

Iterative Methods for Sparse Linear Systems

阅读笔记

康豪

2023 年 1 月 27 日

前言

拿着这本迭代方法的圣经已经读了有段时间了。奈何该书内容过于全面，笔者读至书中就已忘了前文。故特意计划每日编写几页读书笔记，一来激励自己每日坚持学习，二来方便日后时时复习纠错。

数学不论天才，平庸是懒惰的借口。希望自己日后愈加勤勉，坚持走完从理学学士到理学博士的漫长旅程，拿到学术圈的入门券。

康豪

2023 年 1 月 10 日

目录

第一章 数学基础	1
1.1 矩阵	1
1.2 方阵和特征值	2
1.3 矩阵的类型	3
1.4 向量内积和范数	5
1.5 矩阵范数	6
1.6 子空间、值域和核	7
1.7 正交向量和子空间	8
1.7.1 正交向量和正交子空间	8
1.7.2 正交基的构造	9
1.8 矩阵的标准形	11
1.8.1 矩阵的 Jordan 标准形	11
1.8.2 矩阵的 Schur 标准形	11
1.9 正规矩阵和 Hermitian 矩阵	11
1.10 非负矩阵和 M-矩阵	11
1.11 正定矩阵	11
1.12 投影算子	11
1.13 摄动分析	12
1.14 偏微分方程概论	12

目录	II
1.15 有限差分方法	12
1.16 有限元方法	12
1.17 有限体积方法	12
1.18 稀疏矩阵的图表示	12
1.19 稀疏矩阵的存储方法	12
第二章 古典迭代法	13
2.1 Jacobi, Gauss-Seidel 和 SOR 方法	13
2.2 收敛性分析	16
第三章 Krylov 子空间方法	18
3.1 正交投影方法	18
3.1.1 共轭梯度法	19
3.2 正交化方法	20
3.2.1 GMRES 方法	20
3.2.2 GMRES 方法的变种	21
第四章 预处理技术	22
4.1 Krylov 子空间的预条件子	22
4.1.1 预处理与预条件子	22
4.1.2 预处理 CG 方法	23
4.1.3 预处理 GMRES 方法	24
4.2 一般矩阵预条件子	25
4.2.1 Jacobi, SOR 和 SSOR 预条件子	26
第五章 并行方法和并行预条件子	28
第六章 多重网格方法	29
第七章 区域分解方法	30

第一章 数学基础

第一章的主要内容是在迭代方法和预处理技术的理论中涉及的数学基础知识，主要包括矩阵分析、偏微分方程的数值解法、稀疏矩阵与图论等三个板块.

1.1 矩阵

为了一般性，本章中所有向量空间都考虑为复空间.

一个 $n \times m$ 的复矩阵 (matrix) A 就是指一个 $n \times m$ 大小的数组，这个数组的每个元素都是复数：

$$a_{ij}, i = 1, \dots, n, j = 1, \dots, m,$$

全体 $n \times m$ 的矩阵的集合构成复数域上的线性空间，并记为 $\mathbb{C}^{n \times m}$. 矩阵可以定义加法、数量乘法和矩阵乘法，在此不做回顾. 矩阵 A 的转置 (transpose) 是指将矩阵的行和列的各元素互换位置，并记为 A^T . 但在复空间中，我们应该使用矩阵的共轭转置 (transpose conjugate) 才更为合理. 矩阵 A 的共轭转置 A^H 定义为：

$$A^H = \overline{A}^T = \overline{A^T}$$

在有限维线性空间中，矩阵和线性映射的关系非常紧密. 事实上，给定像空间和原像空间的一组基后，这两个空间之间一个线性映射就可以被唯

一地表示为一个矩阵。在泛函分析的讨论中，即对无限维线性空间的讨论里，这种关系被合理地推广到了更一般的度量空间和算子之中。

1.2 方阵和特征值

如果矩阵的行数恰好等于其列数，则称这个矩阵是一个方阵 (square matrices). 单位矩阵 (Identity matrix) 是一个重要的方阵。对任意的矩阵 A ，它满足 $IA = AI = A$. 由此我们还可以定义矩阵的逆矩阵 (inverse matrices). 如果一个矩阵 A 的逆存在，则其逆矩阵 C 满足

$$CA = AC = I$$

并记逆矩阵 C 为 A^{-1} .

矩阵的行列式 (determinant) 是矩阵不同行不同列的元素的乘积. 如果矩阵的行列式为 0，则称矩阵是奇异 (singular) 的，否则称矩阵是非奇异的 (nonsingular).

矩阵的行列式具有以下性质：

- $\det(AB) = \det(A) \det(B)$
- $\det(A^t) = \det(A)$
- $\det(\alpha A) = \alpha^n \det(A)$
- $\det(\overline{A}) = \overline{\det(A)}$

定义 1.2.1. 如果 \mathbb{C}^n 中的一个非零向量 u 满足 $Au = \lambda u$ ，则称复标量 λ 是方阵 A 的一个特征值 (eigenvalue)，向量 u 是 A 与 λ 相关的特征向量 (eigenvector). 矩阵 A 的全体特征向量的集合称为矩阵 A 的谱 (spectrum)，并记作 $\sigma(A)$.

值得注意的是，如果变量 λ 是矩阵 A 的一个特征值，当且仅当 λ 是 A 的特征多项式的一个根。因此，矩阵 A 至多有 n 个不同的特征值。

命题 1.2.2. 矩阵 A 是非奇异的，当且仅当它的逆存在。

上面的命题告诉我们矩阵的逆是否存在可以通过矩阵的行列式是否为 0 来判断。

矩阵特征值的绝对值的最大值叫做矩阵的谱半径 (spectral radius)，并记作 $\rho(A)$ 。

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

矩阵的迹 (trace) 是矩阵对角线上所有元素的和。

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| \operatorname{tr}(A) = \sum_{i=1}^n a_{ii}$$

同时，矩阵的迹也等于矩阵的特征多项式的所有根的和 (重根按重数计算)。

关于矩阵的共轭转置和矩阵特征值的关系我们有如下命题。

命题 1.2.3. 如果 λ 是矩阵 A 的特征值，那么 $\bar{\lambda}$ 是 A^H 的特征值。矩阵 A^H 与 $\bar{\lambda}$ 有关的特征向量 u 被称为矩阵的左特征向量。

当需要进行区分时，矩阵 A 的特征向量则被称为右特征向量，矩阵 A 的左右特征向量满足以下关系：

$$Au = \lambda u, v^H A = \lambda v^H$$

也可以等价地表示为：

$$u^H A^H = \bar{\lambda} u^H, A^H v = \bar{\lambda} v$$

1.3 矩阵的类型

求解线性系统的不同方法的选择要依赖于矩阵 A 的不同结构，下面总结了一些重要类型的矩阵。

- 对称矩阵 (symmetric matrix): $A^T = A$

- 埃尔米特矩阵 (hermitian matrix): $A^H = A$
- 斜对称矩阵 (skew-symmetric matrix): $A^T = -A$
- 斜埃尔米特矩阵 (skew-hermitian matrix): $A^H = -A$
- 正规矩阵 (normal matrix): $A^H A = A A^H$
- 酉矩阵 (unitary matrix): $Q^H Q = I$
- 非负矩阵 (nonnegative matrix): $a_{ij} \geq 0, i = 1, \dots, n, j = 1, \dots, m$

值得注意的是, 上述酉矩阵 (也叫么正) Q 的逆就是 Q 的共轭转置. 因为

$$Q^H Q = I \rightarrow Q^{-1} = Q^H \quad (1.1)$$

满足 $Q^H Q$ 为对角矩阵的矩阵 Q 也被称为是正交的。

一些矩阵因其具有的特殊结构而方便我们进行计算. 下面给出了一些在数值分析和科学计算领域中具有非常重要地位的矩阵.

- 对角矩阵 (diagonal matrix): $a_{ij} = 0, \text{for } i \neq j$
- 上三角矩阵 (upper triangular matrix): $a_{ij} = 0, \text{for } i > j$
- 下三角矩阵 (lower triangular matrix): $a_{ij} = 0, \text{for } i < j$
- 三对角矩阵 (tridiagonal matrix)
- 带状矩阵 (banded matrix)
- 上海森伯格矩阵 (upper hessenberg matrix)
- 块对角矩阵 (block diagonal matrix)
- 置换矩阵 (permutation matrix): A 的列为单位矩阵之列的一个置换.

1.4 向量内积和范数

定义 1.4.1. 定义在向量空间 \mathbb{X} 上的一个向量内积 (*inner product*) 是指从 $\mathbb{X} \times \mathbb{X}$ 映到 \mathbb{C} 上的一个映射 s ,

$$x \in \mathbb{X}, y \in \mathbb{X} \rightarrow s(x, y) \in \mathbb{C}$$

这个映射要满足:

- s 是正定的, 即

$$s(x, x) > 0, \forall x \neq 0,$$

- s 是自共轭的, 即

$$s(y, x) = \overline{s(x, y)}, \forall x, y \in \mathbb{X},$$

- s 保持线性, 即

$$s(\lambda_1 x_1 + \lambda_2 x_2, y) = \lambda_1 s(x_1, y) + \lambda_2 s(x_2, y), \quad \forall x_1, x_2 \in \mathbb{X}, \forall \lambda_1, \lambda_2 \in \mathbb{C}.$$

特别地, 取向量空间为 \mathbb{C} 时, 欧几里得内积

$$(x, y) = \sum_{i=1}^n x_i \bar{y}_i \quad (1.2)$$

为“标准内积”. 有时上式也写作:

$$(x, y) = y^H x \quad (1.3)$$

命题 1.4.2. 对任意的酉矩阵 Q 和向量 x, y , 酉矩阵保持欧几里得内积不变, 即

$$(Qx, Qy) = (x, y)$$

证明.

$$(Qx, Qy) = (x, Q^H Qy) = (x, y)$$

□

定义 1.4.3. 定义在向量空间 \mathbb{X} 上的向量范数 $\|x\|$ 是一个实值函数, 并且需要满足以下三个条件:

- $\|x\| \geq 0, \forall x \in \mathbb{X}$, 且 $\|x\| = 0$ iff $x = 0$,
- $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{X}, \forall \alpha \in \mathbb{C}$,
- $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{X}$.

特别地, 当取 $\mathbb{X} = \mathbb{C}^n$ 时, 我们可以通过内积 (1.2) 定义欧几里得范数:

$$\|x\|_2 = (x, x)^{1/2}$$

由命题 1.4.2, 我们可以推出酉矩阵同时可以保持欧几里得范数.

$$\|Qx\|_2 = \|x\|_2, \forall x$$

也就是说, 酉矩阵相关的线性变换是一种等距变换.

在数值线性代数中, 最常用的一种向量范数是 *Hölder* 范数,

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (1.4)$$

注意, 当 p 趋于无穷大时, $\|x\|_p$ 的极限存在, 并且等于 x_i 的最大模. 这定义了一个范数 $\|\cdot\|_\infty$ $p = 1, p = 2$ 和 $p = \infty$ 的情况导出了实践中最重要的几种范数:

$$\begin{aligned} \|x\|_1 &= |x_1| + |x_2| + \cdots + |x_n| \\ \|x\|_2 &= [|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2]^{1/2} \\ \|x\|_\infty &= \max_{i=1, \dots, n} |x_i| \end{aligned}$$

1.5 矩阵范数

对一般的矩阵 $A \in \mathbb{C}^{n \times m}$, 其矩阵范数有:

$$\|A\|_1 = \max_{j=1, \dots, m} \sum_{i=1}^n |a_{ij}| \quad (1.5)$$

$$\|A\|_2 = [\rho(A^H A)]^{1/2} = [\rho(AA^H)]^{1/2}, \quad (1.6)$$

$$\|A\|_\infty = \max_{i=1, \dots, n} \sum_{j=1}^m |a_{ij}|. \quad (1.7)$$

以及 Frobenius 范数, 简称为 F 范数:

$$\|A\|_F = [\text{tr}(A^H A)]^{1/2} = [\text{tr}(AA^H)]^{1/2} \quad (1.8)$$

由于 AA^H 的取值总是非负的, 它们的平方根称为 A 的奇异值, 记为 $\sigma_i, i = 1, \dots, m$ 因此, 关系式 (1.6) 表明 $\|A\|_2$ 等于 A 的最大奇异值 σ_1 .

对于某种矩阵范数 $\|\cdot\|$, 定义 $k(A) = \|A\|\|A^{-1}\|$ 为矩阵 A 关于此范数的条件数. 当使用标准范数 $\|\cdot\|_p, p = 1, 2, \infty$ 时, 习惯上表示为 $k_p(A) = \|A\|_p\|A^{-1}\|_p$.

1.6 子空间、值域和核

\mathbb{C}_n 的子空间是 \mathbb{C}_n 的一个子集, 所以也是复向量空间. \mathbb{C}_n 的一组向量集合 $G = \{a_1, a_2, \dots, a_q\}$ 的所有线性组合是一个向量空间, 称为 G 的线性张成子空间.

$$\begin{aligned} \text{span}\{G\} &= \text{span}\{a_1, a_2, \dots, a_q\} \\ &= \left\{ z \in \mathbb{C}^n \mid z = \sum_{i=1}^q \alpha_i a_i; \{\alpha_i\}_{i=1, \dots, q} \in \mathbb{C}^q \right\} \end{aligned}$$

如果 a_i 是线性无关的, 则 $\text{span}\{G\}$ 的每一个向量都可以唯一地被表示为 a_i 的线性组合. 集合 G 称为子空间 $\text{span}\{G\}$ 的一组基.

给定两个向量子空间 S_1 和 S_2 , 它们的和 S 是一个子空间, 其每一个向量都是 S_1 的一个向量与 S_2 的一个向量的和. 两个子空间的交也是一个子空间. 如果 S_1 和 S_2 的交退化为 $\{0\}$, 则称 S_1 与 S_2 之和为直和, 表示为 $S = S_1 \oplus S_2$.

与矩阵 $A \in \mathbb{C}^{n \times m}$ 有关的两个重要子空间是其值域

$$\text{Ran}(A) = \{Ax \mid x \in \mathbb{C}^m\}, \quad (1.9)$$

及其核或零空间:

$$\text{Null}(A) = \{x \in \mathbb{C}^m \mid Ax = 0\} \quad (1.10)$$

线性代数中的一个基本结论可以被表达如下;

$$\mathbb{C}^n = \text{Ran}(A) \oplus \text{Null}(A^T) \quad (1.11)$$

同样的结果适用于 A 的转置, 得到:

$$\mathbb{C}^m = \text{Ran}(A^T) \oplus \text{Null}(A) \quad (1.12)$$

当 $AS \subset S$ 时, 称子空间 S 在方阵 A 之下是不变的. 特别地, 对 A 的任意特征值 λ , 子空间 $\text{Ker}(A - \lambda I)$ 在 A 下不变. 子空间 $\text{Ker}(A - \lambda I)$ 称为相应于 λ 的特征子空间, 它包含零向量和 A 相应于 λ 的所有特征向量.

1.7 正交向量和子空间

1.7.1 正交向量和正交子空间

设向量集合 $G = \{a_1, a_2, \dots, a_r\}$, 如果 G 满足

$$(a_i, a_j) = 0, \text{ when } i \neq j,$$

则称向量 G 是正交向量。此外, 如果 G 的每个向量都有一个 2 范数等于 1, 则称 G 是标准正交的. 与子空间 S 的所有向量正交的向量称为与该子空间 S 正交. 所有与 S 正交的向量的集合是一个称为 S 的正交补的向量子空间, 记为 S^\perp . 空间 \mathbb{C}^n 是 S 与其正交补的直和. 因此, 任何向量 x 都可以以唯一的方式写成 S 中的向量与 S^\perp 中的向量之和. 将 x 映射到其在子空间 S 中的分量的算子是到 S 上的正交投影. 每个子空间都有一个标准正交基, 它是通过取任意一个基并将其“标准正交化”而得到的, 标准正交化可以用一些不同的算法实现.

1.7.2 正交基的构造

Gram-Schmidt 正交化方法

给定一组线性无关向量 x_1, x_2, \dots, x_r , 首先归一化向量 x_i , 也就是将其除以其 2 范数, 以获得范数为 1 的缩放向量 q_1 。然后, 通过从 x_2 减去 q_1 的倍数使 x_2 相对于向量 q_1 正交, 以使所得向量正交于 q_1 , 即,

$$x_2 \leftarrow x_2 - (x_2, q_1) q_1$$

然后再次归一化所得到向量以产生 q_2 , 如此持续进行下去。Gram-Schmidt 过程的第 i 步就是使向量 x_i 相对于所有先前向量 q_j 正交化。

Algorithm 1 Pseudocode of Gram-Schmidt

- 1: 计算 $r_{11} := \|x_1\|_2$, 如果 $r_{11} = 0$ 停止, 否则计算 $q_1 := x_1 / r_{11}$;
 - 2: **for** $j = 2, \dots, r$; **do**
 - 3: 计算 $r_{ij} := (x_j, q_i)$, for $i = 1, 2, \dots, j-1$;
 - 4: $\hat{q} := x_j - \sum_{i=1}^{j-1} r_{ij} q_i$;
 - 5: $r_{jj} := \|\hat{q}\|_2$;
 - 6: 如果 $r_{jj} = 0$, 停止, 否则 $q_j := \hat{q} / r_{jj}$
 - 7: **end for**
-

矩阵的 QR 分解和修正的 Gram-Schmit 正交化方法

很容易证明上述算法不会终止, 即只要向量组 x_1, x_2, \dots, x_r 是线性无关的, 算法就能完成所有步骤. 从第 4 行和第 5 行可以清楚地看出, 在算法的每一步, 以下关系成立:

$$x_j = \sum_{i=1}^j r_{ij} q_i$$

如果 $X = [x_1, x_2, \dots, x_r]$, $Q = [q_1, q_2, \dots, q_r]$, 若 R 表示一个 $r \times r$ 的上

三角矩阵, 其非零元素为算法中定义的 r_{ij} , 则上述关系式可写成:

$$X = QR \quad (1.13)$$

这称为 $n \times r$ 矩阵 X 的 QR 分解. 当 X 的列向量形成线性无关的向量集时, 矩阵的 QR 分解存在. 上述算法是标准的 Gram-Schmidt 过程. 存在具有更好数值特性的算法的替代公式. 其中最著名的是改进的 Gram-Schmidt(MGS) 算法.

Algorithm 2 Pseudocode of Modified Gram-Schmidt

```

1: 定义  $r_{11} := \|x_1\|_2$ , 如果  $r_{11} = 0$  停止, 否则计算  $q_1 := x_1 / r_{11}$ ;
2: for  $j = 2, \dots, r$ , do
3:    $\hat{q} := x_j$ 
4:   for  $i = 1, \dots, j - 1$ , do
5:      $r_{ij} := (\hat{q}, q_i)$ ;
6:      $\hat{q} := \hat{q} - r_{ij}q_i$ ;
7:   end for;
8:   计算  $r_{jj} = \|\hat{q}\|_2$ ;
9:   如果  $r_{jj} = 0$ , 停止, 否则  $q_j := \hat{q} / r_{jj}$ 
10: end for

```

Householder 正交化方法

使向量序列正交化的另一个可选方案是 Householder 算法. 这个方法将用到使用 Householder 反射, 即下列形式的矩阵

$$P = I - 2ww^T \quad (1.14)$$

其中, w 是 2 范数下的单位向量. 在几何上, 向量 Px 表示 x 关于超平面 $\text{span}\{w\}^\perp$ 的像.

1.8 矩阵的标准形

1.8.1 矩阵的 Jordan 标准形

1.8.2 矩阵的 Schur 标准形

1.9 正规矩阵和 Hermitian 矩阵

1.10 非负矩阵和 M-矩阵

1.11 正定矩阵

1.12 投影算子

投影算子 P 是从 \mathbb{C}^n 到自身的一个幂等的线性变换，即使得 $P^2 = P$.

1.13 摄动分析

1.14 偏微分方程概论

1.15 有限差分方法

1.16 有限元方法

1.17 有限体积方法

1.18 稀疏矩阵的图表示

1.19 稀疏矩阵的存储方法

第二章 古典迭代法

考虑线性系统

$$Ax = b \quad (2.1)$$

其中, x 是未知数, b 是常数项. 构造迭代法的基本思想是将上式变换为有如下形式的具有相同解的等价方程组:

$$x = Hx + d \quad (2.2)$$

然后任取一个初始值 x^0 , 由迭代公式

$$x^{k+1} = Hx^k + d, k = 1, 2, \dots \quad (2.3)$$

产生迭代序列, 其中 H 被称为迭代矩阵. 若迭代矩阵 H 与迭代指标 k 有关, 则迭代法为定常迭代法, 否则为非定常迭代法. 本章提及的古典迭代法, 也称经典迭代法或传统迭代法, 本质上都属于定常迭代法.

2.1 Jacobi, Gauss-Seidel 和 SOR 方法

给定一个 $n \times n$ 实矩阵 A 和一个实 n 维向量 b , 需要解决的问题是: 求 $x \in \mathbb{R}^n$, 使得

$$Ax = b \quad (2.4)$$

方程 (2.4) 是一个线性系统, A 是系数矩阵, b 是右端项, x 是未知数的向量. 本章介绍的大多数方法都涉及到通过一次修改近似向量解的一个或几个

分量来从一个迭代传递到下一个迭代. 一种方法是零化残差向量 $b - Ax$ 的某些分量. 这些方法的收敛性很少能保证对所有矩阵都是收敛的, 但是对于系数矩阵由椭圆型偏微分方程的有限差分离散化产生的情况, 已经存在大量的理论帮助. 我们从如下这个分解开始:

$$A = D - E - F \quad (2.5)$$

其中 D 是 A 的对角, $-E$ 是 A 的下半部分, $-F$ 是 A 的上半部分, 我们假定 A 的对角元素都是非零的. Jacobi 迭代通过确定下一个近似的第 i 个分量, 以便消去残差向量的第 i 个分量. 在下文中, ξ_i^k 表示迭代 x_k 的第 i 个分量, β_i 表示右端项 b 的第 i 个分量. 因此, 残差:

$$(b - Ax_{k+1})_i = 0, \quad (2.6)$$

可以表示为如下形式:

$$a_{ii}\xi_i^{(k+1)} = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}\xi_j^{(k)} + \beta_i,$$

或者,

$$\xi_i^{(k+1)} = \frac{1}{a_{ii}} \left(\beta_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}\xi_j^{(k)} \right) \quad i = 1, \dots, n. \quad (2.7)$$

这就是 Jacobi 迭代法的一种分量形式. 下一次迭代的所有分量可以被分组到向量 x_{k+1} 中. 上述符号可用于将 Jacobi 迭代 (2.7) 改写为向量形式:

$$x_{k+1} = D^{-1}(E + F)x_k + D^{-1}b. \quad (2.8)$$

类似地, Gauss-Seidel 迭代以 $i = 1, 2, \dots, n$ 依次消去残差的第 i 个分量. 但高赛方法的近似解在确定新分量之后会被立即更新. 由于顺序是 $i = 1, 2, \dots, n$, 所以第 i 步的结果是

$$\beta_i - \sum_{j=1}^{i-1} a_{ij}\xi_j^{(k+1)} - a_{ij}\xi_i^{(k+1)} - \sum_{j=i+1}^n a_{ij}\xi_j^{(k)} = 0, \quad (2.9)$$

这将导出迭代式:

$$\xi_i^{(k+1)} = \frac{1}{a_{ii}} \left(- \sum_{j=1}^{i-1} a_{ij} \xi_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} \xi_j^{(k)} + \beta_i \right), i = 1, \dots, n. \quad (2.10)$$

定义式 (2.9) 可以写成矩阵形式:

$$b + Ex_{k+1} - Dx_{k+1} + Fx_k = 0, \quad (2.11)$$

这直接导出高斯-赛德尔迭代的矩阵形式:

$$x_{k+1} = (D - E)^{-1} Fx_k + (D - E)^{-1} b. \quad (2.12)$$

计算公式 (2.8) 中的新近似需要乘以对角矩阵 D 的逆矩阵. 在公式 2.12 中, 三角方程组必须用 $D-E$ 求解, $D-E$ 是 A 的下三角部分. 因此, 高斯-赛德尔步骤中的新近似可以通过用矩阵 $D-E$ 求解三角方程组来确定, 也可以由公式 2.10 确定.

Jacobi 和 Gauss-Seidel 迭代都具有以下形式:

$$Mx_{k+1} = Nx_k + b = (M - A)x_k + b, \quad (2.13)$$

其中,

$$A = M - N \quad (2.14)$$

是 A 的分裂, 而 $M = D$ (在 Jacobi 方法中), $M = D-E$ (正向高斯-赛德尔方法中), $M = D-F$ (反向高斯-赛德尔中). 对于式 (2.14) 的任何分裂形式, 只要 M 是非奇异的, 都可以定义式 (2.13) 的迭代方法. 而超松弛方法是基于分裂

$$\omega A = (D - \omega E) - (\omega F + (1 - \omega)D),$$

并通过递推给出了相应的逐次超松弛 (SOR) 方法

$$(D - \omega E)x_{k+1} = [\omega F + (1 - \omega)D]x_k + \omega b. \quad (2.15)$$

上述迭代对应于松弛序列

$$\xi_i^{(k+1)} = \omega \xi_i^{GS} + (1 - \omega) \xi_i^{(k)}, i = 1, 2, \dots, n,$$

式中 ξ_i^{GS} 由式 (2.10) 右边的表达式定义. 反向 SOR 可以类似于反向高斯-赛德尔定义.

2.2 收敛性分析

在上一节中提到的所有方法都定义了一个如下的迭代公式:

$$x_{k+1} = Gx_k + f, \quad (2.16)$$

其中 G 是一个确定迭代矩阵. 在研究收敛性时, 我们需要讨论的问题包括:

- a) 若迭代收敛, 则极限真的是原系统的解吗?
- b) 在什么条件下迭代收敛?
- c) 当迭代收敛时, 它的速度有多快?

若上述迭代收敛, 则其极限 x 满足

$$x = Gx + f. \quad (2.17)$$

在上述迭代由分裂 $A = M - N$ 产生的情况下, 很容易看出上述方程组的解 x 与原方程组 $Ax = b$ 的解相同. 实际上, 在这种情况下, 序列 (2.16) 的形式为

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b$$

且其极限满足

$$Mx = Nx + b,$$

也就等价于 $Ax = b$. 这就解答了问题 (a).

如果 $I-G$ 是非奇异的, 则方程 (2.17) 存在解 x^* . 从公式 (2.16) 中减去公式 (2.17) 得到.

$$x_{k+1} - x_* = G(x_k - x_*) = \cdots = G^{k+1}(x_0 - x_*). \quad (2.18)$$

在第 1 章中得到意味着, 如果迭代矩阵 G 的谱半径小于 1, 则 $x_k - x^*$ 收敛于零. 相反, 关系式

$$x_{k+1} - x_k = G(x_k - x_{k-1}) = \cdots = G^k(f - (I - G)x_0).$$

表明, 如果迭代对任意 x_0 和 f 收敛, 则 $G^k v$ 对任意向量 v 收敛到零. 因此, $\rho(G)$ 必须小于 1, 并有以下定理:

定理 2.2.1. 设 G 是一个方阵, 使得 $\rho(G) < 1$. 那么 $I-G$ 是非奇异的, 迭代公式对任意 f 和 x_0 收敛. 相反, 如果对于任意 f 和 x_0 , 迭代公式收敛, 则 $\rho(G) < 1$.

由于计算矩阵的谱半径是非常昂贵的, 因此保证收敛的充分条件在实践中非常有意义. 利用不等式 $\rho(G) < \|G\|$, 对于任意矩阵范数, 可以得到这样一个充分条件.

推论 2.2.2. 设 G 是一个方阵, 使得 $\|G\| < 1$, 矩阵范数为 $\|\cdot\|$, 则 $I-G$ 是非奇异的, 迭代公式对任意初始向量 x_0 收敛.

第三章 Krylov 子空间方法

Krylov 子空间方法是一类非定常迭代法。

求解线性方程组

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, x, b \in \mathbb{R}^n \quad (3.1)$$

的一般投影方法是从 m 维仿射子空间 $x_0 + \mathcal{K}_m$ (称为搜索空间) 中寻求一个近似解 x_m , 它使用了 *Petrov-Galerkin* 条件:

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

其中, \mathcal{L}_m 是另一个 m 维子空间 (称为约束空间)。

Krylov 子空间方法就是将上述内容中的空间 \mathcal{K}_m 取成 Krylov 子空间:

$$\mathcal{K}_m(A, v) = \text{span} \{v, Av, A^2v, \dots, A^{m-1}v\} \quad (3.2)$$

Krylov 子空间方法的基础是投影方法, 而正交基的构造是投影方法的基础. 根据正交基构造方法的不同可以得到不同的算法. 所以正交基的构造自然成为了 Krylov 子空间方法的重要组成部分, 也是影响方法数值稳定性和成败的关键.

3.1 正交投影方法

取 $\mathcal{L}_m = \mathcal{K}_m(A, r_0)$, 这类 Krylov 子空间方法叫做正交投影 (Ritz-Galerkin) 方法. 其中最重要的一种方法就是共轭梯度法 (CG), 共轭梯度法要求矩阵

A 是对称正定的。CG 是一种理想的 Krylov 子空间方法，其能使误差的能量范数 (A 范数) 在子空间 $x_0 + \mathcal{K}_m$ 中极小，且具有短递归公式，理论上 n 步就能达到真实解。

除此之外的全正交化 (FOM) 方法和正交残差 (ORTHORES) 方法 (不要求 A 是对称正定矩阵)，也是正交化方法，且能极小化残差，但不具有短递归公式。

3.1.1 共轭梯度法

共轭梯度 (CG) 方法是 1952 年由 Hestenes 和 Stiefel 首先提出的，在没有舍入误差的情况下，至多用 n 步即可求出线性方程组的精确解。1976 年 Concus 等提出了预处理 CG 方法，至此，CG 方法得到越来越多的关注，产生了一个大的 Krylov 子空间方法解法器簇。可以说，CG 方法的出现，标志着求解线性方程组的 Krylov 子空间方法的诞生。共轭梯度方法是求解稀疏对称正定线性方程组最著名的迭代方法之一，它是到 Krylov 子空间 $\mathcal{K}_m(A, r_0)$ 上正交投影的一种实现，其中 r_0 是初始残差，它与后面的 FOM 方法是数学等价的，然而，由于 A 是对称的，用 Lanczos 三项递归的一些简化将得到更精致的算法。CG 方法的基本算法如下：

Algorithm 3 Pseudocode of Conjugate Gradient Algorithm

- 1: 计算 $r_0 := b - Ax_0$, $p_0 := r_0$;
 - 2: **for** $j = 0, 1, \dots$; 直到收敛 **do**
 - 3: $\alpha_j := (r_j, r_j) / (Ap_j, p_j)$;
 - 4: $x_{j+1} := x_j + \alpha_j p_j$;
 - 5: $r_{j+1} := r_j - \alpha_j Ap_j$;
 - 6: $\beta_j := (r_{j+1}, r_{j+1}) / (r_j, r_j)$;
 - 7: $p_{j+1} := r_{j+1} + \beta_j p_j$
 - 8: **end for**
-

3.2 正交化方法

3.2.1 GMRES 方法

广义最小残差法 (GMRES) 是基于取 $\mathcal{K} = \mathcal{K}_m$ 和 $\mathcal{L} = A\mathcal{K}_m$ 的投影方法, 其中 \mathcal{K}_m 是第 m 个 Krylov 子空间, $v_1 = r_0/\|r_0\|_2$. 这种方法使 $x_0 + \mathcal{K}_m$ 中所有向量的剩余范数最小. 基于该方法的算法的实现类似于 FOM 算法的实现.

$x_0 + \mathcal{K}_m$ 中的任何向量 x 都可以表示为:

$$x = x_0 + V_m y, \quad (3.3)$$

其中 y 是 m 维度向量. 定义

$$J(y) = \|b - Ax\|_2 = \|b - A(x_0 + V_m y)\|_2 \quad (3.4)$$

由关系

$$AV_m = V_{m+1} \bar{H}_m$$

可得:

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) \\ &= r_0 - AV_m y \\ &= \beta v_1 - V_{m+1} \bar{H}_m y \\ &= V_{m+1} (\beta e_1 - \bar{H}_m y). \end{aligned} \quad (3.5)$$

由于 V_{m+1} 的列向量是标准正交的, 则

$$J(y) \equiv \|b - A(x_0 + V_m y)\|_2 = \|\beta e_1 - \bar{H}_m y\|_2. \quad (3.6)$$

GMRES 近似是 $x_0 + \mathcal{K}_m$ 的唯一向量. 由式 (3.3) 和式 (3.6) 可以非常简单地得到 $x_m = x_0 + V_m y_m$, 其中 y_m 使函数 $J(y) = \|\beta e_1 - \bar{H}_m y\|_2$ 最小, 即

$$x_m = x_0 + V_m y_m, \quad \text{where} \quad (3.7)$$

$$y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2. \quad (3.8)$$

得到 y_m 廉价的计算, 因为它需要的解决方案 $(m+1) \times m$ 通常是小的最小二乘问题. 这给出下面的算法.

Algorithm 4 Pseudocode of GMRES

- 1: 计算 $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, and $v_1 := r_0/\beta$;
 - 2: **for** $j = 1, 2, \dots, m$; **do**
 - 3: 计算 $w_j := Av_j$;
 - 4: **for** $i = 1, \dots, j$ **do**
 - 5: $h_{ij} := (w_j, v_i)$;
 - 6: $w_j := w_j - h_{ij}v_i$;
 - 7: **end for**
 - 8: $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ set $m := j$ and go to 11;
 - 9: $v_{j+1} = w_j/h_{j+1,j}$
 - 10: **end for**
 - 11: Define the $(m+1) \times m$ Hessenberg matrix $\bar{H}_m = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}$.
 - 12: Compute y_m the minimizer of $\|\beta e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$.
-

3.2.2 GMRES 方法的变种

第四章 预处理技术

4.1 Krylov 子空间的预条件子

虽然前面章节中介绍的方法在理论上是有充分根据的，但对于流体动力学或电子设备模拟等典型应用中出现的问题，它们都可能出现收敛缓慢的情况。预处理是 Krylov 子空间方法在这些应用中取得成功的关键因素。本章讨论的是已经出现的迭代法的预处理版本。标准预处理技术将在下一章中介绍。

4.1.1 预处理与预条件子

相对于直接解法，缺乏鲁棒性是迭代法的一个公认的弱点。这个缺点妨碍了迭代法在工业应用中的接受，尽管它们对于非常大的线性系统具有内在的吸引力。预处理可以提高迭代算法的效率和鲁棒性。预处理是将原始线性系统转换为具有相同解的线性系统的一种方法，但可能更容易用迭代方法求解。一般来说，迭代技术的可靠性，在处理各种应用时，更多地取决于预条件子的质量，而不是所使用的特定 Krylov 子空间加速器。我们将在下一章中详细介绍其中的一些预条件子。

首先，预处理的第一步是求预处理矩阵 M 。矩阵 M 可以用许多不同的方式定义，但它必须满足一些最低要求。从实用的观点看，对 M 的最大要求是求解线性方程组 $Mx = b$ 的费用低。这是因为预处理算法在每一步都需要具有矩阵 M 的线性系统解。 M 也应该在某种意义上接近于 A ，而且它应

该是非奇异的.

有 3 种方式可以实现预条件子. 第一种是左预处理,

$$M^{-1}Ax = M^{-1}b \quad (4.1)$$

或者, 从右边进行预处理

$$AM^{-1}u = b, \quad x \equiv M^{-1}u \quad (4.2)$$

注意, 上述公式相当于做变量代换 $u = Mx$ 变化, 并相对于未知 u 求解系统. 最后, 一种常见的情况是预条件子以因式分解的形式可用

$$M = M_L M_R$$

其中, 通常 M_L 和 M_R 是三角矩阵. 在这种情况下, 预处理可以拆分为

$$M_L^{-1}AM_R^{-1}u = M_L^{-1}b, \quad x \equiv M_R^{-1}u \quad (4.3)$$

当原始矩阵对称时, 必须保持对称性, 因此必须分解预条件子. 然而, 即使 M 不能以因式分解的形式得到, 也有其他保持对称性的方法, 或者更确切地说, 利用对称性的方法. 这将在接下来的共轭梯度法中讨论.

4.1.2 预处理 CG 方法

CG 方法的预处理需要考虑如何保持对称性. 当 M 有不完全 Cholesky 因式分解的形式时, 即, 当

$$M = LL^T,$$

则保持对称性的简单方法是使用“分裂”预条件子, 这样可以产生对称正定矩阵.

$$L^{-1}AL^{-T}u = L^{-1}b, \quad x = L^{-T}u \quad (4.4)$$

除此之外，注意到 $M^{-1}A$ 对于 M-内积是自伴的，

$$(M^{-1}Ax, y)_M = (Ax, y) = (x, Ay) = (x, M(M^{-1}A)y) = (x, M^{-1}Ay)_M$$

因此，另一种方法是用 M-内积代替共轭梯度算法中常用的欧氏内积. 如果针对这个新的内积重写 CG 算法，用 $r_j = b - Ax_j$ 表示原始残差，用 $z_j = M^{-1}r_j$ 表示预处理系统的残差. 由于 $(z_j, z_j)_M = (r_j, r_j)$ 和 $(M^{-1}Ap_j, p_j)_M = (Ap_j, p_j)$ ，M-内积不需要显式计算. 利用该结论，获得以下算法.

Algorithm 5 Pseudocode of Preconditioned Conjugate Gradient Algorithm

```

1: 计算  $r_0 := b - Ax_0$ ,  $z_0 = M^{-1}r_0$ ,  $p_0 := r_0$ ;
2: for  $j = 0, 1, \dots$ ; 直到收敛 do
3:    $\alpha_j := (r_j, r_j) / (Ap_j, p_j)$ ;
4:    $x_{j+1} := x_j + \alpha_j p_j$ ;
5:    $r_{j+1} := r_j - \alpha_j Ap_j$ ;
6:    $z_{j+1} := M^{-1}r_{j+1}$ ;
7:    $\beta_j := (r_{j+1}, z_{j+1}) / (r_j, z_j)$ ;
8:    $p_{j+1} := z_{j+1} + \beta_j p_j$ 
9: end for
```

4.1.3 预处理 GMRES 方法

对于 GMRES 或其他非对称迭代求解器，可以使用与“共轭梯度”相同的三个选项来应用预处理操作 (即左预处理、分割预处理和右预处理). 然而，有一个根本的区别——正确的预处理版本将产生所谓的灵活变体，即：其中预条件子可以在每一步改变的变体. 这种能力在某些应用中非常有用.

左预处理 GMRES

如前所述, 将左预处理 GMRES 算法定义为应用于系统的 GMRES 算法,

$$M^{-1}Ax = M^{-1}b.$$

将 GMRES 直接应用于上述线性系统, 得到以下 GMRES 的预处理版本.

Algorithm 6 Pseudocode of GMRES with Left Preconditioning

- 1: Compute $r_0 = M^{-1}(b - Ax_0)$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$;
 - 2: **for** $j = 1, 2, \dots, m$; **do**
 - 3: Compute $w := M^{-1}Av_j$;
 - 4: **for** $i = 1, \dots, j$ **do**
 - 5: $h_{ij} := (w_j, v_i)$;
 - 6: $w_j := w_j - h_{ij}v_i$;
 - 7: **end for**
 - 8: $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$;
 - 9: **end for**
 - 10: Define $V_m := [v_1, \dots, v_m]$, $\bar{H}_m = \{h_{i,j}\}_{1 \leq i \leq j+1; 1 \leq j \leq m}$
 - 11: Compute $y_m = \operatorname{argmin}_y \|\beta e_1 - H_m y\|_2$, and $x_m = x_0 + V_m y_m$
 - 12: If satisfied Stop, else set $x_0 := x_m$ and GoTo 1
-

4.2 一般矩阵预条件子

寻找一个好的预条件子来求解给定的稀疏线性系统通常被视为艺术和科学的结合. 预条件子可以定义为任何辅助的近似解算器, 它与外部迭代技术相结合, 典型的是前面章节中看到的 Krylov 子空间迭代. 这一小节介绍了一些用于预处理一般稀疏线性系统的最成功的技术.

粗略地说, 预条件子是对原线性系统的任何形式的显式或显式修改, 它使线性系统“更容易”用给定的迭代方法求解. 例如, 缩放线性系统的所有行以使对角元素等于一是预处理的显式形式, 所得到的系统可以通过 Krylov 子空间方法来求解, 并且可能需要比原始系统更少的步骤来收敛. 作为另一个例子, 求解线性系统

$$M^{-1}Ax = M^{-1}b$$

其中 M^{-1} 是一些复杂的映射, 可能涉及 FFT 变换、积分计算和辅助线性系统解, 可能是另一种形式的预处理. 在这里, 矩阵 M 和 $M^{-1}A$ 不太可能被显式计算. 相反, 只要需要, 迭代过程就以 A 和 M^{-1} 进行. 实际上, 预处理运算 M^{-1} 应该可以很方便地应用于任意向量.

定义预条件子的最简单方法之一是对原矩阵 A 进行不完全分解. 这需要分解 $A = LU - R$, 其中 L 和 U 分别与 A 的上、下部分具有相同的非零结构, R 是分解的残差或误差. 这种不完全因式分解称为 $ILU(0)$, 计算起来相当简单且成本低廉.

4.2.1 Jacobi, SOR 和 SSOR 预条件子

求解线性方程组的不动点迭代法

$$Ax = b$$

采取一般形式

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b \quad (4.5)$$

其中 M 和 N 实现了将 A 分裂为

$$A = M - N. \quad (4.6)$$

上述迭代的形式为

$$x_{k+1} = Gx_k + f \quad (4.7)$$

其中 $f = M^{-1}b$, 且

$$\begin{aligned} G &= M^{-1}N = M^{-1}(M - A) \\ &= I - M^{-1}A. \end{aligned}$$

对于 Jacobi 和 Gauss Seidel, 已经证明了

$$G_{JA}(A) = I - D^{-1}A \quad (4.8)$$

$$G_{GS}(A) = I - (D - E)^{-1}A, \quad (4.9)$$

其中 $A = D - E - F$.

第五章 并行方法和并行预条件子

第六章 多重网格方法

第七章 区域分解方法

名词索引

- Krylov 子空间, 18
- 三对角矩阵 (tridiagonal matrix), 4
- 上三角矩阵 (Upper triangular matrix), 4
- 上海森伯格矩阵 (upper hessenberg matrix), 4
- 下三角矩阵 (lower triangular matrix), 4
- 共轭梯度法 (CG), 18
- 共轭转置 (transpose conjugate), 1
- 内积 (inner product), 5
- 单位矩阵 (Identity matrix), 2
- 块对角矩阵 (block diagonal matrix), 4
- 埃尔米特矩阵 (hermitian matrix), 4
- 奇异 (singular), 2
- 对称矩阵 (symmetric matrix), 3
- 对角矩阵 (diagonal matrix), 4
- 带状矩阵 (banded matrix), 4
- 斜埃尔米特矩阵 (skew-hermitian matrix), 4
- 斜对称矩阵 (skew-symmetric matrix), 4
- 方阵 (square matrices), 2
- 正交投影方法 (Ritz-Galerkin), 18
- 正规矩阵 (normal matrix), 4
- 特征值 (eigenvalue), 2
- 特征向量 (eigenvector), 2
- 矩阵 (matrix), 1
- 置换矩阵 (permutation matrix), 4
- 行列式 (determinant), 2
- 谱 (spectrum), 2
- 谱半径 (spectral radius), 3
- 转置 (transpose), 1
- 迹 (trace), 3

逆矩阵 (inverse matrices), 2

酉矩阵 (unitary matrix), 4

非奇异 (nonsingular), 2

非负矩阵 (nonnegative matrix), 4