

Krylov 子空间迭代法笔记

Shun Li

Jul 25, 2025

1 Krylov 子空间迭代方法

主要参考 [矩阵计算/数值线性代数-潘建瑜](#)。

1.1 子空间迭代方法

子空间迭代方法是求解大型（稀疏）线性方程组的一类方法，相比只能求解对称正定问题的共轭梯度法等，子空间迭代方法对方阵没有额外要求，只要方阵非奇异。子空间迭代方法的基本思想是降维：在一个子空间序列中寻找精确解的最优近似，主要过程可以分解为下面的三步：

1. 寻找合适的低维子空间；
2. 在该子空间中求（在某种意义下的）最优近似解；
3. 如果这个最优近似解满足精度要求，计算终止；否则回到第一步，构造一个更大的子空间。

这里存在两个关键问题：

- 如何选择和更新子空间？目前较为成功的，也是普遍采用的方案是 Krylov 子空间。
- 在给定的子空间中，如何定义和求解最优近似解？不同的选择标准，分别对应不同的具体算法，大致分为残量最小化和残量正交化两类。

Remark. 子空间迭代方法在求解大型（稀疏）矩阵的特征值问题中也有广泛的应用，这里不做讨论。

1.2 Krylov 子空间

Definition 1.1 (Krylov 子空间). 设 $A \in \mathbb{R}^{n \times n}$, $r \in \mathbb{R}^n$, 由 A 和 r 生成的第 m 个 Krylov 子空间为：

$$\mathcal{K}_m = \mathcal{K}_m(A, r) = \text{span}\{r, Ar, \dots, A^{m-1}r\} \subset \mathbb{R}^n, \quad m = 1, 2, \dots,$$

Krylov 子空间显然有如下性质

Proposition 1.1.

- Krylov 子空间序列具有嵌套关系： $\mathcal{K}_m \subset \mathcal{K}_{m+1}$

- $\mathcal{K}_m = \{p(A)r : p(x) \in \mathbb{P}^{m-1}(x)\}$, 其中 $\mathbb{P}^{m-1}$ 代表次数不超过 $m-1$ 的多项式。

事实上, 只有前几个 Krylov 子空间是真包含关系, 那么

1. 记 A 的特征多项式为 φ_A , $\deg(\varphi_A) = n$, 由 Cayley-Hamilton 定理可得: $\varphi_A(A) = O$, 因此 $\mathcal{K}_n = \bigcup_{m=1}^{\infty} \mathcal{K}_m$;
2. 记 A 的极小多项式为 d_A , $\deg(d_A) \leq n$, 由 $d_A(A) = 0$ 可得 $\mathcal{K}_{\deg(d_A)} = \bigcup_{m=1}^{\infty} \mathcal{K}_m$;
3. 记 A 关于 r 的极小多项式为 $d_{A,r}$, $\deg(d_{A,r}) \leq \deg(d_A) \leq n$, 由 $d_{A,r}(A)r = 0$ 可得 $\mathcal{K}_{\deg(d_{A,r})} = \bigcup_{m=1}^{\infty} \mathcal{K}_m$ 。

有如下结论:

Proposition 1.2. 对于 Krylov 子空间序列 $\{\mathcal{K}_m = \mathcal{K}_m(A, r), m = 1, 2, \dots\}$:

1. 当且仅当 $m \leq \deg(d_{A,r})$, $\{r, Ar, A^2r, \dots, A^{m-1}r\}$ 线性独立;
2. $\mathcal{K}_{\deg(d_{A,r})}(A, r)$ 是一个 A 不变子空间;
3. $\mathcal{K}_{\deg(d_{A,r})}(A, r)$ 是 Krylov 子空间序列中的最大子空间, 或者称 Krylov 子空间序列在第 $\deg(d_{A,r})$ 项终止;
4. 对于非零向量 r , Krylov 子空间的维度为

$$\dim(\mathcal{K}_m) = \text{rank}\begin{pmatrix} r & Ar & \dots & A^{m-1}r \end{pmatrix} = \min(m, \deg(d_{A,r}))$$

5. 前几个 Krylov 子空间之间是真包含关系, 后续的 Krylov 子空间都是同一个空间

$$\mathcal{K}_1 \subsetneq \mathcal{K}_2 \subsetneq \dots \subsetneq \mathcal{K}_{\deg(d_{A,r})} = \mathcal{K}_{\deg(d_{A,r}+1)} = \dots$$

Remark. Krylov 子空间序列最终充满整个 $\mathbb{R}^n$ 空间 $\iff \deg(d_{A,r}) = n$, 易知 $d_{A,r} | d_A$ 以及 $d_A | \varphi_A$, 因此这个要求非常强: 首先, A 需要保证 $d_A = \varphi_A$, 这等价于 A 的 Jordan 分解的几何重数均为 1, 即每个特征值只能对应一个 Jordan 块; 其次, r 还需要保证 $d_{A,r} = d_A$ 。

Proposition 1.3. 对于方阵 $A \in \mathbb{R}^{n \times n}$, 记特征多项式为 φ_A , 极小多项式为 d_A , 那么

$$A \text{ 可逆} \iff \varphi_A(0) \neq 0 \iff d_A(0) \neq 0$$

Proof. 显然, 因为两边都等价于 0 不是 A 的特征值。 □

一般的化零多项式可以给出可逆性的充分条件。

Proposition 1.4. 对于方阵 $A \in \mathbb{R}^{n \times n}$, 记 A 的一个化零多项式为

$$d(x) = \sum_{i=0}^m c_i x^i,$$

那么

$$c_0 \neq 0 \implies A \text{ 可逆}$$

在 $c_0 \neq 0$ 时, 可以给出 A^{-1} 的具体构造

$$A^{-1} = - \sum_{i=1}^m \frac{c_i}{c_0} A^{i-1}$$

由此可以得到如下推论:

Corollary 1.5. 设 A 可逆, 对于线性方程组问题 $Ax = b$, 那么

$$x = A^{-1}b \in \mathcal{K}_m(A, b), \quad m = \deg(d_{A,b})$$

也可以充分利用初值的信息, 求解相对于初值的偏移量问题:

Corollary 1.6. 设 A 可逆, 对于线性方程组问题 $Ax = b$, 任取初值 x_0 , 对应的残量为 $r_0 = b - Ax_0$, 考虑偏移量 $y = x - x_0$ 满足的线性方程组

$$Ay = b - Ax_0 = r_0$$

那么

$$y = A^{-1}r_0 \in \mathcal{K}_m(A, r_0), \quad m = \deg(d_{A,r_0})$$

Remark. 这两个推论为下文的 Krylov 子空间迭代法提供了理论保证, 在实际的计算过程中还需要考虑数值误差的影响。

Remark. 本文主要考虑基于初值残量生成的 Krylov 子空间 $\mathcal{K}_m = \mathcal{K}_m(A, r_0)$, 其中 $r_0 = b - Ax_0$ 。有的资料中主要考虑 $\mathcal{K}_m(A, b)$, 可以将其视作使用零向量作为初值的情形。

Remark. 对于某些问题 (计算更一般的 $f(A)b$) 也会考虑扩展的 Krylov 子空间 (extended Krylov Subspace)

$$\mathcal{K}_m^{ext} = \mathcal{K}_m(A, r_0) + \mathcal{K}_m(A^{-1}, r_0).$$

1.3 正交基构造

1.3.1 Arnoldi 过程

在使用 Krylov 子空间的前提下, 我们还需要解决如下问题:

1. 寻找 $\mathcal{K}_m$ 的一组合适的基 $v_1, v_2, \dots, v_m$: (最好是单位正交基)
2. 求出最优近似在这组基下的系数表示。

显然 $\mathcal{K}_m$ 有一组最简单的基: $\{r, Ar, A^2r, \dots, A^{m-1}r\}$, 但是这不是正交基, 稳定性得不到保证。可以通过 Gram-Schmidt 正交化获得对应的一组正交基, 这就是 Arnoldi 过程 (算法 1)。

Remark. 数值实验表明, 这些基的正交化在实际计算中可能很快就会损失掉, 可以使用重正交化技术避免, 也就是再重复执行一次 Gram-Schmidt 正交化。

下面推导 Arnoldi 过程的矩阵表示, 暂时不考虑 Arnoldi 过程的提前中断情况。

$$Av_j - h_{1,j}v_1 - \dots - h_{j,j}v_j = h_{j+1,j}v_{j+1}$$

Algorithm 1: Arnoldi 过程 (MGS)**Input:** 方阵 $A \in \mathbb{R}^{n \times n}$, 向量 $r \in \mathbb{R}^n$, $m \in \mathbb{N}$ **Output:** 单位正交基 $V_{m+1} = [v_1, \dots, v_{m+1}] \in \mathbb{R}^{n \times (m+1)}$, Hessenberg 矩阵

$$H_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$$

```

1  $v_1 \leftarrow r / \|r\|$ ;
2 for  $j = 1$  to  $m$  do
3    $w \leftarrow Av_j$ ;
4   for  $i = 1$  to  $j$  do
5      $h_{i,j} \leftarrow (v_i, w)$ ;  $w \leftarrow w - h_{i,j}v_i$ ;
6   end
7    $h_{j+1,j} \leftarrow \|w\|$ ;
8   if  $h_{j+1,j} = 0$  then
9     break;
10  end
11   $v_{j+1} \leftarrow w / h_{j+1,j}$ ;
12 end

```

即

$$Av_j = \begin{bmatrix} v_1 & \cdots & v_{j+1} \end{bmatrix} \begin{bmatrix} h_{1,j} \\ \vdots \\ h_{j+1,j} \end{bmatrix} = \begin{bmatrix} v_1 & \cdots & v_{m+1} \end{bmatrix} \begin{bmatrix} h_{1,j} \\ \vdots \\ h_{j+1,j} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = V_{m+1} H_{m+1,m}(:, j)$$

其中

$$V_{m+1} = \begin{bmatrix} v_1 & v_2 & \cdots & v_m & v_{m+1} \end{bmatrix} \in \mathbb{R}^{n \times (m+1)}$$

$$H_{m+1,m} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,m} \\ & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix} \in \mathbb{R}^{(m+1) \times m}$$

因此

$$AV_m = V_{m+1} H_{m+1,m}$$

为了记号的方便, 定义

$$H_m = H_{m+1,m}(1:m, 1:m) \in \mathbb{R}^{m \times m}$$

则有

$$V_{m+1} H_{m+1,m} = \begin{bmatrix} V_m & v_{m+1} \end{bmatrix} \begin{bmatrix} H_m \\ h_{m+1,m} e_m^\top \end{bmatrix} = V_m H_m + h_{m+1,m} v_{m+1} e_m^\top$$

其中 $e_m = [0, \dots, 0, 1]^T \in \mathbb{R}^m$, 因此整理得到

$$AV_m = V_{m+1}H_{m+1,m} = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \quad (1)$$

对上式左乘 V_m^T 可得 (注意到 $V_m^T V_m = I$ 以及 $V_m^T v_{m+1} = 0$)

$$V_m^T A V_m = H_m \quad (2)$$

公式 1 和 2 是 Arnoldi 过程的两个重要性质。

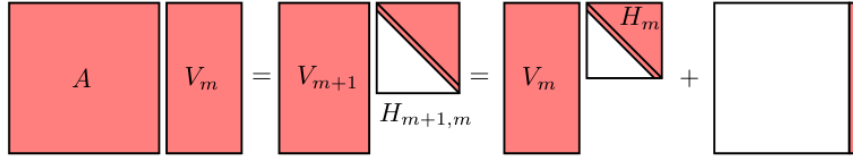


图 1: 公式 1 的示意图

如果 Arnoldi 过程提前中断, 不妨记作 $h_{k+1,k} = 0$, 则有

$$AV_k = V_k H_k$$

这表明 $AK_k \subset \mathcal{K}_k$, 即 $\mathcal{K}_k$ 是一个 A 不变子空间, 此时已经达到了最大的 Krylov 子空间。Arnoldi 过程的提前中断在数值求解算法中是一个非常受欢迎的情况, 考虑到数值误差的存在, 实际的判断可能是 $|h_{k+1,k}| < \varepsilon$ 足够小, 也可能直接忽略提前中断的情况。

Remark. Arnoldi 过程最早用于经典 Arnoldi 算法中, 被用来近似求解大型 (稀疏非对称) 方阵的少数几个特征对: 使用 Arnoldi 过程得到 H_m 满足等式

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T$$

求解 H_m 的特征对 (μ, y) 满足 $H_m y = \mu y$, 那么

$$AV_m y = V_m H_m y + h_{m+1,m} v_{m+1} e_m^T y$$

因此

$$\|A(V_m y) - \mu(V_m y)\| = |h_{m+1,m}| |v_{m+1} e_m^T y|$$

如果右端项足够小, 那么 $(\mu, V_m y)$ 就可以作为 A 的近似特征对。

1.3.2 Lanczos 过程

如果 A 是对称方阵, 那么上述 Arnoldi 过程具有更特殊的形式, 又称为 Lanczos 过程 (算法 2)。

Proposition 1.7. 若 A 为对称方阵, 那么 H_m 为对称的三对角方阵, 此时习惯将 H_m 记作

T_m

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

对应的公式为

$$AV_m = V_m T_m + \beta_m v_{m+1} e_m^T \quad (3)$$

正交化步骤具有三项递推公式形式（补充 $v_0 = 0$ 和 $\beta_0 = 0$ ）

$$Av_j - \beta_{j-1} v_{j-1} - \alpha_j v_j = \beta_j v_{j+1}$$

Proof. Arnoldi 过程保证 $T_m = H_m$ 是一个 Hessenberg 方阵，由于 $T_m = V_m^T A V_m$ 仍然是一个对称方阵，因此它是一个对称的三对角方阵。 $\square$

Algorithm 2: Lanczos 过程

Input: 对称方阵 $A \in \mathbb{R}^{n \times n}$, 向量 $r \in \mathbb{R}^n$, $m \in \mathbb{N}$ **Output:** 单位正交基 $V_{m+1} = [v_1, \dots, v_{m+1}] \in \mathbb{R}^{n \times (m+1)}$, Hessenberg 矩阵

$$H_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$$

```

1  $v_0 \leftarrow 0, \beta_0 \leftarrow 0;$ 
2  $v_1 \leftarrow r / \|r\|;$ 
3 for  $j = 1$  to  $m$  do
4    $w \leftarrow Av_j;$ 
5    $\alpha_j \leftarrow (v_j, w);$ 
6    $w \leftarrow w - \beta_{j-1} v_{j-1} - \alpha_j v_j;$ 
7    $\beta_j \leftarrow \|w\|;$ 
8   if  $\beta_j = 0$  then
9     break;
10  end
11   $v_{j+1} \leftarrow w / \beta_j;$ 
12 end

```

Remark. 由于 $T_m = V_m^T A V_m$ ，如果 A 对称正定，那么 T_m 显然也是对称正定的。通常 V_m 不是方阵，因此这个结论反过来不成立。

1.4 最优近似

接下来的工作是寻找在给定的 Krylov 子空间下的最优近似，为了充分利用迭代初值信息，我们在如下仿射空间中寻找最优近似。

$$x_0 + \mathcal{K}_m = \{x_0 + z : z \in \mathcal{K}_m\}$$

Krylov 子空间算法的基本框架如下（算法 3）：对于线性方程组 $Ax = b$,

1. 令 $m = 1$, 选取迭代初值 x_0 , 计算初值对应的残量 $r_0 = b - Ax_0$;
2. 定义 Krylov 子空间 $\mathcal{K}_m = \mathcal{K}_m(A, r_0)$;
3. 在仿射空间 $x_0 + \mathcal{K}_m$ 寻找原问题的最优近似解, 记作 x_m ;
4. 如果满足迭代终止条件, 则返回 x_m ; 否则令 $m = m + 1$, 回到第二步。

关于上述流程的补充说明:

1. 在第二步通常会附带 $\mathcal{K}_m$ 的正交基构造, 由于 Krylov 子空间具有嵌套关系, 只需要计算 v_m 即可;
2. 在第三步中通常不需要显式地解出 x_m 就可以判定是否达到精度要求 (通常是对残量的要求), 只需要在迭代退出时解出对应的 x_m 即可。

Remark. 从计算效率的角度考虑, 我们并不需要在每次增大 Krylov 子空间后求解最优近似, 而是可以在生成一个足够大的 Krylov 子空间 $\mathcal{K}_m$ 之后求解即可, 也可以在每次增大 Krylov 子空间后计算残量 r_m , 仅在残量足够小时终止并求解 x_m 。

Algorithm 3: Krylov 子空间算法的基本框架

Input: 方阵 $A \in \mathbb{R}^{n \times n}$, 右端向量 $b \in \mathbb{R}^n$, 迭代初值 $x_0 \in \mathbb{R}^n$

Output: 最优近似解 $\tilde{x}$

```

1  $r_0 \leftarrow b - Ax_0$ ;
2  $v_1 \leftarrow r_0 / \|r_0\|$ ;
3 求解最优近似解  $x_1 \in x_0 + \mathcal{K}_1 = x_0 + \text{span}(v_1)$ ;
4 if  $x_1$  满足精度要求 then
5    $\tilde{x} \leftarrow x_1$ ; break;
6 end
7 for  $m = 2$  to  $n$  do
8   调用 Arnoldi 过程 (算法 1) 或 Lanczos 过程 (算法 2) 计算基向量  $v_m$ ;
9   求解最优近似解  $x_m \in x_0 + \mathcal{K}_m = x_0 + \text{span}(v_1, \dots, v_m)$ ;
10  if  $x_m$  满足精度要求 then
11    $\tilde{x} \leftarrow x_m$ ; break;
12  end
13 end

```

为了完成算法, 我们必须给出“最优近似解”的严格定义, 不同的定义会导出不同的算法。记精确解为 x_* , 一个最简单直接的想法是误差最小化 $\|x_m - x_*\|$, 但是这通常难以实现, 因为 x_* 未知。常见的做法包括残量最小化和残量正交化两类:

1. 残量最小化 $\|r_m\| = \|b - Ax_m\|$
 - (a) (A 对称不定) 极小残量法 (MINRES)
 - (b) (A 非对称) 广义极小残量法 (GMRES)
2. 残量正交化 $r_m = b - Ax_m \perp \mathcal{K}_m$

- (a) (A 对称正定) 共轭梯度法 (CG)
- (b) (A 对称不定) SYMMLQ 方法
- (c) (A 非对称) 完全正交方法 (FOM)

其中 GMRES 算法是 MINRES 算法在非对称情形下的推广, 共轭梯度法可以同时视作采用多种最优近似解的定义: (相互等价)

1. 残量正交化 $r_m = b - Ax_m \perp \mathcal{K}_m$;
2. 误差加权最小化 $\|x_m - x_*\|_A$;
3. 残量加权最小化 $\|r_m\|_{A^{-1}} = \|b - Ax_m\|_{A^{-1}}$;

这里 A 是对称正定矩阵可以保证范数 $\|\cdot\|_A$ 和 $\|\cdot\|_{A^{-1}}$ 都是良定义的:

$$\|v\|_A = \sqrt{v^T A v}, \quad \|v\|_{A^{-1}} = \sqrt{v^T A^{-1} v}.$$

Proposition 1.8. 残量最小化条件 $\|r_m\| = \|b - Ax_m\|$ 等价于如下的正交化条件

$$r_m = b - Ax_m \perp AK_m$$

其中 $AK_m = \{Az : z \in \mathcal{K}_m\}$ 。

Proof. 设 $\mathcal{K}_m$ 的单位正交基为 V_m , 定义凸泛函

$$\varphi(y) = \|b - A(x_0 + V_m y)\|^2$$

φ 的唯一极小值点 y_m 所满足的条件为

$$V_m^T A^T (b - A(x_0 + V_m y)) = 0$$

易知 $x_m = x_0 + V_m y_m$ 满足残量最小化条件, 同时也满足

$$V_m^T A^T (b - Ax_m) = 0$$

□

1.5 投影法

上一节所讨论的最优近似可以从投影法的角度重写。对于线性方程组问题 $Ax = b$, 其中 $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, 初值 $x_0 \in \mathbb{R}^n$, 投影方法的一般模型为:

$$\text{find } \tilde{x} \in x_0 + \mathcal{K}, \quad \text{s.t.} \quad r = b - A\tilde{x} \perp \mathcal{L}$$

其中子空间 $\mathcal{K}$ 称为搜索空间, 子空间 $\mathcal{L}$ 称为约束空间。可以简单分类:

1. 若 $\mathcal{K} = \mathcal{L}$, 上述约束称为 Galerkin 条件, 对应正交投影;
2. 若 $\mathcal{K} \neq \mathcal{L}$, 上述约束称为 Petrov-Galerkin 条件, 对应斜投影。

为了保证解的存在唯一性，显然要求

$$\dim(\mathcal{K}) = \dim(\mathcal{L}) = m$$

通常 $m \ll n$ 。如果记 $\mathcal{K}$ 和 $\mathcal{L}$ 的一组基分别为

$$V = \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix}, \quad W = \begin{bmatrix} w_1 & \cdots & w_m \end{bmatrix}$$

那么 $\tilde{x}$ 可以表示为

$$\tilde{x} = x_0 + Vy, \quad y \in \mathbb{R}^m$$

由正交性条件可得（记初值的残量 $r_0 = b - Ax_0$ ）

$$0 = W^T(b - A(x_0 + Vy)) = W^T(r_0 - AVy)$$

若 W^TAV 非奇异，可以解得

$$\begin{aligned} y &= (W^TAV)^{-1}W^Tr_0 \\ \tilde{x} &= x_0 + V(W^TAV)^{-1}W^Tr_0 \end{aligned}$$

注意这只是理论上的公式，在实际计算中并不会采用上述表达式。关于 W^TAV 非奇异的条件，有如下定理可以保证：

Proposition 1.9. 若 A , $\mathcal{K}$ 和 $\mathcal{L}$ 满足下面的条件之一

1. A 对称正定，并且 $\mathcal{L} = \mathcal{K}$;
2. A 非奇异，并且 $\mathcal{L} = A\mathcal{K}$

可以保证 W^TAV 非奇异。

Remark.

1. 取 $\mathcal{L} = \mathcal{K}$ 对应残量正交化，代表方法包括：FOM，CG；
2. 取 $\mathcal{L} = A\mathcal{K}$ 对应残量最小化，代表方法包括：MINRES，GMRES。

1.6 代表算法

1.6.1 完全正交方法（FOM）

完全正交方法（FOM）是基于残量正交化得到的非对称线性方程组的求解算法，基于之前的 Krylov 子空间算法框架，我们只需要补充使用残量正交化作为最优近似标准的计算细节。

$$r_m = b - Ax_m \perp \mathcal{K}_m$$

记 $x = x_0 + V_my \in x_0 + \mathcal{K}_m$ ，其中 $y \in \mathbb{R}^m$ ，那么

$$0 = V_m^Tr = V_m^T(b - Ax) = V_m^Tb - V_m^TA(x_0 + V_my) = V_m^Tr_0 - V_m^TAV_my$$

利用等式 2 和 $\beta = \|r_0\|$ ， $v_1 = r_0/\|r_0\|$ 可得

$$0 = V_m^Tr_0 - V_m^TAV_my = V_m^T\beta v_1 - H_my = \beta e_1 - H_my$$

如果 H_m 非奇异, 可以解得

$$y_m = H_m^{-1}\beta e_1, \quad x_m = x_0 + \beta V_m H_m^{-1} e_1$$

需要计算残量作为算法收敛的判据

$$\begin{aligned} r_m &= b - Ax_m = r_0 - AV_m y_m \\ &= (\beta v_1 - V_m H_m y_m) - h_{m+1,m}(e_m^T y_m)v_{m+1} \\ &= -h_{m+1,m}(e_m^T y_m)v_{m+1} \\ \|r_m\| &= h_{m+1,m}|e_m^T y_m| = h_{m+1,m}|y_m(m)| \end{aligned}$$

最终可以得到 FOM 算法 4。以下假设 A 非奇异; 若中途的 H_j 奇异, 则 FOM 当前投影方程可能无解, 需要报告中断或改用 GMRES。

Algorithm 4: FOM 算法

Input: 方阵 $A \in \mathbb{R}^{n \times n}$, 右端向量 $b \in \mathbb{R}^n$, 迭代初值 $x_0 \in \mathbb{R}^n$, 精度要求 ε (控制相对残量)

Output: 最优近似解 $\tilde{x}$

```

1  $r_0 \leftarrow b - Ax_0$ ;
2  $\beta \leftarrow \|r_0\|$ ;
3 if  $\beta = 0$  then
4   输出  $\tilde{x} = x_0$  并结束;
5 end
6  $v_1 \leftarrow r_0/\beta$ ;
7 for  $j = 1$  to  $\min(\text{IterMax}, n)$  do
8    $w_j \leftarrow Av_j$ ;
9   for  $i = 1$  to  $j$  do
10     $h_{i,j} \leftarrow (v_i, w_j)$ ;  $w_j \leftarrow w_j - h_{i,j}v_i$ ;
11  end
12   $h_{j+1,j} \leftarrow \|w_j\|$ ;
13  if  $H_j$  奇异 then
14    报告 FOM 中断并结束;
15  end
16  求解线性方程组  $H_j y_j = \beta e_1$ , 令  $m \leftarrow j$ ;
17  if  $h_{j+1,j}|y_j(j)| \leq \varepsilon\beta$  then
18    break;
19  end
20  if  $j = \min(\text{IterMax}, n)$  then
21    break;
22  end
23   $v_{j+1} \leftarrow w_j/h_{j+1,j}$ ;
24 end
25  $\tilde{x} \leftarrow x_0 + V_m y_m$ ;

```

1.6.2 广义极小残量法 (GMRES)

广义极小残量法 (GMRES) 是基于残量最小化得到的非对称线性方程组的求解算法, 也是目前求解非对称线性方程组的最常用算法。基于之前的 Krylov 子空间算法框架, 我们只需要补充使用残量最小化作为最优近似标准的计算细节。

$$x_m = \arg \min_{x \in x_0 + \mathcal{K}_m} \|b - Ax\| = x_0 + \arg \min_{x \in \mathcal{K}_m} \|r_0 - Ax\|$$

记 $x = x_0 + V_m y \in x_0 + \mathcal{K}_m$, 其中 $y \in \mathbb{R}^m$, 那么

$$r = b - Ax = b - A(x_0 + V_m y) = r_0 - AV_m y$$

利用等式 1 可得

$$\begin{aligned} r &= r_0 - AV_m y = V_{m+1} \beta e_1 - V_{m+1} H_{m+1,m} y, \quad (\beta = \|r_0\|) \\ &= V_{m+1} (\beta e_1 - H_{m+1,m} y) \end{aligned}$$

利用 V_{m+1} 的列向量单位正交可得

$$\|r\| = \|V_{m+1} (\beta e_1 - H_{m+1,m} y)\| = \|\beta e_1 - H_{m+1,m} y\|$$

因此转化为求解如下最优化问题

$$y_m = \arg \min_{y \in \mathbb{R}^m} \|\beta e_1 - H_{m+1,m} y\|, \quad x_m = x_0 + V_m y_m.$$

由于 $m \ll n$, 这个问题的规模较小, 可以使用 QR 分解算法进行求解: 设 $H_{m+1,m}$ 的 QR 分解如下

$$Q_{m+1}^T H_{m+1,m} = \begin{bmatrix} R_m \\ 0 \end{bmatrix}$$

其中 $Q_{m+1} \in \mathbb{R}^{(m+1) \times (m+1)}$ 为正交方阵, $R_m \in \mathbb{R}^{m \times m}$ 为上三角方阵。假设 A 非奇异, 则 Arnoldi 过程得到的 $H_{m+1,m}$ 列满秩, 因此 R_m 可逆。记 $g^{(m)} = Q_{m+1}^T e_1$ 。代入得到

$$\|\beta e_1 - H_{m+1,m} y\| = \|\beta Q_{m+1}^T e_1 - \begin{bmatrix} R_m \\ 0 \end{bmatrix} y\| = \|\beta g^{(m)} - \begin{bmatrix} R_m \\ 0 \end{bmatrix} y\|$$

因此

$$y_m = \beta R_m^{-1} g^{(m)}(1:m) \in \mathbb{R}^m$$

对应的残量为

$$\|r_m\| = \|\beta e_1 - H_{m+1,m} y_m\| = \beta |g^{(m)}(m+1)|, \quad \frac{\|r_m\|}{\|r_0\|} = |g^{(m)}(m+1)|$$

注意残量的计算并不需要提前解出 y_m 和 x_m 。最终可以得到 GMRES 算法 5, 关于如何高效可靠地对 $H_{m+1,m}$ 进行 QR 分解的具体细节, 这里略去。

对于非奇异 n 阶矩阵 A , 完整、未重启的 GMRES 在精确算术下至多 n 步得到精确解, 不要求 A 正规或可相似对角化。若 $h_{j+1,j} = 0$, Krylov 子空间已不再增长, 此时应直接求解当前最小二乘问题并终止, 无须构造 v_{j+1} 。某些矩阵会使残量在很多步内停滞, 重启 GMRES 也可能不收敛; 有限精度计算还会受到舍入误差的影响。达到迭代上限时, 输出的是当前近似解, 应另外检查残量是否满足精度要求。

Algorithm 5: GMRES 算法

Input: 非奇异方阵 $A \in \mathbb{R}^{n \times n}$, 右端向量 $b \in \mathbb{R}^n$, 迭代初值 $x_0 \in \mathbb{R}^n$, 精度要求 ε (控制相对残量)

Output: 最优近似解 $\tilde{x}$

```

1  $r_0 \leftarrow b - Ax_0$ ;
2  $\beta \leftarrow \|r_0\|$ ;
3 if  $\beta = 0$  then
4   输出  $\tilde{x} = x_0$  并结束;
5 end
6  $v_1 \leftarrow r_0/\beta$ ;
7 for  $j = 1$  to  $\min(\text{IterMax}, n)$  do
8    $w_j \leftarrow Av_j$ ;
9   for  $i = 1$  to  $j$  do
10     $h_{i,j} \leftarrow (v_i, w_j)$ ;  $w_j \leftarrow w_j - h_{i,j}v_i$ ;
11  end
12   $h_{j+1,j} \leftarrow \|w_j\|$ ;
13  对  $H_{j+1,j}$  进行 QR 分解, 获取  $Q_{j+1}$  和  $R_j$ ;
14   $g^{(j)} \leftarrow Q_{j+1}^\top e_1$ ,  $m \leftarrow j$ ;
15  if  $h_{j+1,j} = 0$  或  $|g^{(j)}(j+1)| \leq \varepsilon$  then
16    break;
17  end
18  if  $j = \min(\text{IterMax}, n)$  then
19    break;
20  end
21   $v_{j+1} \leftarrow w_j/h_{j+1,j}$ ;
22 end
23 求解上三角线性方程组  $R_m y_m = \beta g^{(m)}(1:m)$ ;
24  $\tilde{x} \leftarrow x_0 + V_m y_m$ ;

```

Remark. 由于迭代次数的增加会导致运算量和存储量迅速增加, 一个简单的优化策略是重启动: 如果迭代固定步数后仍然没有达到精度要求, 将其作为新的迭代初值, 放弃之前计算的正交基, 重新生成 Krylov 子空间进行计算。

Remark. GMRES 算法针对的是非对称线性方程组, 如果矩阵 A 是对称的, 那么 Arnoldi 过程就会变成 Lanczos 过程, 此时可以利用三项递推公式简化计算和存储, 这样就得到了针对对称方程组问题的 MINRES 算法。

1.6.3 共轭梯度法 (CG)

对于对称正定方程组的求解, 使用如下的误差加权最小化作为最优近似标准, 就可以得到著名的共轭梯度法 (CG)

$$x_m = \arg \min_{x \in x_0 + \mathcal{K}_m} \|x - x_*\|_A = x_0 + \arg \min_{z \in \mathcal{K}_m} \|x_0 + z - x_*\|_A$$

值得注意的是, 即使精确解 x_* 未知, 上述最优化问题仍然可以求解, 有如下命题成立:

Proposition 1.10. 设 A 对称正定, 那么下面三个表述等价:

1. $x_m \in x_0 + \mathcal{K}_m$ 满足

$$\|x_m - x_*\|_A = \min_{x \in x_0 + \mathcal{K}_m} \|x - x_*\|_A$$

2. $x_m \in x_0 + \mathcal{K}_m$ 满足

$$\|b - Ax_m\|_{A^{-1}} = \min_{x \in x_0 + \mathcal{K}_m} \|b - Ax\|_{A^{-1}}$$

3. $x_m \in x_0 + \mathcal{K}_m$ 满足

$$b - Ax_m \perp \mathcal{K}_m$$

Proof. 注意到

$$\begin{aligned} \|b - Ax\|_{A^{-1}}^2 &= (b - Ax)^\top A^{-1} (b - Ax) \\ &= (A^{-1}b - x)^\top A^\top A^{-1} A (A^{-1}b - x) \\ &= (x_* - x)^\top A (x_* - x) \\ &= \|x - x_*\|_A^2 \end{aligned}$$

因此条件 1 和条件 2 等价。条件 2 等价于 $y = V_m^\top (x_m - x_0)$ 是如下凸泛函的唯一极小值点

$$\varphi(y) = \|b - A(x_0 + V_m y)\|_{A^{-1}}^2, \quad y \in \mathbb{R}^n$$

展开得到

$$\varphi(y) = y^\top V_m^\top A V_m y - 2(b - Ax_0)^\top V_m y + (b - Ax_0)^\top A^{-1} (b - Ax_0)$$

因此极值点满足

$$2V_m^\top (A V_m y - (b - Ax_0)) = 2V_m^\top (Ax_m - b) = 0$$

因此与条件 3 等价。 □

对任意向量 $x = x_0 + V_m y \in x_0 + \mathcal{K}_m$, 其中 $y \in \mathbb{R}^m$, 由正交性条件可得

$$b - Ax \perp \mathcal{K}_m \iff V_m^T(b - Ax) = 0 \iff V_m^T(r_0 - AV_m y) = 0$$

利用等式 3 可得 (记 $\beta = \|r_0\|$)

$$0 = V_m^T(r_0 - AV_m y) = V_m^T V_{m+1} \beta e_1 - V_m^T V_{m+1} T_m y = \beta e_1 - T_m y$$

因此

$$\begin{aligned} y_m &= T_m^{-1}(\beta e_1), \\ x_m &= x_0 + V_m y_m = x_0 + V_m T_m^{-1}(\beta e_1). \end{aligned}$$

由于 T_m 是对称正定的三对角方阵, 可以使用快速分解算法进行求逆: 记 T_m 的 LDL^T 分解为 $T_m = L_m D_m L_m^T$, 其中 L_m 是单位下三角方阵, D_m 是对角阵, 那么

$$T_m^{-1} = L_m^{-T} D_m^{-1} L_m^{-1}.$$

代入得到

$$x_m = x_0 + V_m T_m^{-1}(\beta e_1) = x_0 + V_m L_m^{-T} D_m^{-1} L_m^{-1}(\beta e_1)$$

利用 $\{T_m\}$ 的特殊结构以及递推关系可以导出 $\{L_m\}$ 和 $\{D_m\}$ 的递推关系, 进一步简化计算量, 得到通常形式下的共轭梯度法 6, 推导比较繁琐, 这里略去。

Remark. CG 方法是设计用于求解对称正定方程组的算法, 对于对称不定问题, CG 方法也可以用于实际求解, 但是求解过程中可能产生中断。对于对称不定方程组的求解还可以使用 SYMMLQ 算法, 它在 A 对称正定时与 CG 等价。

Remark. 对于非对称问题 $Ax = b$, 也可以将其直接转换为正规方程

$$A^T A x = A^T b,$$

然后就可以利用 CG 方法进行求解。当然, 直接计算 $A^T A$ 显然是不合适的, 可以稍加修改进行避免, 这就得到了 CGLS 方法 (CG for Least Squares problem)。除此之外, 还有一个 LSQR 方法, 在不考虑数值误差的情况下, LSQR 方法与 CGLS 方法等价, 但是对于条件数很大的问题, LSQR 方法的数值表现更好。

1.7 预条件子

大规模病态线性方程组的数值求解会遇到困难: 首先是对精度的影响很大, 即使在不考虑精度的情况下, 也会导致迭代法收敛速度很慢。预条件方法是目前普遍采用的优化病态矩阵数值求解的做法。按照预条件子的位置, 可以分成左预条件, 右预条件和双边预条件三类, 三者的效果通常并不等价, 需要根据问题本身和所用的方法来选择。下面以最一般性的双边预条件为例进行讨论。

考虑线性方程组问题

$$Ax = b,$$

其中 $A \in \mathbb{R}^{n \times n}$ 非奇异, $b \in \mathbb{R}^n$ 。选取两个可逆矩阵 $L \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{n \times n}$, 可以得到新方程组

$$L^{-1} A R^{-1} y = L^{-1} b. \quad x = R^{-1} y.$$

这里的矩阵 L, R 称为预条件子, 在不考虑数值求解误差的情况下, 预条件后的方程组显然与原方程组等价, 但是两者的数值求解并不相同。取 $L = I$ 或 $R = I$ 则退化为单边的预条件。对于预条件子的选取需要考虑:

Algorithm 6: 共轭梯度法

Input: 对称正定方阵 $A \in \mathbb{R}^{n \times n}$, 右端向量 $b \in \mathbb{R}^n$, 迭代初值 $x_0 \in \mathbb{R}^n$, 精度要求 ε (控制相对残量)

Output: 最优近似解 $\tilde{x}$

```

1  $r_0 \leftarrow b - Ax_0$ ;
2  $\beta \leftarrow \|r_0\|$ ;
3 if  $\beta = 0$  then
4   输出  $\tilde{x} = x_0$  并结束;
5 end
6 for  $k = 1$  to  $IterMax$  do
7    $\rho \leftarrow r_{k-1}^\top r_{k-1}$ 
8   if  $k > 1$  then
9      $\mu_{k-1} \leftarrow \rho / \rho_0$ ,  $p_k \leftarrow r_{k-1} + \mu_{k-1} p_{k-1}$ ;
10  else
11     $p_k = r_0$ ;
12  end
13   $q_k \leftarrow Ap_k$ ,  $\xi_k \leftarrow \rho / (p_k^\top q_k)$ ;
14   $x_k \leftarrow x_{k-1} + \xi_k p_k$ ,  $r_k \leftarrow r_{k-1} - \xi_k q_k$ ;
15  if  $\|r_k\| \leq \varepsilon \beta$  then
16    break;
17  end
18   $\rho_0 \leftarrow \rho$ ;
19 end
20  $\tilde{x} \leftarrow x_k$ ;

```

- 预条件子需要优化数值求解: 将 $Ax = b$ 转换为 $L^{-1}AR^{-1}y = L^{-1}b$ 进行求解, 目标是 $L^{-1}AR^{-1}$ 相比 A 具有更小的条件数。对于特征值问题来说, 目标则是 $L^{-1}AR^{-1}$ 相比 A 具有更好的特征值分布。
- 预条件子 L, R 的计算成本低: 我们需要显式构造 L, R , 不需要显式计算 L^{-1} 和 R^{-1} , 但是需要保证线性方程组 $Lz = r$ 和 $Rz = r$ 都是容易求解的, 这个要求在下面的讨论中会自然体现。

直接将 Krylov 子空间方法应用于求解预条件后的新方程组, 涉及到的计算包括 (记 $\tilde{A} = L^{-1}AR^{-1}$, $\tilde{b} = L^{-1}b$)

1. 右端项 $\tilde{b} = L^{-1}b$ 需要求解方程组

$$L\tilde{b} = b, \rightarrow \tilde{b}$$

2. 在 Krylov 子空间中, 显式计算 $v = \tilde{A}u = L^{-1}AR^{-1}u$ 的过程实际为求解两个方程组:

$$Rw = u, \rightarrow w,$$

$$Lv = Aw, \rightarrow v$$

因为引入了很多 $Lz = r$ 和 $Rz = r$ 类型的方程组求解, 预条件子的选择必须保证它们的计算都是高效的。

Remark. 一个有趣的事实是, 如果矩阵 A 是病态的, 如果我们希望 $P^{-1}A$ 或 AP^{-1} 的条件数大大降低, 通常也需要选取一个病态的预条件子 P 。

Remark. 在迭代法中, 我们既可以固定预条件子, 也可以在迭代过程中使用不同的预条件子, 例如 Flexible GMRES 方法。

Remark. 以左预条件为例, 一个好的预条件子 P 应该是 A 在某种意义下很好的近似, 或者说 P 包含了 A 尽可能多的信息, 这可以保证 $P^{-1}A$ 接近于单位阵 I , 而且 $Pz = r$ 的计算比 $Az = r$ 容易很多。

Remark. 预条件子的构造通常都是纯代数的角度出发, 利用矩阵 A 的系数信息和特定结构进行构造, 也有一些预条件子的构造基于方程组问题背后的实际物理背景。