J Matrix Anal Appl*, 2015, 36: 820–838
- 15 Jia Z X, Sun Y Q. Implicitly restarted generalized second-order Arnoldi type algorithms for the quadratic eigenvalue problem. *Taiwanese J Math*, 2015, 19: 1–30
- 16 Jia Z X. The refined harmonic Arnoldi method and an implicitly restarted refined algorithm for computing interior eigenpairs of large matrices. *Appl Numer Math*, 2002, 42: 489–512
- 17 Stewart G W. A Krylov-Schur algorithm for large eigenproblems. *SIAM J Matrix Anal Appl*, 2002, 23: 601–614

A new shift strategy for the implicitly restarted generalized second-order Arnoldi method

GONG FangHui & SUN YuQuan

Abstract In the implicitly restarted generalized second-order Arnoldi (GSOAR) method, we can get $2m$ approximate eigenpairs by projecting the quadratic eigenvalue problem (QEP) onto an m -dimensional subspace.

During implicitly restarted processes, the problem is the mismatch between the number of shifts and the dimension of the subspace. In order to cure the problem, a new shift strategy for GSOAR method is proposed in this paper. We prove that the novel method can use all $2l$ shift candidates and preserve the special structure. Numerical experiments illustrate that the new method enhances the overall efficiency of the algorithm by increasing the efficiency of every restart process.

Keywords quadratic eigenvalue problem, GSOAR method, implicitly restart, shifts

MSC(2010) 65F15, 65Y20

doi: 10.1360/012016-22