arXiv:2305.02044v1 [math.NA] 3 May 2023

最小二乘与最小范数问题中类CG算法的误差估计

Jan Papež[a] 和 Petr Tichý[b]

[a]捷克科学院数学研究所，布拉格，捷克共和国[b]查理大学数学与物理学院，布拉格，捷克共和国

2023年5月4日版本

摘要

在 [Meurant、Papež 和 Tichý；《数值算法》 88, 2021]，中，我们提出了一种用于估计共轭梯度（CG）法中误差能量范数的自适应估计方法。本文将该估计方法扩展至求解线性逼近问题的算法，这类问题涉及一般的、可能为矩形的矩阵，其求解基于对由原始矩阵构造的正定（半正定）矩阵所形成的系统应用共轭梯度法。我们证明，所得估计方法保留了其关键性质：计算成本极低，并且在满足某些温和假设的情况下，在有限精度算术中具有数值可靠性。我们讨论了基于 Hestenes–Stiefel 类实现（文献中通常称为 CGLS 和 CGNE）以及基于双对角化（LSQR 和 CRAIG）的算法，同时也讨论了非预条件和预条件变体。数值实验验证了该估计方法的稳健性及其令人十分满意的表现。

引言

求解线性逼近问题（使用一般的、可能为矩形的矩阵）是科学计算中的常见任务。本文中，我们考虑一个最小二乘问题

$$\min_{z\in\mathbb{R}^n} \|b - Az\|, \qquad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ m \geq n,$$

以及一个最小范数问题

$$\min_{z\in\mathbb{R}^n} \|z\| \quad \text{subject to} \quad Az = b, \qquad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ b \in \mathcal{R}(A),$$

其中，$\|\cdot\|$ 表示欧几里得范数，$\mathcal{R}(A)$ 是 A 的值域，即由 A 的列张成的空间。

当 A 较大且稀疏时，通常使用迭代方法求得该问题（可能不唯一）解的近似值。使用任何迭代方法时，都需要适当的停止准则，而这通常需要估计某个能够衡量所求近似值质量的量。

Krylov 子空间方法是用于迭代求解最小二乘和最小范数问题的最常用、研究最广泛的方法之一。其数学推导基于在 Krylov 子空间（维数逐步增加）上最小化某个误差或残差范数；例如参见 [21]。通常，存在若干在数学上等价的算法（它们基于在相同子空间上求解同一个最小化问题），但这些算法的实现方式不同，因此在有限精度算术中可能表现出不同的行为。

本文旨在将 [24] 中为共轭梯度（CG）法推导的（一种基于启发式的）自适应误差估计扩展到一类用于求解上述问题的算法。我们考虑了各种算法变体，它们在数学上等价于将共轭梯度法应用于一个具有对称性的系统。

正定（半正定）矩阵 A^TA 或 AA^T，由原始矩阵 A 构造而成。本文讨论的算法具体包括 CGLS、LSQR、CGNE 和 CRAIG（基于双对角化的 Craig 方法）。对于这些算法中的每一种，我们都给出了估计值的推导，并说明该估计值的两个关键性质在共轭梯度法中已知且依然成立。首先，该估计值的计算成本很低。其次，假设有限精度计算过程中保持了局部正交性，且尚未达到最大可达精度，那么即使舍入误差严重影响算法的收敛，该估计值在数值上仍然可靠。收敛可能会显著延迟，但延迟会反映在估计值中；它能估计计算所得近似值的实际误差。关于共轭梯度法局部正交性的舍入误差分析见 [32]；另见 [23]。对本文讨论的算法也可以进行这类分析，但需要补充许多技术细节。最大可达精度的分析，例如见 [16] 和 [31]，其中讨论的背景更为一般。请注意，本文讨论的某些估计值已见于文献，例如 [19, 3, 2, 11],，但这些文献既未使用我们的自适应技术，也未讨论如何在有限精度算术中使用这些估计值。

本文结构安排如下。首先简要回顾 [24] 中的自适应误差估计。接着，在第2节介绍最小二乘问题，以及分别由 [18] 和 [27], 提出的 CGLS 和 LSQR 求解方法。第3节讨论最小范数问题，以及 Craig 方法的误差估计（该方法最初由 [8, 9], 提出，并使用 CGNE 实现（[14, 第 504] 页），以及基于 Golub–Kahan 双对角化的算法 [27]，我们称之为 CRAIG；例如参见 [30]）。然后，在第4节讨论这些算法的预条件变体的误差估计。数值实验结果见第5节，最后给出总结性评论。带误差估计的算法 MATLAB 代码可从 GitHub 仓库 [28] 获取。

1 共轭梯度法中的自适应误差估计

本节简要回顾 [24] 中为共轭梯度法推导的自适应误差估计。

首先回顾使用共轭梯度法求解具有对称正定矩阵 A 的系统 $Ax = b$ 的基本思想。从初始猜测 x_0 及其相应残差 $r_0 = b - Ax_0$ 出发，共轭梯度法生成近似值

$$x_k \in x_0 + \mathcal{K}_k(A, r_0), \qquad \mathcal{K}_k(A, r_0) \equiv \operatorname{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\},$$

使得

$$x_k = \underset{y \in x_0 + \mathcal{K}_k(A,r_0)}{\arg\min} \|x - y\|_A, \tag{1}$$

其中，$\|\cdot\|_A$ 表示 A-范数，$\|v\|_A^2 = v^TAv$。共轭梯度法的标准 Hestenes-Stiefel 实现见算法 1。其中，我们在第 6 行调用函数 adaptive(·)，该函数执行自适应误差估计的计算，具体说明如下。

注意，共轭梯度法也可应用于矩阵 A 仅为对称半正定矩阵的系统 $Ax = b$；参见例如 [20]。此时，$\|\cdot\|_A$ 仅是 $\mathcal{R}$（A）上的范数，因此我们需要一个附加假设，使得（1）有意义。特别地，我们必须假设该系统是相容的，即 $b \in \mathcal{R}$（A），并从 $\mathcal{R}$（A）中选择初始猜测 x_0。这保证了对于 $k = 0$、1、.....，$\mathcal{K}_k$（A，r_0）$\subset \mathcal{R}$（A）且 $x_k \in \mathcal{R}$（A），从而可以在 $\mathcal{R}$（A）上考虑最小化问题（1）。注意，在有限精度算术中，系统矩阵 A 的奇异性可能导致一些数值不稳定性；特别是，误差范数在达到最大可达精度后可能开始增大；参见 [20, 第 4] 节。

[24] 的自适应误差估计基于该公式（我们称之为 Hestenes–Stiefel 公式），

$$\|x - x_\ell\|_A^2 = \sum_{j=\ell}^{k} \Delta_j + \|x - x_{k+1}\|_A^2, \qquad \Delta_j \equiv \gamma_j \|r_j\|^2, \tag{2}$$

算法 1 $Ax = b$ 的共轭梯度法（共轭梯度法）

输入 A, b, x_0
$r_0 = b - Ax_0$, $p_0 = r_0$
$\ell = 0$, $\tau = 0.25$ 指定误差估计的容差
对于 $k = 0$, 1, ...直到收敛**执行**
  $\gamma_k = \|r_k\|^2 / \|p_k\|_A^2 = r_k^T r_k \,/\, p_k^T A p_k$
  自适应$(k, \ell, \Delta_k = \gamma_k \|r_k\|^2, \tau)$
  $x_{k+1} = x_k + \gamma_k p_k$
  $r_{k+1} = r_k - \gamma_k A p_k$
  $\delta_{k+1} = \|r_{k+1}\|^2 / \|r_k\|^2$
  $p_{k+1} = r_{k+1} + \delta_{k+1} p_k$
结束循环

对于 $\ell \le k$，以及一个自适应地求得 $\ell = \ell(k)$ 的过程；给定 k，使得

$$\frac{\|x - x_\ell\|_A^2 - \sum_{j=\ell}^{k} \Delta_j}{\|x - x_\ell\|_A^2} \le \tau$$

对于某个预设的容差 $\tau \in (0, 1)$（通常设为 0.25）。然后，误差 $\|x - x_k\|_A^2$ 估计为

$$\Delta_{\ell:k} \equiv \sum_{j=\ell}^{k} \Delta_j \approx \|x - x_\ell\|_A^2.$$

由（2）可知， $\Delta_{\ell:k}$ 表示 $\|x - x_\ell\|_A^2$的下界。界 $\Delta_{\ell:k}$ 的计算成本非常低，数值稳定 [32, 33], ，并且如果正确设置 ℓ ，它可以提供足够准确的估计。

算法 2 自适应() — 延迟 $k - \ell$ 的自适应选择

输入 k, ℓ, Δ_k, τ2: 容差 $= 10^{-4}$ 设置一个用于误差估计的参数
输出 ℓ4: 将 m 设为满足以下条件的最大索引 j, $0 \le j < k$, 使得

$$\frac{\Delta_{\ell:k}}{\Delta_{j:k}} \le \mathrm{TOL}$$

  如果不存在这样的 m , 设定 $m = 0$
确定 S 为

$$S = \max_{m \le j < k} \Delta_{j:k} / \Delta_j$$

当 $S\Delta_k / \Delta_{\ell:k-1} \le \tau$ **且** $\ell < k$ **时，执行**
  接受 $\Delta_{\ell:k}$ 作为第 ℓ次迭代时误差的估计值
  $\ell = \ell + 1$
结束循环

用于设置（自适应）延迟 $k - \ell$ 并计算误差估计的函数 adaptive(·) 在算法 2 中给出。正如可以观察到的，在计算近似值 x_{k+1} 的第 k 次共轭梯度迭代中，会对之前的某个近似值 x_ℓ 进行误差估计。有关详细描述、推导和原理，请参见 [24]。请注意，在原始论文中，算法 2 第 7 行使用了 [24]、 $\Delta_{\ell:k-1}$ 来进行误差估计。由于 $\Delta_{\ell:k}$ 可用且 $\Delta_{\ell:k-1} < \Delta_{\ell:k} \le \|x - x_\ell\|_A^2$，我们建议使用更紧的估计值。显然，可以轻松地将此函数添加到现有的共轭梯度代码中。

如 [24, 第 4], 节所述，自适应误差估计也可以扩展到预条件共轭梯度法，同时保留所有关键性质。为简洁起见，此处不介绍所得算法，请参阅 [24, 第 4] 节。

2 最小二乘问题中的误差估计

考虑一个最小二乘问题

$$x = \arg\min_{z \in \mathbb{R}^n} \|b - Az\| \tag{3}$$

其中， $A \in \mathbb{R}^{m \times n}$、 $m \geq n$ 是给定矩阵， $b \in \mathbb{R}^m$ 是右端项。众所周知， x 是最小二乘解，当且仅当 $b - Ax \perp \mathcal{R}(A)$；等价地，当且仅当 x 是以下正规方程组的解

$$A^T A x = A^T b. \tag{4}$$

如果 rank(A) $= n$，那么矩阵 $A^T A$ 非奇异，且方程组 (3) 的解唯一。此外， $A^T A$ 对称正定，因此可以将共轭梯度法应用于 (4)。

从初始猜测 x_0 出发，将共轭梯度法应用于 (4)，并令 $r_0 = b - Ax_0$，则在第 k 次迭代中计算得到的近似值 x_k 满足

$$x_k \in x_0 + \mathcal{K}_k(A^T A, A^T r_0),$$

and

$$x_k = \arg\min_{y \in x_0 + \mathcal{K}_k(A^T A, A^T r_0)} \|x - y\|_{A^T A}. \tag{5}$$

共轭梯度法应用于 (4) 有多种实现方式。有关综述，请参阅 [5] 或一系列论文 [11, 12, 13]。

算法 3 CGLS

1. **输入** A， b， x_0
2. $r_0 = b - Ax_0$
3. $s_0 = p_0 = A^T r_0$
4. **对于** $k = 0, 1, \ldots$ **执行**
5. $q_k = Ap_k$
6. $\gamma_k = \|s_k\|^2 / \|q_k\|^2$
7. $x_{k+1} = x_k + \gamma_k p_k$
8. $r_{k+1} = r_k - \gamma_k q_k$
9. $s_{k+1} = A^T r_{k+1}$
10. $\delta_{k+1} = \|s_{k+1}\|^2 / \|s_k\|^2$
11. $p_{k+1} = s_{k+1} + \delta_{k+1} p_k$
12. **结束对于**

CGLS 算法（算法 3）已在 CG 的开创性论文 [18, 第 424] 页中进行了描述。它对应于将 CG 算法应用于正规方程组，同时进行轻微的代数重排，以避免出现形如 $A^T Ap$ 的向量。由于这种重排，CGLS 的数值性质优于直接应用于（4）的 CG（算法 1）。CGLS 算法可见于例如 [27,第 7 节1] 以及 [5, 算法 31]。基于 Golub–Kahan 双对角化、与 CGLS 在数学上等价的一种算法已在 [27] 中推导出来。该算法称为 LSQR，见算法 4。

由于共轭梯度法也可应用于奇异系统（见第 1 节中的相应讨论），因此，如果秩(A) $< n$，CGLS 和 LSQR 算法也可以应用。然而，在这种情况下，(3) 的最小二乘解并不唯一。

算法 4LSQR

输入 A, b
$\beta_1 u_1 = b$ β_i是归一化系数，使其具有 $\|u_i\| = 1$、 $i = 1$、2、.....
$\alpha_1 v_1 = A^T u_1$ α_i是归一化系数，使其具有 $\|v_i\| = 1$、 $i = 1$、2、.....
$w_1 = v_1$
$x_0 = 0$
$\bar{\phi}_1 = \beta_1$
$\bar{\rho}_1 = \alpha_1$
</style>**对** $k = 1, 2, \ldots$ **执行**
    $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$
    $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$
    $\rho_k = (\bar{\rho}_k^2 + \beta_{k+1}^2)^{1/2}$
    $c_k = \bar{\rho}_k / \rho_k$
    $s_k = \beta_{k+1} / \rho_k$
    $\theta_{k+1} = s_k \alpha_{k+1}$
    $\bar{\rho}_{k+1} = -c_k \alpha_{k+1}$
    $\phi_k = c_k \bar{\phi}_k$
    $\bar{\phi}_{k+1} = s_k \bar{\phi}_k$
    $x_k = x_{k-1} + (\phi_k / \rho_k) w_k$
    $w_{k+1} = v_{k+1} + (\theta_{k+1} / \rho_k) w_k$
结束循环

2.1 CGLS 和 LSQR 中的误差估计

令 $b_{|\mathcal{R}(A)}$ 为 b 在 A 的值域上的正交投影，并记为

$$r = b - b_{|\mathcal{R}(A)}.$$

本节重点讨论误差的估计

$$\|x - x_k\|_{A^T A}^2 = \|A(x - x_k)\|^2 = \|b - r - Ax_k\|^2 = \|r_k - r\|^2 = \|r_k\|^2 - \|r\|^2,$$

其中我们设 $r_k = b - Ax_k$，并利用了 $r \perp r - r_k$ 这一事实。因此，如果 $b \in \mathcal{R}$（A），也就是说 $r = 0$，则误差 $\|x - x_k\|_{A^T A}^2$ 等于可完全计算的量 $\|r_k\|^2 = \|b - Ax_k\|^2$，无需使用估计器。下文我们因此假设 $b \notin \mathcal{R}$（A）。此时范数 $\|r\|$ 非零且未知，仅凭残差范数 $\|r_k\|$ 可能不足以确定合适的停止准则。例如，[19] 中就讨论过这一问题。根据残差 r 和迭代残差 r_k 的范数进行比较，并据此设定停止准则，似乎是很自然的做法。不过，这一讨论超出了本文的范围。

误差范数 $\|x - x_k\|_{A^T A}$ 是 CGLS 和 LSQR 算法所关注的量，因为它在迭代过程中会被最小化（因此是单调的）；而且，正如我们将在第4节中看到的，采用预条件时，该范数也会被最小化。在许多实际情况下，用户还希望估计误差的欧几里得范数 $\|x - x_k\|$。在不采用预条件时，可以基于误差的欧几里得范数与能量范数之间的直接关系，在共轭梯度法中高效地估计该误差范数，参见例如 [18] 或 [32], ；也可以基于共轭梯度法与高斯求积之间的关系进行估计 [15, 22]。随后，可以开发类似的技术来提高估计精度，这些技术使用式（2）的一个类似形式。所有这些结果都可以从共轭梯度法推广到 CGLS 和 LSQR，但同样仅适用于不采用预条件的情况。采用预条件时，通常不存在误差的欧几里得范数与能量范数之间的直接关系。不过，可以利用.....来界定误差的欧几里得范数。

$$\|x - x_k\| \leq \frac{1}{\sigma} \|x - x_k\|_{A^T A}$$

这需要预先知道 A 的最小奇异值的一个下界 σ；参见例如 [2, 13, 17]。显然，上述上界未必能很好地估计 $\|x - x_k\|$，但一般而言，我们手头并没有更好的方法。请注意，要使用上述上界，我们仍然需要估计量 $\|x - x_k\|_{A^TA}$，而这正是本文的研究目标。

2.2 CGLS 中的误差估计

要推导出类似于 [24] 的估计，第一步是找出一个与 (2) 类似的表达式。通过简单的代数运算，见 [33,第 798] 页，可得

$$\|x - x_k\|^2_{A^TA} - \|x - x_{k+1}\|^2_{A^TA} = \|x_{k+1} - x_k\|^2_{A^TA} + 2\left(x_{k+1} - x_k\right)^T A^T A\left(x - x_{k+1}\right). \tag{6}$$

下文中，我们只使用在有限精度算术计算过程中也成立（允许存在微小误差）的关系。为简化术语，我们称这些恒等式在数值上成立。特别地，我们避免使用向量的全局正交性，因为它通常会很快丧失。由于在 CGLS 中

$$x_{k+1} - x_k = \gamma_k p_k, \qquad s_{k+1} = A^T A(x - x_{k+1}) \tag{7}$$

可得

$$\|x - x_k\|^2_{A^TA} - \|x - x_{k+1}\|^2_{A^TA} = \gamma_k^2\|p_k\|^2_{A^TA} + 2\gamma_k p_k^T s_{k+1}.$$

利用计算 γ_k 的公式（第 6 行）以及 s_{k+1} 与 p_k 之间的（局部）正交性，

$$\gamma_k = \frac{\|s_k\|^2}{\|p_k\|^2_{A^TA}}, \quad p_k^T s_{k+1} = 0, \tag{8}$$

我们得到一个类似于 (2) 的结果，

$$\|x - x_\ell\|^2_{A^TA} = \sum_{j=\ell}^{k} \gamma_j\|s_j\|^2 + \|x - x_{k+1}\|^2_{A^TA}$$

以及误差估计器

$$\Delta^{\mathrm{CGLS}}_{\ell:k} \equiv \sum_{j=\ell}^{k} \gamma_j\|s_j\|^2 \approx \|x - x_\ell\|^2_{A^TA} = \|r_\ell\|^2 - \|r\|^2.$$

为了推导式 (2) 的类似形式，我们使用了关系式 (7) 和 (8)。这些关系在数值上确实成立，直至达到最大可达精度；参见例如 [32]。

2.3 LSQR 中的误差估计

对于 LSQR，情况则更为微妙。这里的解释需要更多篇幅，并与推导过程紧密结合；该推导受到 [32] 中思想的启发。

这些向量 u_i 和 v_i 是通过 Golub–Kahan 双对角化在 LSQR（算法 4）中生成的，它们满足 fy

$$\begin{aligned} AV_k &= U_{k+1}B_k \\ A^TU_{k+1} &= V_kB_k^T + \alpha_{k+1}v_{k+1}e_{k+1}^T \end{aligned}$$

其中 $V_k = [v_1, v_2, \ldots, v_k]$，$U_{k+1} = [u_1, u_2, \ldots, u_{k+1}]$，以及

$$B_k = \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \ddots & & & \\ & \ddots & \ddots & & \\ & & \beta_{k-1} & \alpha_{k-1} & \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix}.$$

经过简单的代数运算，我们得到

$$A^T A V_k = V_k T_k + \alpha_{k+1}\beta_{k+1} v_{k+1} e_{k+1}^T, \qquad T_k = B_k^T B_k.$$

s因此， v_i 可视为由相应的三项递推生成的 Lanczos 向量，用于 $A^T A$。假设 $x_0 = 0$，LSQR 近似值为

$$x_k = V_k y_k, \quad y_k = \arg\min \|\beta_1 e_1 - B_k y\| \tag{9}$$

以及相应的 最小二乘问题使用 B_k 的 QR 分解求解。特别是 r,

$$Q_k \begin{bmatrix} B_k & \beta_1 e_1 \end{bmatrix} = \begin{bmatrix} R_k & f_k \\ & \tilde{\phi}_{k+1} \end{bmatrix} = \begin{bmatrix} \rho_1 & \theta_2 & & & & \phi_1 \\ & \rho_2 & \theta_3 & & & \phi_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \rho_{k-1} & \theta_k & \phi_{k-1} \\ & & & & \rho_k & \phi_k \\ & & & & & \tilde{\phi}_{k+1} \end{bmatrix},$$

并且 y_k 是 $R_k y = f_k$ 的解。因此， $T_k = B_k^T B_k = R_k^T R_k$，所以 $R_k^T R_k$ 表示 T_k 的 Cholesky 分解。由于 CGLS（将共轭梯度法应用于 $A^T A x = A^T b$）也会（隐式地）计算 T_k 的 Cholesky 分解，参见例如 [25], ，我们得到两个算法中出现的系数之间的关系：

$$\begin{bmatrix} \frac{1}{\sqrt{\gamma_0}} & & & \\ \sqrt{\frac{\delta_1}{\gamma_0}} & \ddots & & \\ & \ddots & \ddots & \\ & & \sqrt{\frac{\delta_{k-1}}{\gamma_{k-2}}} & \frac{1}{\sqrt{\gamma_{k-1}}} \end{bmatrix} = \begin{bmatrix} \rho_1 & & & \\ \theta_2 & \ddots & & \\ & \ddots & & \\ & \theta_{k-1} & \rho_{k-1} & \\ & & \theta_k & \rho_k \end{bmatrix} \tag{10}$$

其中， γ_i 和 δ_i 是 CGLS 系数；参见算法 3。

众所周知，在有限精度计算中， V_k 和 U_k 的列向量之间的全局 正交性通常会很快丧失。换言之，不能指望 $V_k^T V_k = I = U_k^T U_k$。我们的目的是说明，尽管全局正交性丧失，并且在稍后将具体说明的一些自然假设下，LSQR 的 Hestenes–Stiefel 公式具有如下形式：

$$\|x - x_{k-1}\|_{A^T A}^2 - \|x - x_k\|_{A^T A}^2 = \phi_k^2, \tag{11}$$

在数值上仍然成立。请注意，我们不能直接应用 [32, 33] 的结果，因为 LSQR（算法 4）使用的递推关系与共轭梯度法（算法 1）不同。我们不在此像 [32, 33] 中那样，对公式（11）进行详细的舍入误差分析。相反，我们侧重于基于公式（11）的便捷推导，以及关于如何保持局部正交性的讨论来进行论证。我们从以下引理开始；另见 [27, 第 52] 页。

引理1。 对于 *LSQR*（算法 *4*）生成的量，有

$$A^T(b - Ax_k) \quad = \quad -\theta_{k+1}\phi_k v_{k+1}. \tag{12}$$

证明。注意， y_k 是最小二乘问题 (9) 的解，因此 $B_k^T B_k y_k = B_k^T \beta_1 e_1$。由于$x_k = V_k y_k$ 和 $AV_k = U_{k+1} B_k$，我们得到

$$\begin{aligned} A^T(b - Ax_k) &= A^T U_{k+1}(\beta_1 e_1 - B_k y_k) \\ &= \left(V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T\right)(\beta_1 e_1 - B_k y_k) \\ &= V_k\left(B_k^T \beta_1 e_1 - B_k^T B_k y_k\right) - \alpha_{k+1} v_{k+1} e_{k+1}^T B_k y_k \\ &= -\alpha_{k+1} v_{k+1}\left[e_{k+1}^T B_k y_k\right] \\ &= \alpha_{k+1} v_{k+1} e_{k+1}^T (\beta_1 e_1 - B_k y_k), \end{aligned}$$

其中我们使用了 $e_{k+1}^T e_1 = 0$。利用 QR 分解

$$\begin{bmatrix} B_k & \beta_1 e_1 \end{bmatrix} = Q_k^T \begin{bmatrix} R_k & f_k \\ & \bar{\phi}_{k+1} \end{bmatrix}$$

并且 $R_k y_k = f_k$ 我们得到

$$\beta_1 e_1 - B_k y_k = Q_k^T \begin{bmatrix} 0 \\ \bar{\phi}_{k+1} \end{bmatrix};$$

¯参见 [27, 公式 (4.8)]。因此， $\beta_1 e_1 - B_k y_k$ 的最后一个分量为 $-c_k \phi_{k+1}$。此外，利用$\alpha_{k+1} c_k \bar{\phi}_{k+1} = ($ $\alpha_{k+1} s_k) \left(c_k \bar{\phi}_k\right) = \theta_{k+1} \phi_k$ ，我们得到

$$A^T (b - A x_k) \quad = \quad -\alpha_{k+1} c_k \bar{\phi}_{k+1} v_{k+1} = -\theta_{k+1} \phi_k v_{k+1}.$$

□

由于关系 $x_k = V_k y_k$ 和 $AV_k = U_{k+1} B_k$ 对于计算向量和系数也成立（误差不超过与机器精度 ε、向量范数以及 $\|A\|^2$相当的量），并且由于 R_k 的计算是后向稳定的，我们可以预期，数值上式 (12) 将一直成立，直到达到最大可达精度。

现在，我们在不使用任何正交性假设的情况下推导 (11) 的一个类似结果。我们将看到，所得公式将包含局部 正交性项

$$\ell_k^{(w)} \equiv \frac{\theta_{k+1}}{\rho_k} v_{k+1}^T w_k \tag{13}$$

这些量在精确算术中为零。

定理 1. 对于 *LSQR*（算法 *4*）生成的量，有

$$\|x - x_{k-1}\|_{A^T A}^2 - \|x - x_k\|_{A^T A}^2 = \phi_k^2 \left(1 + \ell_{k-1}^{(w)} - \ell_k^{(w)}\right).$$

证明. 经过简单的代数运算，我们得到

$$\|x - x_{k-1}\|_{A^T A}^2 - \|x - x_k\|_{A^T A}^2 = (x_k - x_{k-1})^T A^T A (x - x_{k-1}) + (x_k - x_{k-1})^T A^T A (x - x_k),$$

这只是对公式 (6) 的微小修改。利用这些关系

$$A^T A(x - x_k) = -\theta_{k+1} \phi_k v_{k+1}, \quad x_k - x_{k-1} = \frac{\phi_k}{\rho_k} w_k$$

我们得到

$$\|x - x_{k-1}\|_{A^T A}^2 - \|x - x_k\|_{A^T A}^2 = \phi_k^2 \left(\left[-\frac{\theta_k \phi_{k-1}}{\rho_k \phi_k}\right] w_k^T v_k - \frac{\theta_{k+1}}{\rho_k} v_{k+1}^T w_k\right).$$

注意

$$-\frac{\theta_k \phi_{k-1}}{\rho_k \phi_k} = -\frac{(s_{k-1} \alpha_k)(c_{k-1} \bar{\phi}_{k-1})}{\rho_k \phi_k} = \frac{\bar{\rho}_k \bar{\phi}_k}{\rho_k \phi_k} = 1 \tag{14}$$

从而

$$\|x - x_{k-1}\|_{A^T A}^2 - \|x - x_k\|_{A^T A}^2 = \phi_k^2 \left(w_k^T v_k - \frac{\theta_{k+1}}{\rho_k} v_{k+1}^T w_k\right).$$

最后，根据算法 4 的第 19 行，

$$v_k^T w_k = v_k^T \left(v_k + \frac{\theta_k}{\rho_{k-1}} w_{k-1}\right) = 1 + \frac{\theta_k}{\rho_{k-1}} v_k^T w_{k-1}.$$

□

因此，如果局部正交性项 (13) 很小（即其幅值远小于 1），并且 (12) 在数值上成立（即尚未达到最大可达精度水平），则 (11) 在数值上应当成立。注意，(12) 已用于定理 1 的证明。因此，在有限精度算术中，对恒等式 (11) 的合理性进行论证的问题，以这种方式归结为界定计算向量 v_{k+1} 和 w_k 之间的局部正交性的问题。

为了更好地理解术语 $\ell_k^{(w)}$，我们来认识到，LSQR 在精确算术中计算出的近似值与 CGLS 相同。因此，

$$s_k = A^T(b - Ax_k) = -\theta_{k+1}\phi_k v_{k+1}, \tag{15}$$

其中 $s_k = A^T r_k$ 在算法 3 中计算。通过比较两种算法中的 $x_k - x_{k-1}$，我们得到

$$\gamma_{k-1}p_{k-1} = \frac{\phi_k}{\rho_k}w_k\,.$$

因此，CGLS 和 LSQR 计算出相同的向量，只是缩放方式不同。特别地，w_k 是 p_{k-1} 的倍数，而 v_{k+1} 是 s_k 的倍数。这两种算法都可以视为将共轭梯度法应用于正规方程组 $A^TAx = A^Tb$ 的一种变体。最后，注意到 $\gamma_{k-1} = \rho_k^{-2}$，见式（10），我们得到

$$\ell_k^{(w)} = \frac{\theta_{k+1}}{\rho_k}v_{k+1}^T w_k = -\frac{s_k^T p_{k-1}}{\rho_k^2\phi_k^2} = -\frac{s_k^T p_{k-1}}{\|s_{k-1}\|^2},$$

其中，最后一个等式由 (14) 和 (15) 得出，从而得到 $\|s_{k-1}\|^2 = \theta_k^2\phi_{k-1}^2$。因此，局部正交性项 $\ell_k^{(w)}$ 可视为在共轭梯度法中经过详细分析的项的对应项；参见[32, 定理 9.1]。根据 [32] 的结果，可以预期 $\ell_k^{(w)}$ 的大小可由机器精度 ε 乘以某个因子来界定，该因子可能取决于 $\kappa(A^TA)$（或者在 A 奇异时取决于其对应量）、问题维数以及迭代次数。然而，进行严谨的舍入误差分析以得出上述结果超出了本文的范围。下面我们通过绘制相对误差来数值检验 (11) 的有效性。

$$\frac{\left|\left(\|x - x_{k-1}\|_{A^TA}^2 - \|x - x_k\|_{A^TA}^2\right) - \phi_k^2\right|}{\|x - x_{k-1}\|_{A^TA}^2 - \|x - x_k\|_{A^TA}^2} \tag{16}$$

对于来自 [5]、条件数为 $\kappa(A^TA) = 10^{12}$ 的（在数值稳定性方面）具有挑战性的测试案例；参见图 1。我们观察到，相对误差与误差 $\|x - x_k\|_{A^TA}$（近似）成反比，并且在达到最大可达精度之前始终显著小于 1。

总之，由 (11) 可得

$$\|x - x_\ell\|_{A^TA}^2 = \sum_{j=\ell}^{k}\phi_{j+1}^2 + \|x - x_{k+1}\|_{A^TA}^2 \tag{17}$$

从而得到误差估计器

$$\Delta_{\ell:k}^{\mathrm{LSQR}} \equiv \sum_{j=\ell}^{k}\phi_{j+1}^2 \approx \|x - x_\ell\|_{A^TA}^2 = \|r_\ell\|^2 - \|r\|^2.$$

误差估计器仅包含 LSQR 迭代过程中可获取的标量，计算成本极低。如果 v_k 和 w_{k-1} 之间的局部正交性得到了良好保持，那么即使在有限精度计算中，它也应当可靠，直至达到最大可达精度。

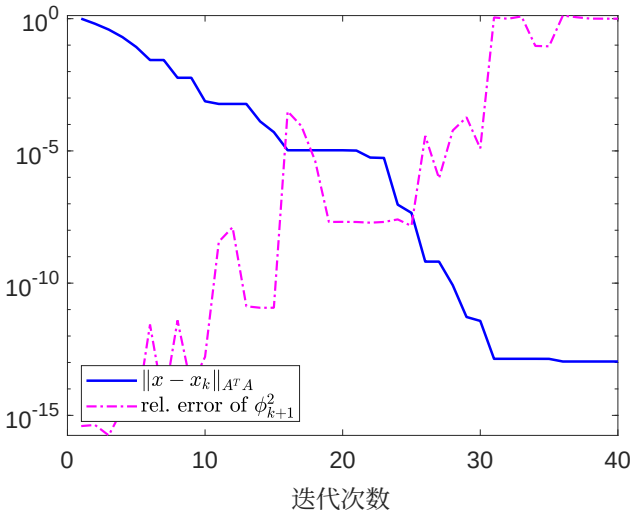


图 1：来自 [5, p. 733] 的测试案例：误差范数和相对误差 (16)。

3 最小范数问题中的误差估计

现在来考虑一个最小范数问题

$$x = \arg\min_{z \in \mathbb{R}^n} \|z\| \quad \text{subject to} \quad Az = b, \tag{18}$$

其中 $A \in \mathbb{R}^{m \times n}$ 和 $b \in \mathbb{R}^m$, $b \in \mathcal{R}(A)$。解决这一（相容）问题的一种优选方法可能是 Craig 方法，该方法将在下一节中介绍。

3.1 Craig 方法

该方法的思路是将 (18) 的解 x 写成 $x = A^T y$，其中 y 是一个适当的向量，并计算近似值 $x_k = A^T y_k$；其中 y_k 是 y 的一个近似值，该近似值由将共轭梯度法应用于该系统得到。

$$AA^T y = b, \tag{19}$$

sta从某些 y_0开始。记 $r_0 = b - AA^T y_0 = b - Ax_0$，近似值 y_k 满足 y

$$y_k \in y_0 + \mathcal{K}_k(AA^T, r_0), \qquad \|y - y_k\|_{AA^T} = \min_{z \in y_0 + \mathcal{K}_k(AA^T, r_0)} \|y - z\|_{AA^T}.$$

经过简单的代数运算可得

$$\|y - y_k\|_{AA^T}^2 = \|x - x_k\|^2, \tag{20}$$

这意味着，Craig 方法在仿射空间上最小化 x_k 的误差的欧几里得范数

$$A^T y_0 + A^T \mathcal{K}_k(AA^T, r_0) = x_0 + \mathcal{K}_k(A^T A, A^T r_0).$$

上述讨论也适用于矩阵 AA^T 为奇异矩阵的情况；参见第 1 节中的讨论。

与求解正规方程 (4) 类似，应用于 (19) 的共轭梯度法有几种不同方式。文献 [5]研究了各种实现方式的数值稳定性。

第一个算法由 Craig 在 [8, 9], 提出，最初是作为求解非对称系统的算法开发的（参见 Forsythe 在 MathSciNet 上撰写的评论）。Fadeev 和 Fadeeva 开发了几乎相同的算法 [14, 第 504] 页，其区别在于，该算法不是使用方向向量，而是

算法 5CGNE

输入 A, b, x_0
$r_0 = b - Ax_0$
$p_0 = A^T r_0$
对于 $k = 0, 1, 2, \ldots$ **执行**
  $\gamma_k = \|r_k\|^2 / \|p_k\|^2$
  $x_{k+1} = x_k + \gamma_k p_k$
  $r_{k+1} = r_k - \gamma_k A p_k$
  $\delta_{k+1} = \|r_{k+1}\|^2 / \|r_k\|^2$
  $p_{k+1} = A^T r_{k+1} + \delta_{k+1} p_k$
结束循环

更新向量 $g_k \equiv A^T p_k$。所得算法称为 CGNE，并列于算法 5 中。除记号不同外，相同的算法也见于 [5, 算法 21]，其中设定 $\mu = 0$。

Paige [26] 和 Paige and Saunders [27] 利用 Golub–Kahan 双对角化提出了该算法的另一种数学等价形式。该算法还可以计算 $AA^T y = b$ 的近似值 y_k。我们在算法 6 中给出该算法，并将其称为 CRAIG，以与 Paige 和 Saunders 引入的记号保持相容。

算法 6CRAIG

输入 A, b
$\zeta_0 = -1$
$x_0 = y_0 = d_0 = 0$
$\beta_1 u_1 = b$　　β_i 是归一化系数，使其具有 $\|u_i\| = 1$、$i = 1$、2、.....
$\alpha_1 v_1 = A^T u_1$　　α_i 是归一化系数，使其具有 $\|v_i\| = 1$、$i = 1$、2、.....
对于 $k = 1, 2, \ldots$ **执行**
  $\zeta_k = -\frac{\beta_k}{\alpha_k} \zeta_{k-1}$
  $x_k = x_{k-1} + \zeta_k v_k$
  $d_k = (u_k - \beta_k d_{k-1}) / \alpha_k$
  $y_k = y_{k-1} + \zeta_k d_k$　　如果不需要近似值 y_k，则可以删除第 9–10 行。
  $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$
  $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$
结束循环

现在，我们给出一种估计 Craig 方法两种实现中误差 $\|x - x_k\|^2$ 的方法。与式（6）类似，通过简单的代数运算，我们得到

$$\|x - x_k\|^2 - \|x - x_{k+1}\|^2 = \|x_{k+1} - x_k\|^2 + 2\,(x_{k+1} - x_k)^T\,(x - x_{k+1})\,. \tag{21}$$

3.2 CGNE 中的误差估计

首先注意到，在 CGNE 中，

$$\|x_{k+1} - x_k\|^2 = \gamma_k^2 \|p_k\|^2 = \gamma_k \|r_k\|^2,$$

其中我们使用了 $\gamma_k \|p_k\|^2 = \|r_k\|^2$。

形式上，考虑用于计算向量 y_k 的递推关系，使得 $x_k = A^T y_k$；以及用于计算相应方向向量 s_k 的递推关系，使得 $p_k = A^T s_k$，

$$\begin{aligned} y_{k+1} &= y_k + \gamma_k s_k, \\ s_{k+1} &= r_{k+1} + \delta_{k+1} s_k, \end{aligned}$$

其中 y_0 满足 $x_0 = A^T y_0$ 和 $s_0 = r_0$。那么，(21) 右端的最后一项对应于 t o

$$(x_{k+1} - x_k)^T (x - x_{k+1}) = (y_{k+1} - y_k)^T A (x - x_{k+1}) = \gamma_k s_k^T r_{k+1},$$

因此使得

$$\|x - x_k\|^2 - \|x - x_{k+1}\|^2 = \gamma_k \|r_k\|^2 + 2\gamma_k s_k^T r_{k+1}.$$

借助 [32] 中关于应用于 $AA^T y = b$ 的共轭梯度法结果的类比，可以预期，项 $s_k^T r_{k+1}$ 的大小相较于 $\|r_k\|^2$ 应可忽略，并且该恒等式

$$\|x - x_k\|^2 - \|x - x_{k+1}\|^2 = \gamma_k \|r_k\|^2$$

在数值上成立，直至达到最大可达精度的水平。因此，对于 CGNE，我们得到与 (2) 类似的形式：

$$\|x - x_\ell\|^2 = \sum_{j=\ell}^{k} \gamma_j \|r_j\|^2 + \|x - x_{k+1}\|^2$$

给出误差估计器

$$\Delta_{\ell:k}^{\mathrm{CGNE}} \equiv \sum_{j=\ell}^{k} \gamma_j \|r_j\|^2 \approx \|x - x_\ell\|^2.$$

3.3 CRAIG 中的误差估计

Co考虑迭代索引 $k-1$ 对应的公式（21），并使用 $x_k = A^T y_k$，我们得到 n

$$\|x - x_{k-1}\|^2 - \|x - x_k\|^2 = \|x_k - x_{k-1}\|^2 + 2 (y_k - y_{k-1})^T A (x - x_k).$$

我们首先注意到

$$\|x_k - x_{k-1}\|^2 = \zeta_k^2 \|v_k\|^2 = \zeta_k^2.$$

使用与引理1中类似的技巧，可以证明

$$A(x - x_k) = AA^T (y - y_k) = -\zeta_k \beta_{k+1} u_{k+1},$$

因此，由算法6第10行可得，

$$(y_k - y_{k-1})^T A (x - x_k) = -\zeta_k^2 \beta_{k+1} d_k^T u_{k+1}.$$

因此，假设尚未达到最大精度水平，并且 d_k（式（19）中 CG 的方向向量）与 u_{k+1}（式（19）中 CG 的缩放残差向量）之间的局部正交性得到了良好保持，则恒等式

$$\|x - x_{k-1}\|^2 - \|x - x_k\|^2 = \zeta_k^2$$

从而得到

$$\|x - x_\ell\|^2 = \sum_{j=\ell}^{k} \zeta_{j+1}^2 + \|x - x_{k+1}\|^2$$

在数值上成立。关于 CRAIG 中局部正交性的保持问题，详细的舍入误差分析超出了本文的范围。最后，CRAIG 的误差估计器形式如下

$$\Delta_{\ell:k}^{\mathrm{CRAIG}} \equiv \sum_{j=\ell}^{k} \zeta_{j+1}^2 \approx \|x - x_\ell\|^2.$$

4 预条件算法中的误差估计

本节介绍这些算法的预条件变体，并推导相应的误差估计。我们仅考虑正规方程的分裂预条件子类，即形式上将系统 (4) 和 (19) 转换为

$$\big(L^{-1}A^TAL^{-T}\big)\big(L^Tx\big) = L^{-1}A^Tb \quad \text{and} \quad \big(L^{-1}AA^TL^{-T}\big)\big(L^Ty\big) = L^{-1}b, \text{ respectively.}$$

在 [6, 第 1], 节中讨论了几种用于最小二乘问题的预条件技术。其中大多数可以表示为正规方程的分裂预条件子。一般而言，分裂预条件子的典型代表是矩阵 A^TA或 AA^T的不完全分解。现已有无需显式构造 A^TA 或 AA^T 即可高效计算此类分解的代码；例如，可参见 [1]中的 HSL_MI35。

首先我们将讨论 CGLS 和 LSQR 的预条件，然后讨论 CGNE 和 CRAIG 的预条件。

4.1 预条件共轭梯度最小二乘法与 LSQR

设给定一个非奇异矩阵 $L \in \mathbb{R}^{n\times n}$ 。对于最小二乘问题，我们将原问题（3）修改为如下形式:

$$x = \underset{z\in\mathbb{R}^n}{\arg\min} \|b - \underbrace{AL^{-T}}_{\hat{A}}\underbrace{L^Tz}_{\hat{z}}\| \;=\; \underset{\hat{z}\in\mathbb{R}^n}{\arg\min} \|b - \hat{A}\hat{z}\|, \tag{22}$$

从而得到相应的正规方程组

$$\underbrace{L^{-1}A^T}_{\hat{A}^T}\underbrace{AL^{-T}}_{\hat{A}}\underbrace{L^Tx}_{\hat{x}} = \underbrace{L^{-1}A^T}_{\hat{A}^T}b. \tag{23}$$

因此， L 可以看作矩阵 A^TA 的分裂预条件子。设 $\hat{}x_k$ 为预条件系统的第 k 个共轭梯度法近似解，并用 $x_k = L^{-T}\hat{}x_k$ 定义原问题 (3) 的解 x 的近似值。由 (23) 可知 $\hat{}x = L^Tx$ 和 $\hat{}A = AL^{-T}$，我们有

$$\|\hat{x} - \hat{x}_k\|^2_{\hat{A}^T\hat{A}} = (\hat{x} - \hat{x}_k)^T\hat{A}^T\hat{A}(\hat{x} - \hat{x}_k) = (x - x_k)^TLL^{-1}A^TAL^{-T}L^T(x - x_k) = \|x - x_k\|^2_{A^TA}. \tag{24}$$

用于求解 (22) 的 CGLS 算法基于算法 3，见算法 7。经过简单的代数运算可得， $x_k = L^{-T}\hat{}x_k$ 对应的残差 $r_k = b - Ax_k$ 满足 $r_k = \hat{}r_k$。因此，我们可以写出如算法 8 所示的预条件 CGLS 变体，其中显式计算近似值 x_k 及其对应的残差 r_k 。

与第 2 节类似，我们可以证明这些误差满足

$$\|\hat{x} - \hat{x}_k\|^2_{\hat{A}^T\hat{A}} = \sum_{j=\ell}^{k}\hat{\gamma}_j\|\hat{s}_j\|^2 + \|\hat{x} - \hat{x}_{k+1}\|^2_{\hat{A}^T\hat{A}}.$$

因此，我们可以使用该估计器估算预条件共轭梯度最小二乘法中的误差。

$$\Delta^{\text{PCGLS}}_{\ell:k} \equiv \sum_{j=\ell}^{k}\hat{\gamma}_j\|\hat{s}_j\|^2 \approx \|x - x_\ell\|^2_{A^TA},$$

具有与 $\Delta^{\text{CGLS}}_{\ell:k}$ 相同的良好性质。

当预条件子无法以因式分解形式提供时，也可以轻松修改算法和估计器以适用于这种情况。例如，可在 [29,算法 4]中找到 PCGLS 的无因式分解变体。

算法 7 形式上的预条件 CGLS

输入 A, b, x_0

$r_0 = b - Ax_0$

$\hat{s}_0 = p_0 = A^T r_0$

对于 $k = 0, 1, \ldots$ **do**

$q_k = Ap_k$

$\gamma_k = \|\hat{s}_k\|^2 / \|q_k\|^2$

$x_{k+1} = x_k + \gamma_k p_k$

$r_{k+1} = r_k - \gamma_k q_k$

$\hat{s}_{k+1} = A^T r_{k+1}$

$\delta_{k+1} = \|\hat{s}_{k+1}\|^2 / \|\hat{s}_k\|^2$

$p_{k+1} = \hat{s}_{k+1} + \delta_{k+1} p_k$

结束循环

算法 8 预条件共轭梯度最小二乘法

输入 A, b, x_0, L

$r_0 = b - Ax_0$

<escape>$\hat{s}_0 = p_0 = L^{-1}A^T r_0$</escape>

对于 $k = 0, 1, \ldots$ **do**

$t_k = L^{-T} p_k$

$q_k = At_k$

$\gamma_k = \|\hat{s}_k\|^2 / \|q_k\|^2$

$x_{k+1} = x_k + \gamma_k t_k$

$r_{k+1} = r_k - \gamma_k \hat{q}_k$

$\hat{s}_{k+1} = L^{-1}A^T r_{k+1}$

$\delta_{k+1} = \|\hat{s}_{k+1}\|^2 / \|\hat{s}_k\|^2$

$p_{k+1} = \hat{s}_{k+1} + \delta_{k+1} p_k$

结束循环

对于 LSQR，我们可以采用与 CGLS 类似的方法，直接将 LSQR 应用于 (22)。在手头获得 x_k 后停止迭代，原始系统解的近似值 x 计算如下：$x_k = L^{-T} x_k$。所得算法如算法 9 所示。

根据估计量 $\Delta_{\ell k}^{\mathrm{LSQR}}$: 的定义（针对 (22) 考虑）以及 (24)，我们推导出预条件 LSQR 的估计量。

$$\Delta_{\ell:k}^{\mathrm{PLSQR}} \equiv \sum_{j=\ell}^{k} \hat{\phi}_{j+1}^2 \approx \|\hat{x} - \hat{x}_\ell\|_{\hat{A}^T \hat{A}}^2 = \|x - x_\ell\|_{A^T A}^2.$$

算法 9 预条件 LSQR

输入 A, b, L

$\beta_1 u_1 = b$ $\quad$ $\hat{\beta}_i$ 是使 $\|u_i\| = 1$, $i = 1, 2, \ldots$ 具有所需形式的归一化系数

$\alpha_1 v_1 = L^{-1}A^T u_1$ $\quad$ α_i 是使 $\|v_i\| = 1$, $i = 1, 2, \ldots$ 具有所需形式的归一化系数

$\hat{w}_1 = \hat{v}_1$

$x_0 = 0$

$\bar{\phi}_1 = \hat{\beta}_1$

$\bar{\rho}_1 = \hat{\alpha}_1$

对于 $k = 1, 2, \ldots$ **执行**

$\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k = AL^{-T} v_k - \alpha_k u_k$

$\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k = L^{-1}A^T u_{k+1} - \beta_{k+1} v_k$

$\rho_k = (\bar{\rho}_k^2 + \beta_{k+1}^2)^{1/2}$

$\hat{c}_k = \bar{\rho}_k / \rho_k$

$\hat{s}_k = \beta_{k+1} / \rho_k$

$\hat{\theta}_{k+1} = \hat{s}_k \hat{\alpha}_{k+1}$

$\bar{\rho}_{k+1} = -\hat{c}_k \hat{\alpha}_{k+1}$

$\phi_k = \hat{c}_k \bar{\phi}_k$

$\bar{\phi}_{k+1} = \hat{s}_k \bar{\phi}_k$

$x_k = x_{k-1} + (\bar{\phi}_k / \rho_k) w_k$

$w_{k+1} = v_{k+1} + (\hat{\theta}_{k+1} / \rho_k) w_k$

结束循环

$x_k = L^{-T} x_k$

Fa预条件 LSQR 的无分解变体可见于例如 [4, 算法 2] .

算法 10 预条件 CGNE

1: **输入** A, b, x_0, L2: $\hat{r}_0 = L^{-1}($
$b - Ax_0$)3: $\hat{p}_0 = A^T L^{-T} \hat{r}_0$4: **对于**
$k = 0, 1, 2, \ldots$ **执行**5: $\hat{\gamma}_k = \|\hat{r}$
$r_k\|^2 / \|\hat{p}_k\|^2$6: $x_{k+1} = x_k + \hat{\gamma}_k \hat{p}_k$
7 $r_{k+1} = r_k -$ $p_k : \hat{\ } \hat{\ } \hat{\ }$
8
$\gamma_k L^{-1} A\hat{}$: $\hat{\delta}_{k+1} = \|\hat{r}_{k+1}\|^2 / \|\hat{r}_k\|^2$
9: $\hat{p}_{k+1} = A^T L^{-T} \hat{r}_{k+1} + \hat{\delta}_{k+1} \hat{p}_k$

10: **结束循环**

4.2 预条件 CGNE 和 CRAIG

设非奇异的 L $\in \mathbb{R}^{m \times m}$ 已给定。我们考虑原问题（18）的以下修改)

$$x = \arg\min_{z \in \mathbb{R}^n} \|z\| \quad \text{subject to} \quad \underbrace{L^{-1}A}_{\hat{A}} z = \underbrace{L^{-1}b}_{\hat{b}},$$

并且，与之前一样，将所考虑的算法应用于相应的预条件系统时，可以得到近似解 x_k

$$\underbrace{L^{-1}A}_{\hat{A}} \underbrace{A^T L^{-T}}_{\hat{A}^T} \underbrace{L^T y}_{\hat{y}} = \underbrace{L^{-1}b}_{\hat{b}}.$$

因此， L 可以被视为矩阵 AA^T 的一种分裂预条件子。我们注意到，无需对计算得到的近似值进行任何变换，即可获得解 x 的一个近似值。类似于第 3.1 节，我们可以证明，预条件变体在仿射空间 $x_0 + \mathcal{K}_k($ $L^{-1}A^T AL^{-T}, A^T L^{-T} L^{-1} r_0)$ 上最小化误差 $\|x - x_k\|^2$ 的欧几里得范数。

对于 CGNE（算法 5），我们得到其预条件变体算法 10。每当迭代过程中需要残差$r_k = b - Ax_k$时，我们可以使用变换 $\hat{r}_k = L^{-1} r_k$，并将算法 10 中的第 2 行和第 7 行替换为

$$2 : r_0 = b - Ax_0; \quad \hat{r}_0 = L^{-1} r_0$$
$$7 : r_{k+1} = r_k - \hat{\gamma}_k A\hat{p}_k; \quad \hat{r}_k = L^{-1} r_k$$

同样，我们可以对 GRAIG 算法（算法 6）进行预条件处理。不计算y_k 向量的版本见算法 11。

按照第 3.2 节和第 3.3 节中的推导，我们得到 PCGNE 和 PCRAIG 中的误差估计器，

$$\Delta_{\ell:k}^{\text{PCGNE}} \equiv \sum_{j=\ell}^{k} \hat{\gamma}_j \|\hat{r}_j\|^2 \approx \|x - x_\ell\|^2, \quad \Delta_{\ell:k}^{\text{PCRAIG}} \equiv \sum_{j=\ell}^{k} \hat{\zeta}_{j+1}^2 \approx \|x - x_\ell\|^2.$$

5 数值实验

对于数值测试，我们考虑了来自 SuiteSparse[1] 矩阵集合的若干矩阵和系统，其规模为

illc1033*: 1033×320illc1850*:
1850×712well1033*: 1033×320
well1850*: 1850×712sls:
$1\,748\,122 \times 62\,729$

Delor64K: $64\,719 \times 1\,785\,345$
Delor338K: $343\,236 \times 887\,058$花朵_7_4:
$27\,693 \times 67\,593$猫_耳朵_4_4:
$19\,020 \times 44\,448$lp_飞行员: 1441×4860

[1] https://sparse.tamu.edu, [10]

算法 11 预条件化 CRAIG

输入 A, b, L
$\hat{\zeta}_0 = -1$
$x_0 = 0$
$\hat{\beta}_1 \hat{u}_1 = L^{-1} b$　　$\hat{\beta}_i$ 是归一化系数，使其具有 $\|\hat{u}_i\| = 1$, $i = 1, 2, \ldots$
$\hat{\alpha}_1 v_1 = A^T L^{-T} \hat{u}_1$　　$\hat{\alpha}_i$ 是归一化系数，使其具有 $\|\hat{v}_i\| = 1$, $i = 1, 2, \ldots$
对于 $k = 1, 2, \ldots$ **做**
　$\zeta_k = -\frac{\hat{\beta}_k}{\hat{\alpha}_k} \hat{\zeta}_{k-1}$
　$x_k = x_{k-1} + \hat{\zeta}_k \hat{v}_k$
　$\hat{\beta}_{k+1} \hat{u}_{k+1} = L^{-1} A \hat{v}_k - \hat{\alpha}_k \hat{u}_k$
　$\hat{\alpha}_{k+1} \hat{v}_{k+1} = A^T L^{-T} \hat{u}_{k+1} - \hat{\beta}_{k+1} \hat{v}_k$
结束循环

带星号的矩阵与右端项 b 一同给出。对于其余问题，我们按如下方式生成了 b （使用 MATLAB 表示法）

```
x = ones(size(A,2),1); x(2:2:end) = -2;
x(5:5:end) = 0; b_LN = A*x; b_LS = b_LN +
randn(size(b_LN))*norm(b_LN);
```

其中， b_{LN} 用于最小范数问题， b_{LS} 用于最小二乘问题。随后使用 MATLAB 内置函数 lsqminnorm 计算精确解。

与 [24], 中一样，我们绘制误差量以及（自适应）下界，并将自适应延迟值 $k - \ell$ 与理想值进行比较，即确保下界达到规定精度所需的最小延迟。

5.1 最小二乘问题

在图 2–6 中，我们绘出了使用 CGLS 和 LSQR 求解最小二乘问题时自适应误差估计的结果。可以看到，该估计值大多紧跟误差变化，延迟几乎达到最优；即使在几乎停滞、最优延迟非常大的情形下也是如此；参见图 2。对于 illc1033 矩阵，我们观察到初始迭代次数中存在一定的低估，但在后续迭代中（通常需要依据误差估计来停止求解器），延迟接近理想值。对于矩阵 sls，我们注意到自适应选取的延迟值高于实际所需。

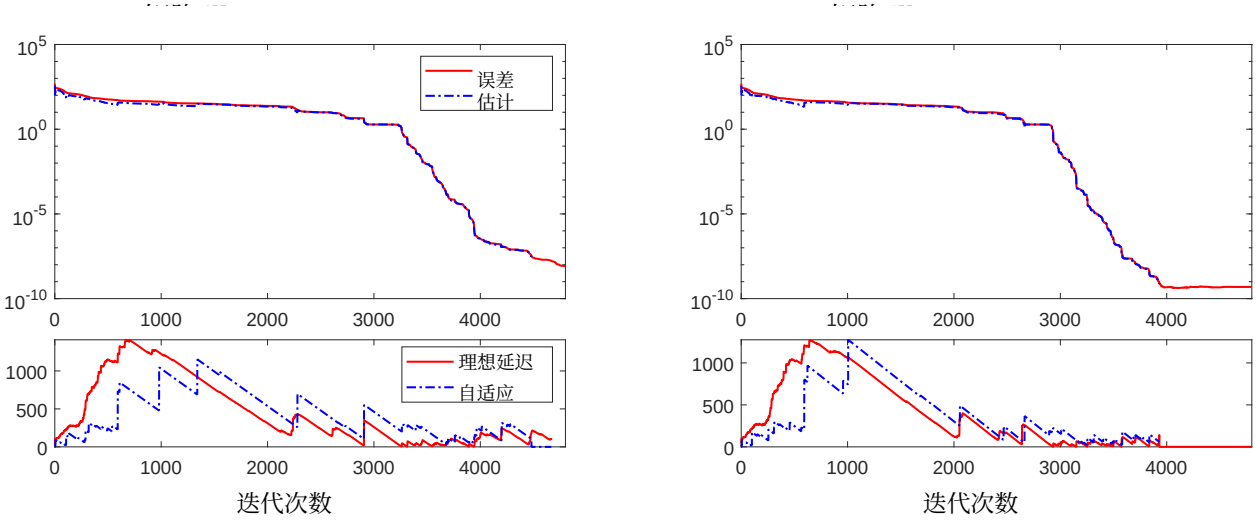


图 2：矩阵 illc1033，CGLS（左）和 LSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 和其理想值（下）

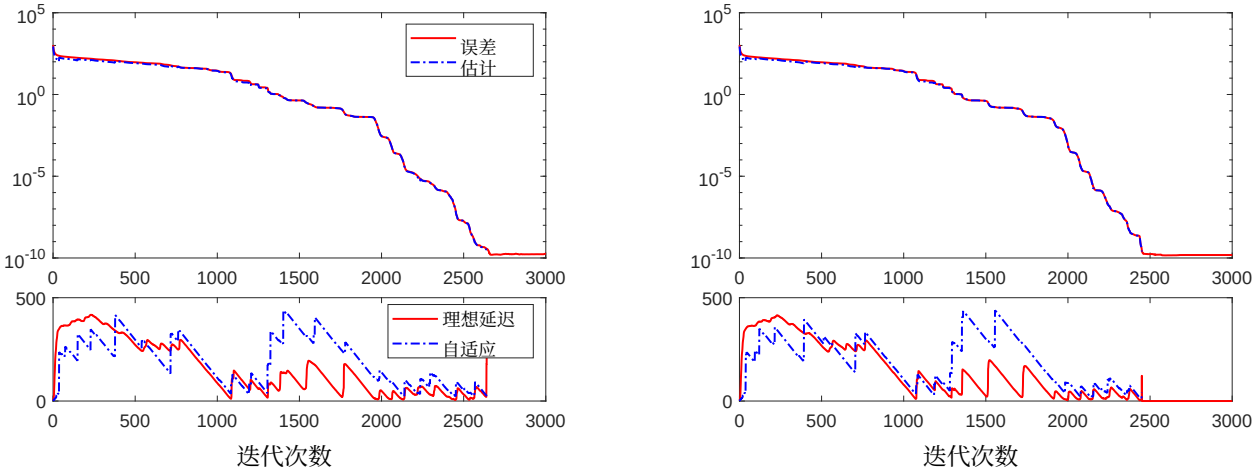


图 3：矩阵 illc1850，CGLS（左）和 LSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

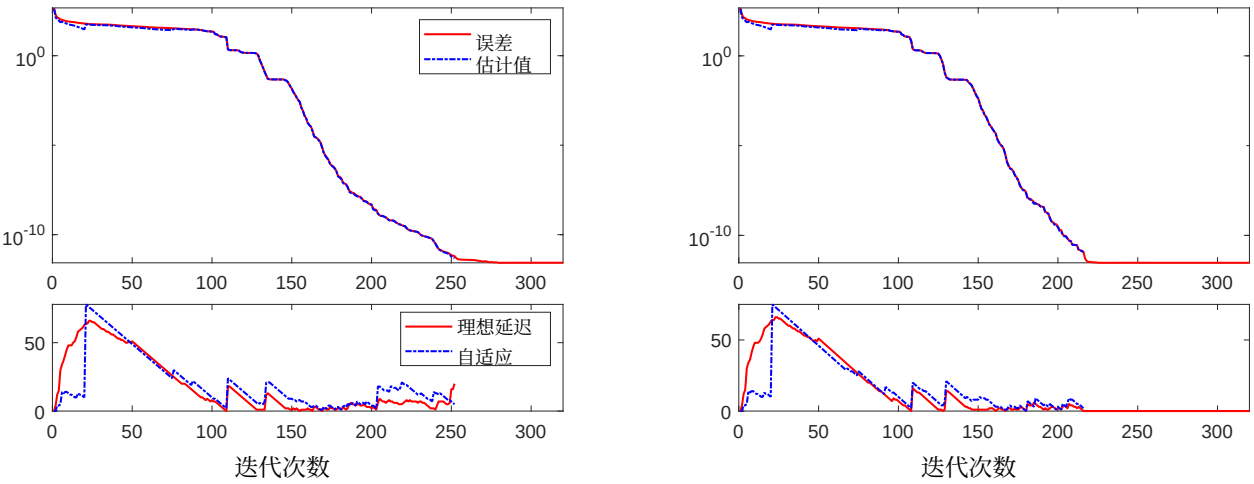


图 4：矩阵 well1033，CGLS（左）和 LSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 及自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

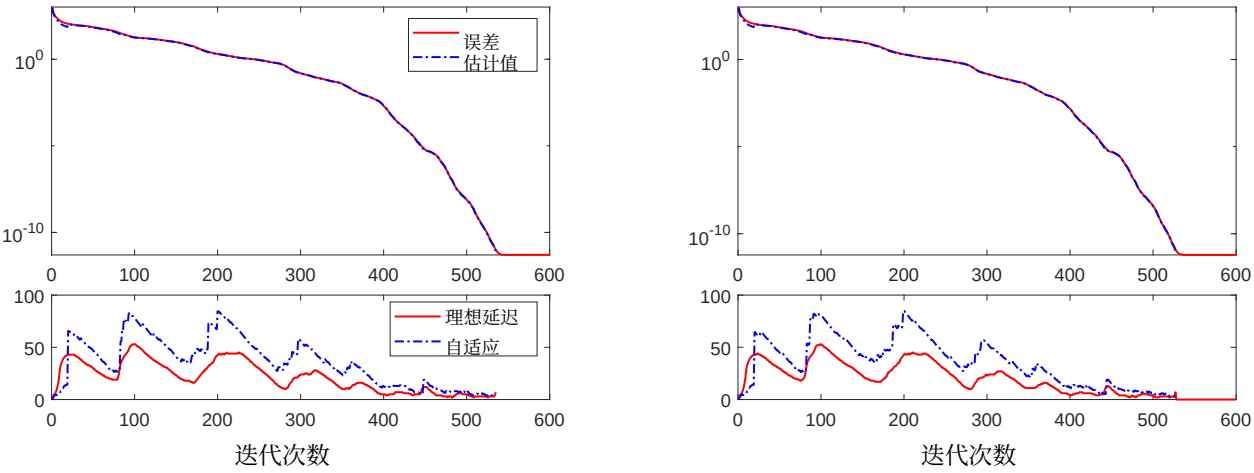


图 5：矩阵 well1850，CGLS（左）和 LSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

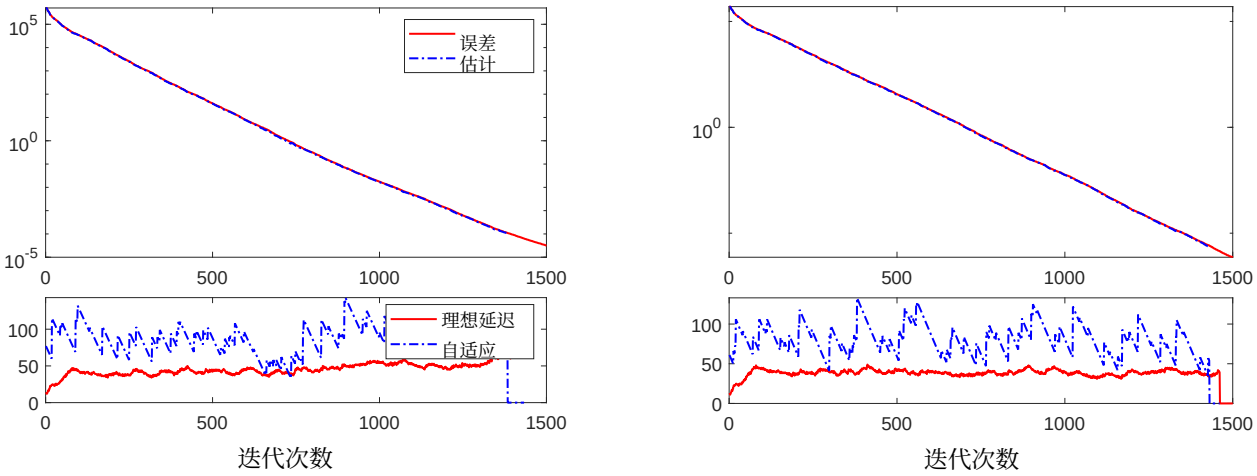


图 6：矩阵 sls，CGLS（左）和 LSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

5.2 最小范数问题

通过 CGNE 和 CRAIG 求解最小范数问题的结果见图 7–11。这里我们观察到非常令人满意的表现，Delor338K 和 lp_pilot 的延迟略高；尽管如此，自适应选择的延迟很好地跟随了理想值的增减。

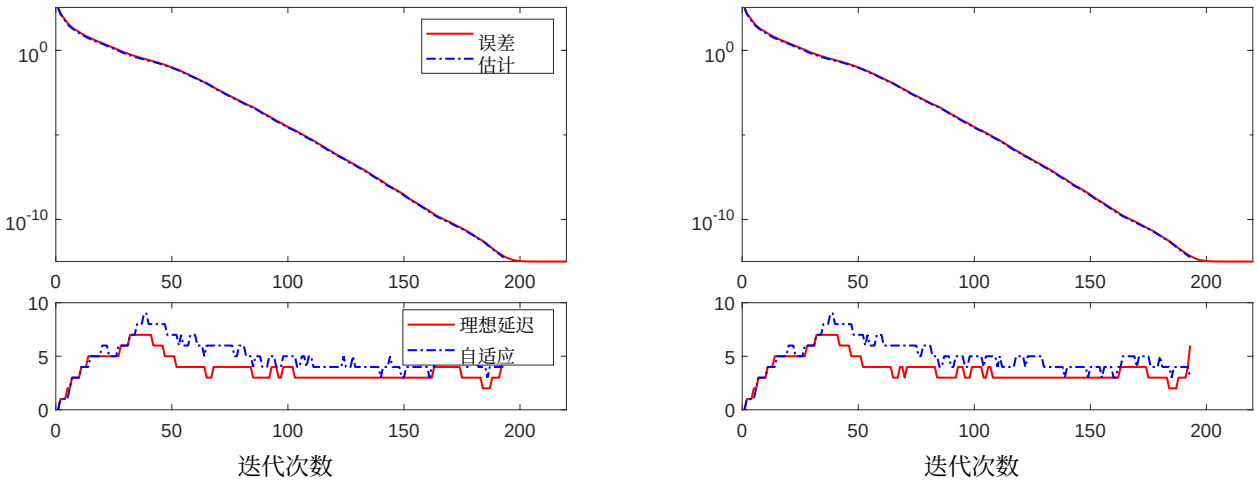


图 7：矩阵 Delor64K，CGNE（左）和 CRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

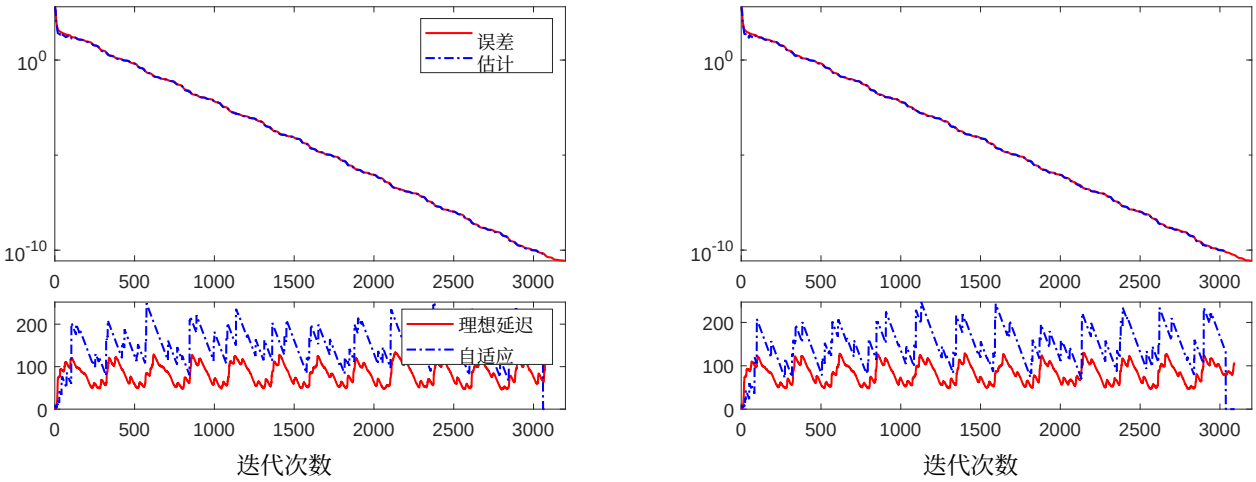


图 8：矩阵 Delor338K，CGNE（左）和 CRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

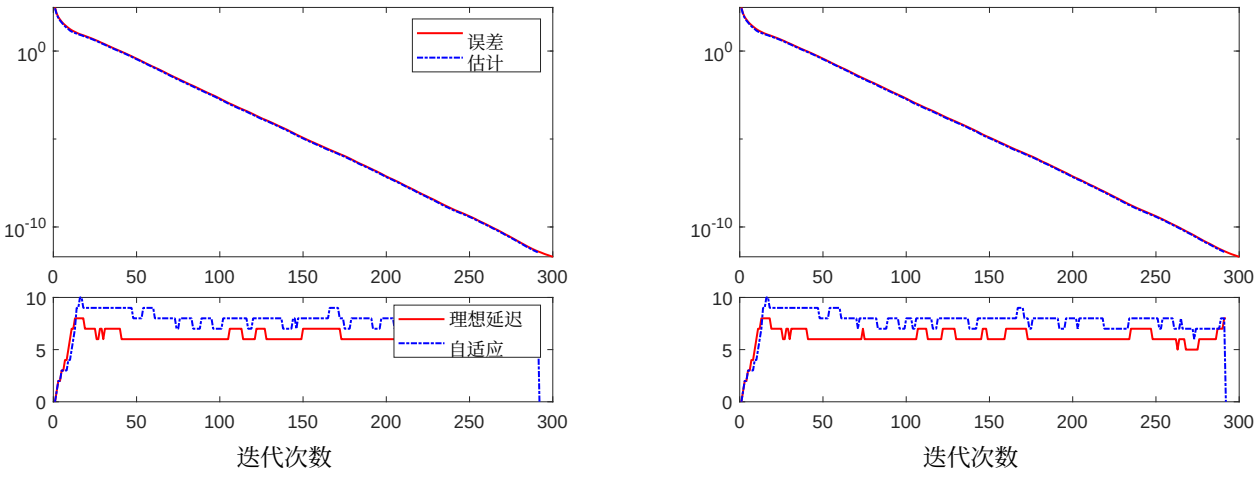


图 9：矩阵花_7_4，CGNE（左）和 CRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

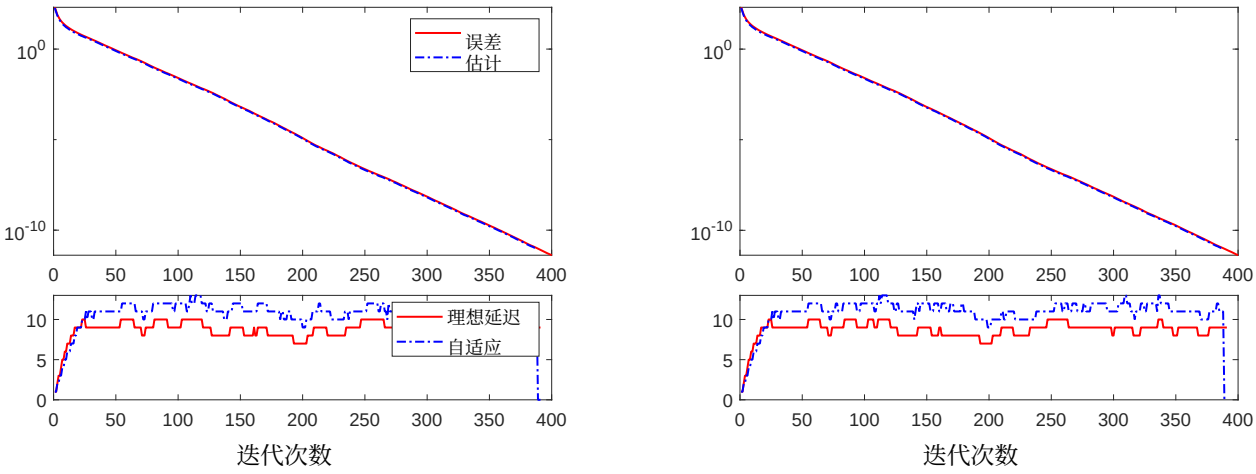


图10：矩阵 cat_ears_4_4，CGNE（左）和 CRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

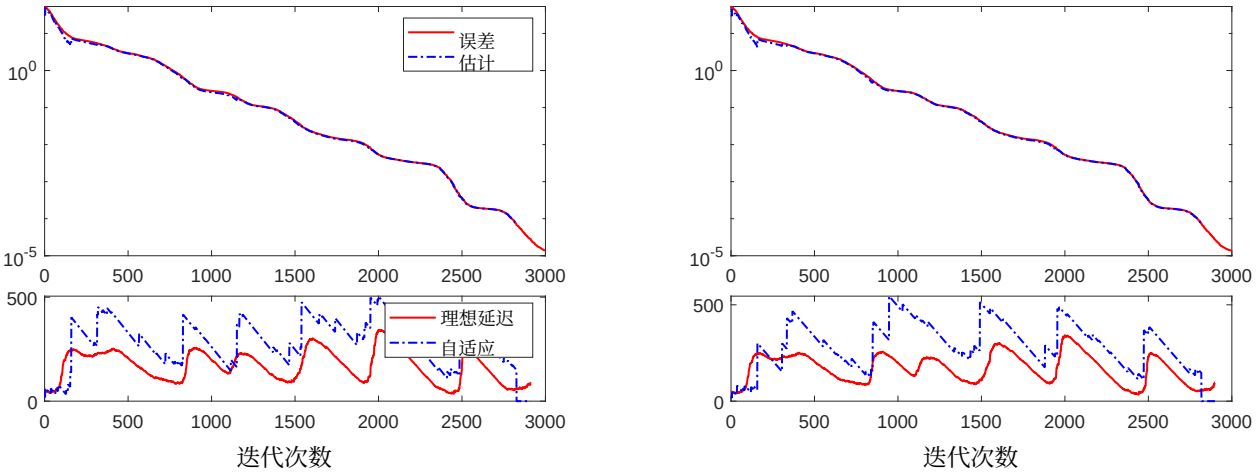


图11：矩阵 lp_pilot，CGNE（左）和 CRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

5.3 预条件最小二乘问题

图 12–16 展示了我们对由 CGLS 和 LSQR 求解的预条件最小二乘问题进行自适应误差估计的结果。预条件子 L 由 [1], 中的 A 构造，使用 HSL_MI35 的 MATLAB 接口，即计算 A^TA的不完全 Cholesky 分解，而不显式形成该矩阵。从图中可以看出，自适应误差估计的表现非常令人满意。

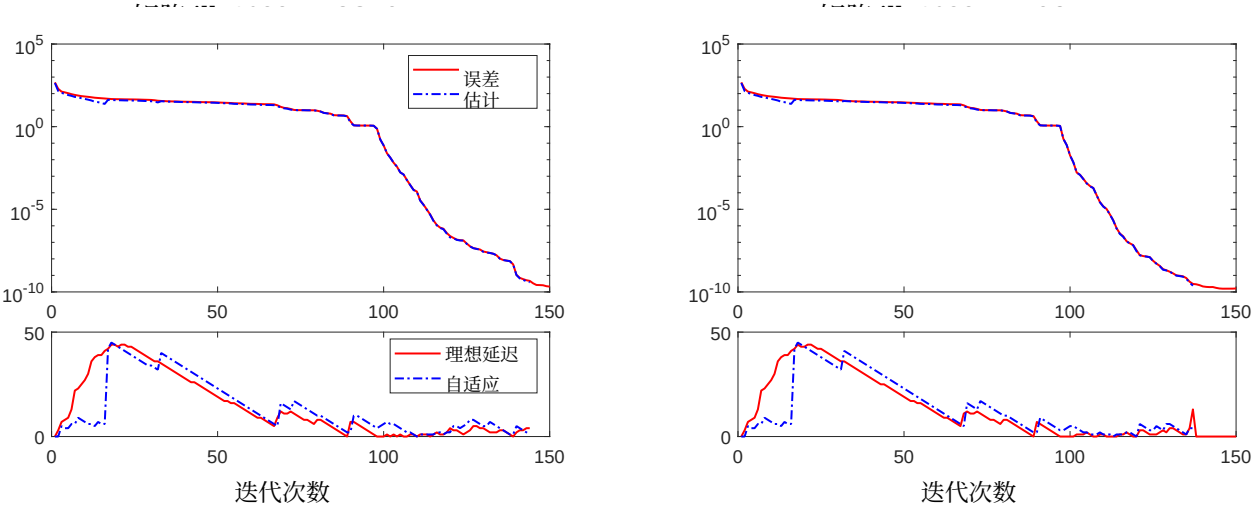


图 12：经过预条件处理的矩阵 illc1033，PCGLS（左）和 PLSQR（右）：误差 $\|x - x_k\|_{A^TA}$和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

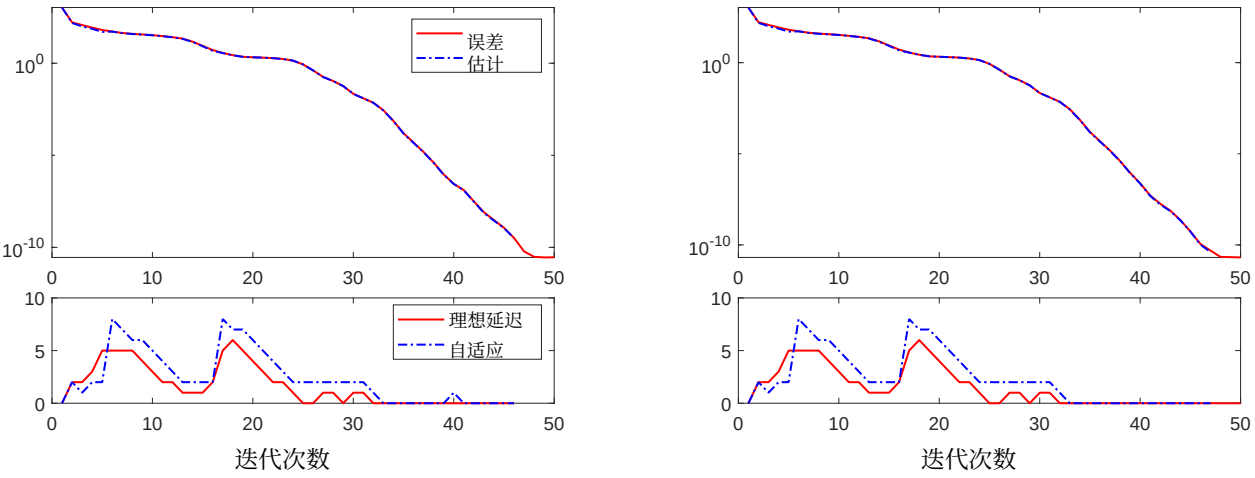


图 13：矩阵 illc1850 经预条件处理后，PCGLS（左）和 PLSQR（右）：误差 $\|x - x_k\|_{A^TA}$及自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

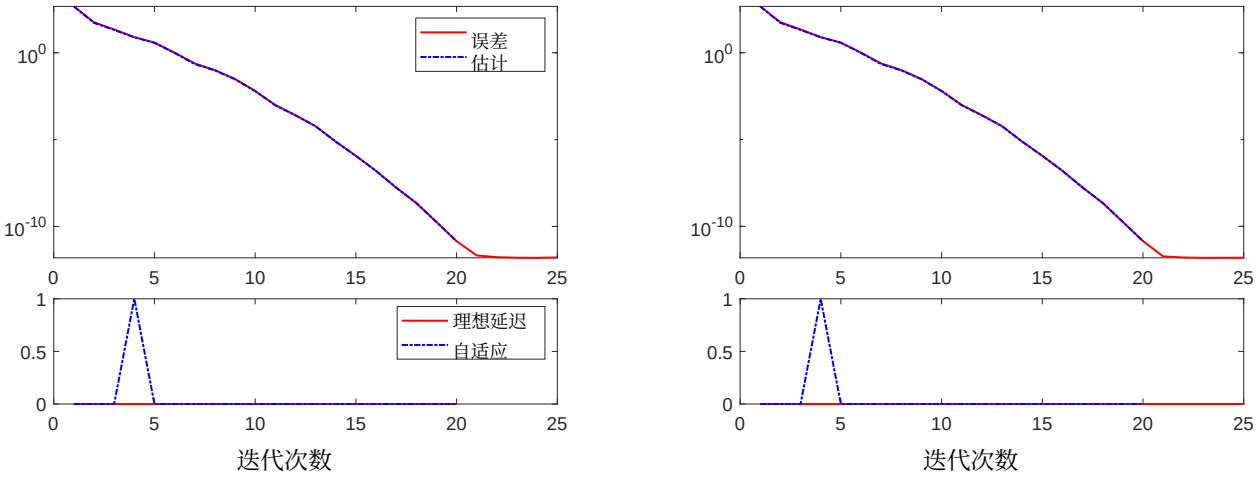


图 14：预条件下的矩阵 well1033，PCGLS（左）和 PLSQR（右）：误差 $\|x - x_k\|_{A^TA}$和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

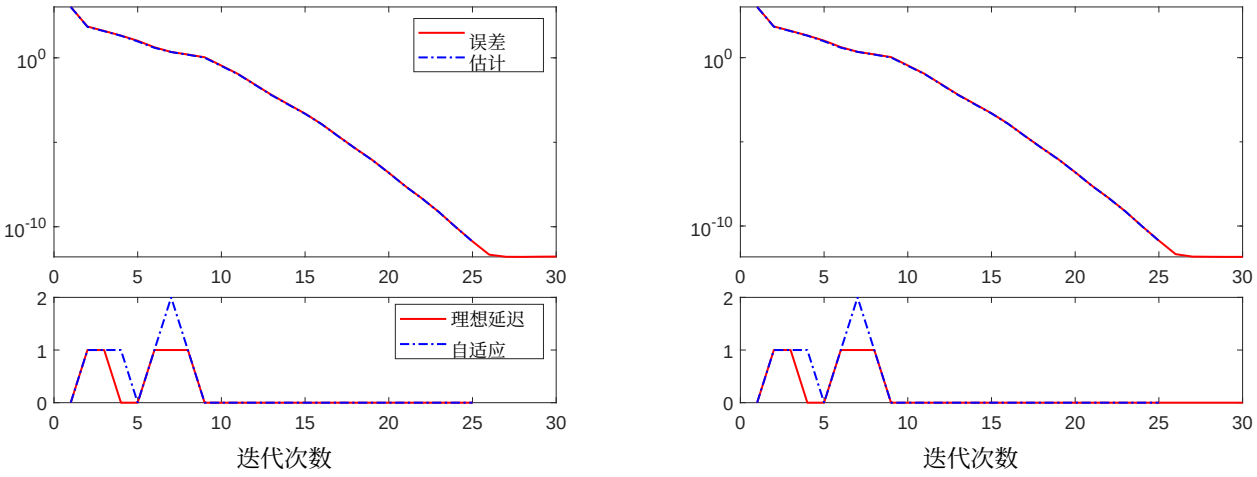


图 15：矩阵 well1850 经预条件处理后，PCGLS（左）和 PLSQR（右）：误差 $\|x - x_k\|_{A^TA}$和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

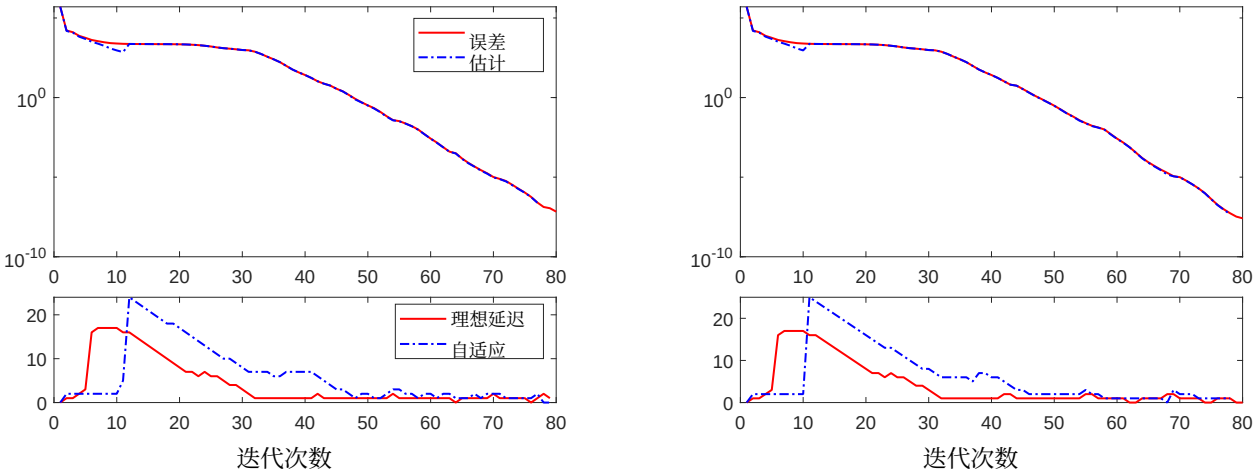


图 16：预条件下的矩阵 sls，PCGLS（左）和 PLSQR（右）：误差 $\|x - x_k\|_{A^TA}$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

5.4 预条件最小范数问题

图 17–21 给出了使用 CGNE 和 CRAIG 求解预条件最小范数问题的结果。与上一小节一样，使用 HSL_MI35 构造预条件子 L。实验确认，该估计也可可靠地用于预条件最小范数问题。

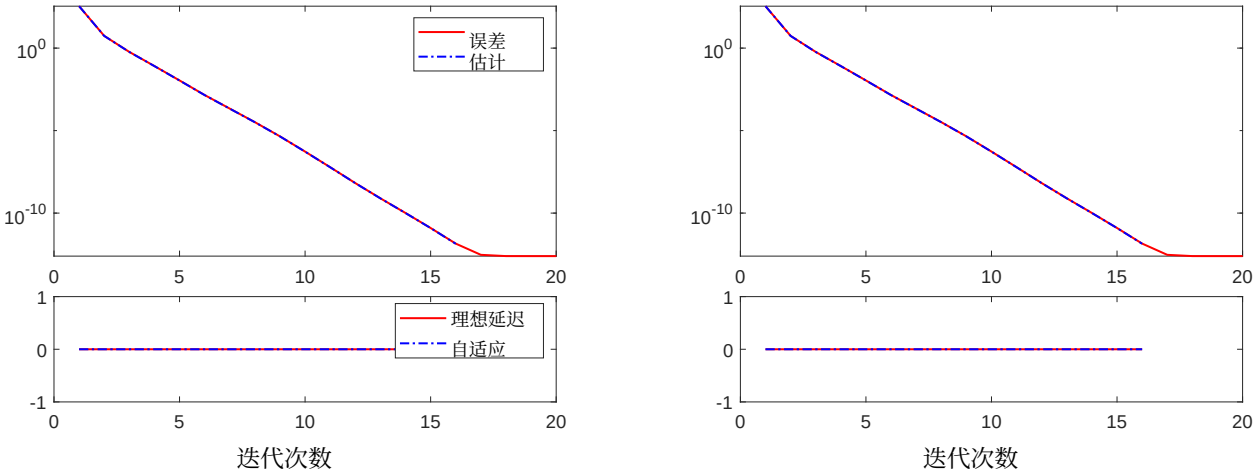


图 17：采用预条件的矩阵 Delor64K，PCGNE（左）和 PCRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

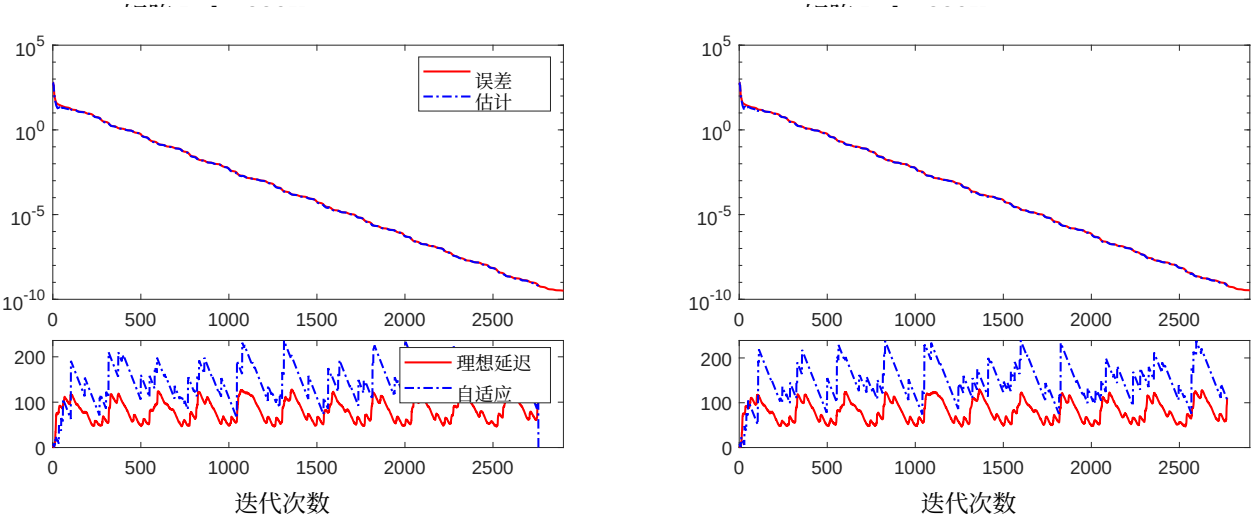


图 18：经过预条件处理的矩阵 Delor338K，PCGNE（左）和 PCRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

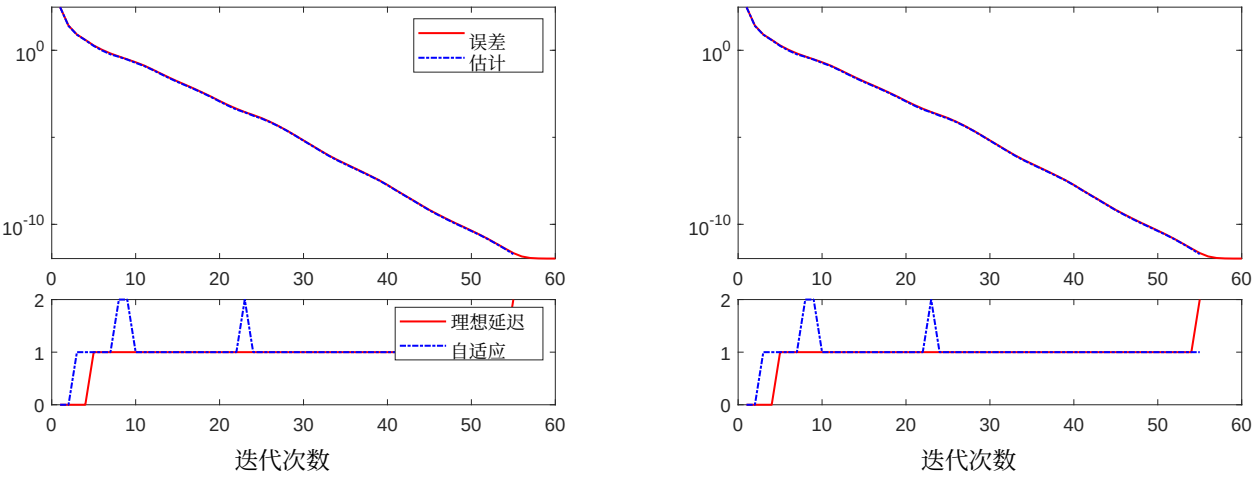


图 19：带预条件的矩阵花_7_4，PCGNE（左）和 PCRAIG（右）：误差 $\|x - x_k\|$和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

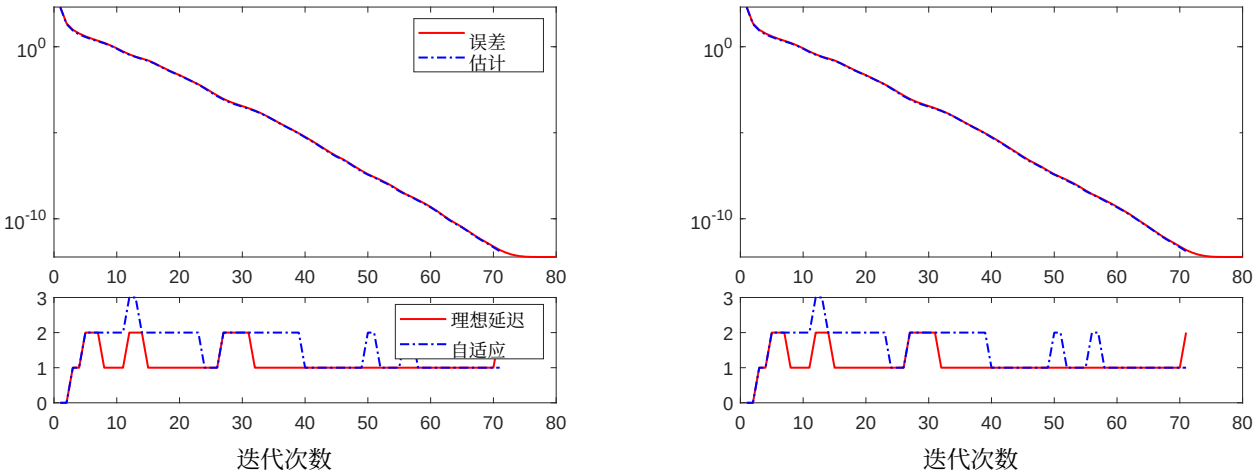


图20：预条件下的矩阵 cat_ears_4_4，PCGNE（左）和 PCRAIG（右）：误差 $\|x - x_k\|$和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

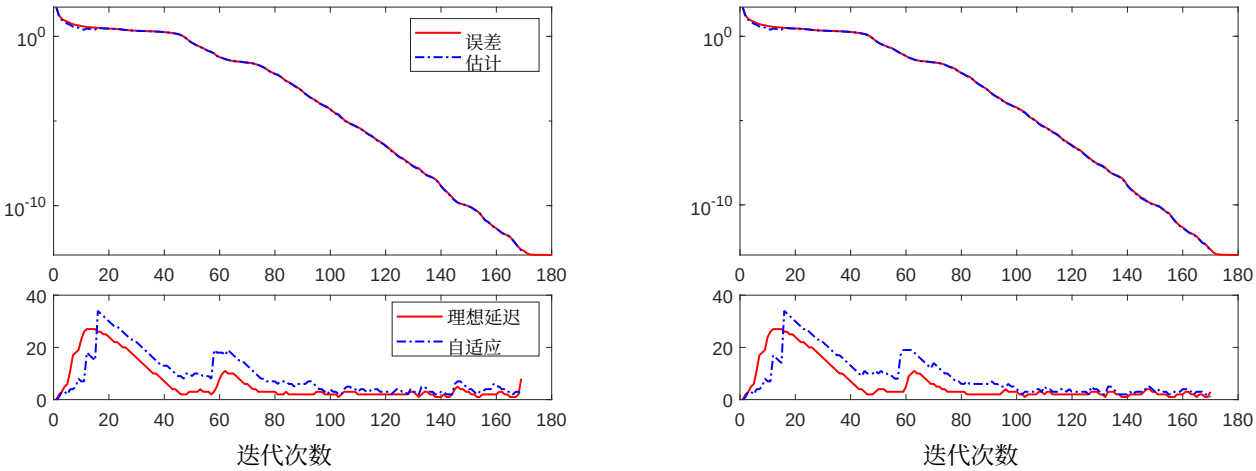


图21：带预条件的矩阵 lp_pilot，PCGNE（左）和 PCRAIG（右）：误差 $\|x - x_k\|$ 和自适应误差估计（上），自适应选择的延迟 $k - \ell$ 及其理想值（下）

6 结论性讨论

本文重点研究了类CG算法求解最小二乘和最小范数问题时误差的准确估计。这些估计仅根据可用的系数计算得出，计算成本非常低。在达到最大可达精度之前，在连续向量之间的局部正交性得以保持这一假设下，这些估计在数值上是可靠的。我们所考虑的连续向量分别对应于底层CG方向向量（迭代索引为 $k-1$）和残差向量（索引为 k）。本文没有详细分析各个算法中局部正交性的保持情况。不过，根据关于CG的既有结果 [32, 33, 23] ，可以预期，对本文讨论的所有算法进行类似分析是可行的，并能得到类似的结果。如有需要，例如为了验证特定应用中估计的可靠性，可以计算局部正交性相关项，通常只需额外进行一次内积运算。

本文旨在获得具有指定相对精度 τ 的误差估计。在当前迭代 k 中，我们估计与某个先前迭代 $\ell \leq k$ 相关的目标量。基于我们先前关于共轭梯度法 [24] 的研究结果，我们制定了一种启发式策略，力求将延迟 $k-\ell$ 缩至最短。数值结果表明，所提出的启发式策略稳健可靠，而且所提出的延迟 $k-\ell$ 往往几乎是最优的。

所提出的方法给出的估计值是目标量的下界。不过，一旦获得具有指定相对精度 τ 的下界 $\Delta_{\ell:k}$，该目标量

$$\frac{\Delta_{\ell:k}}{1-\tau}$$

即为上界；参见 [24, 第 3 节。1] 的讨论。因此，我们也可以轻松获得紧致（但不保证成立）的上界。

总而言之，对于 CGLS 和 LSQR，我们可以获得目标量的紧致估计

$$\|x - x_k\|_{A^TA}^2 = \|r_k\|^2 - \left\|b - b_{|\mathcal{R}(A)}\right\|^2$$

这些估计可用于算法的停止准则。 [7] 和 [19] 中讨论了此类停止准则的一个示例。建议在满足以下条件时停止迭代:

$$\|x - x_k\|_{A^TA}^2 \leq \alpha \|A\| \|x_k\| + \beta \|b\|,$$

其中，$0 \leq \alpha$、$\beta \ll 1$ 是预先给定的容差。在 CGNE 和 CRAIG 算法中，我们能够高效地估计量 $\|x - x_k\|^2$. 。最后，在假设 $x_0 = 0$ 成立的情况下，我们的方法可以直接应用于相对量的估计。

$$\frac{\|x - x_k\|_{A^TA}^2}{\|x\|_{A^TA}^2} \quad \text{and} \quad \frac{\|x - x_k\|^2}{\|x\|^2},$$

分别针对最小二乘问题和最小范数问题；另请参见 [33]。

我们希望本文所呈现的结果能在实际计算中发挥作用。这些结果使我们能够在所考虑算法的迭代过程中，以微不足道的代价近似计算误差，同时考虑预先设定的估计值相对精度。带有误差估计的算法 MATLAB 代码可从 GitHub 仓库 [28] 获取。

致谢： Jan Papež 的研究工作得到了捷克科学院（RVO 67985840）和捷克共和国资助局（资助项目编号：23-06159S）的支持。作者谨向 Gérard Meurant 致以诚挚谢意，感谢他认真审阅稿件并提出宝贵意见，这些意见极大地改善了本文的表述。

参考文献

[1] *HSL*。用于大规模科学计算的 *Fortran* 代码集。*http://www.hsl.rl.ac.uk/*.[2] *M. Arioli*，广义 *Golub–Kahan* 双对角化与停止准则，*SIAM* 矩阵分析与应用杂志，*34*（*2013*），第 *571–592* 页。[3] *M. Arioli* 和 *S. Gratton*，线性回归模型、最小二乘问题、正规方程以及共轭梯度法的停止准则，计算物理通讯，*183*（*2012*），第 *2322–2336* 页。[4] *S. R. Arridge*、*M. M. Betcke* 和 *L. Harhanen*，用于非结构化网格反问题的迭代预条件 *LSQR* 方法，《反问题》，*30*（*2014*），第 *075009*，*27* 页。[5] *Å. Björck*、*T. Elfving* 和 *Z. Strakoš*，线性最小二乘问题中共轭梯度法与 *Lanczos* 方法的稳定性，*SIAM* 矩阵分析与应用杂志，*19*（*1998*），第 *720–736* 页。[6] *R. Bru*、*J. Marín*、*J. Mas* 和 *M. Tůma*，用于求解线性最小二乘问题的预条件迭代法，*SIAM* 科学计算杂志，*36*（*2014*），第 *A2002–A2022* 页。[7] *X.-W. Chang*、*C. C. Paige* 和 *D. Titley-Peloquin*，线性最小二乘问题迭代求解的停止准则，*SIAM* 矩阵分析与应用杂志，*31*（*2009*），第 *831–852* 页。[8] *E. J. Craig*，联立方程的迭代过程，博士论文，麻省理工学院，*1954*。[9] ——, *The* N步迭代过程，《数学与物理杂志》，*34*（*1955*），第 *64–73* 页。[10] *T. A. Davis* 和 *Y. Hu*，佛罗里达大学稀疏矩阵集，*ACM* 数学软件汇刊，*38*（*2011*）。[11] *R. Estrin*、*D. Orban* 和 *M. A. Saunders*，*SYMMLQ* 和共轭梯度法的欧几里得范数误差界，*SIAM* 矩阵分析与应用杂志，*40*（*2019*），第 *235–253* 页。[12] ——,*LNLQ*：一种具有误差最小化性质的最小范数问题迭代方法，*SIAM* 矩阵分析与应用杂志，*40*（*2019*），第 *1102–1124* 页。[13] ——, *LSLQ*：一种具有误差最小化性质的线性最小二乘迭代方法，*SIAM* 矩阵分析与应用杂志，*40*（*2019*），第 *254–275* 页。[14] *D. K. Faddeev* 和 *V. N. Faddeeva*，《线性代数计算方法》，*W. H. Freeman* 出版公司，旧金山，*1963*。[15]*A. Greenbaum* 和 *Z. Strakoš*，二次型公式中的估计，《数值算法》，*8*（*1994*），第 *241–268* 页。[16]*A. Greenbaum*，轻微扰动的 *Lanczos* 与共轭梯度递推的行为，《线性代数及其应用》，*113*（*1989*），第 *7–63* 页。[17] *E. Hallman*，*LSQR* 和共轭梯度法的尖锐 *2*-范数误差界，*SIAM* 矩阵分析与应用杂志，*41*（*2020*），第 *1183–1207* 页。[18] *M. R. Hestenes* 和 *E. Stiefel*，用于求解线性系统的共轭梯度法，《美国国家标准局研究杂志》，*49*（*1952*），第 *409–436* 页。[19] *P. Jiránek* 和 *D. Titley-Peloquin*，*LSQR* 后向误差的估计，*SIAM* 矩阵分析与应用杂志，*31*（*2009/10*），第 *2055–2074* 页。

[20] *E. Kaasschieter*，《用于求解奇异系统的预条件共轭梯度法》，计算与应用数学杂志，*24*（*1988*），第*265–275*页。[21] *J. Liesen* 和 *Z. Strakoš*，《*Krylov* 子空间方法》，《数值数学与科学计算》，牛津大学出版社，牛津，*2013*。[22] *G. Meurant*，《共轭梯度算法中误差的 l_2 范数估计》，《数值算法》，*40*（*2005*），第*157–169*页。[23] *G. Meurant*，《*Lanczos* 与共轭梯度算法：从理论到有限精度计算》，*SIAM*，费城，*2006*。[24] *G. Meurant*、*J. Papež* 和 *P. Tichý*，《共轭梯度法中的精确误差估计》，《数值算法》，*88*（*2021*年），第*1337–1359*页。[25] *G. Meurant* 和 *P. Tichý*，《关于计算共轭梯度法中误差 *A*-范数的基于求积的界》，《数值算法》，*62*（*2013*），第*163–191*页。[26] *C. C. Paige*，《矩阵的双对角化与线性方程的求解》，*SIAM* 数值分析杂志，*11*（*1974*），第*197–209*页。[27] *C. C. Paige* 和 *M. A. Saunders*，《*LSQR*：求解稀疏线性方程和稀疏最小二乘问题的算法》，*ACM* 数学软件汇刊，*8*（*1982*），第*43–71*页。[28] *J. Papež* 和 *P. Tichý*，《带误差估计的类共轭梯度方法》，*GitHub* 仓库。*https://github.com/JanPapez/CGlike-methods-with-error-estimate*，*2023*。[29] *S. Regev* 和 *M. A. Saunders*，《*SSAI*：用于共轭梯度法 *PCG* 和 *PCGLS* 的对称稀疏近似逆预条件子》，*2020*。未发表报告，见 *https://web.stanford.edu/group/SOL/reports/20SSAI.pdf*。[30] *M. A. Saunders*，《使用 *LSQR* 和 *CRAIG* 求解稀疏矩形系统》，*BIT*，*35*（*1995*），第*588–604*页。[31] *G. L. G. Sleijpen*、*H. A. van der Vorst* 和 *D. R. Fokkema*，《*BiCGstab(l)*及其他混合双共轭梯度法》，《数值算法》，*7*（*1994*），第*75–109*页。[32] *Z. Strakoš* 和 *P. Tichý*，《关于共轭梯度法中的误差估计及其在有限精度计算中有效的原因》，电子数值分析汇刊，*13*（*2002*），第*56–80*页。[33] ——，《预条件共轭梯度法中的误差估计》，*BIT*，*45*（*2005*），第*789–817*页。