Krylov子空间算法与预处理技术及其应用

黄廷祝　主编

科学出版社

Krylov 子空间算法与预处理技术及其应用

黄廷祝　主编

科　学　出　版　社
北　京

内 容 简 介

本书深入讨论Krylov子空间算法的核心思想和理论，结合算法的推导过程，介绍Krylov子空间算法和预处理技术的最新进展，同时介绍Krylov子空间算法及预处理技术在电磁计算和数字图像处理中的应用.

本书适合作为理工类，特别是信息与计算科学、应用数学以及相关专业研究生的教科书或参考书，也可供高校师生和科研人员等阅读.

图书在版编目(CIP)数据

Krylov子空间算法与预处理技术及其应用/黄廷祝主编. —北京：科学出版社，2016.3

ISBN 978-7-03-047508-4

Ⅰ. ①K… Ⅱ. ①黄… Ⅲ. ①线性代数计算法 Ⅳ. ①O241.6

中国版本图书馆CIP数据核字（2016）第044465号

责任编辑：李静科 赵彦超／责任校对：张凤琴
责任印制：徐晓晨／封面设计：陈 敬

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

北京教图印刷有限公司 印刷
科学出版社发行 各地新华书店经销

*

2016年3月第 一 版 开本：720×1000 B5
2016年3月第一次印刷 印张：12 1/4
字数：232 000

定价：78.00元

（如有印装质量问题，我社负责调换）

前　言

计算是继理论和实验之后的第三种科学研究手段, 计算数学与科学工程计算、信息科学等领域中的大量问题都可归结为矩阵计算问题. 其中, 大规模代数系统的快速求解方法是数学研究的前沿科学问题, 也是很多工程领域亟待解决的科学计算难题. 本书主要是为计算数学有关专业硕士、博士研究生和相关科技人员提供一本既能反映矩阵计算方法的最新进展, 又具有实用性和应用启发性的教学参考书, 使之通过本书的学习, 能够对大规模线性代数方程组求解的理论、方法和本领域的部分最新研究成果及其应用有一个比较深入、全面的了解, 并为进一步学习与交叉领域应用研究打下一个启发式基础.

本书在介绍前沿算法的前提下, 注重介绍运用 Krylov 子空间算法与预处理技术解决电磁计算、数字图像处理领域的典型实际问题, 主要由 Krylov 子空间迭代算法、预处理技术和算法应用三部分组成, 第一部分包括第 1 章至第 4 章, 第二部分包括第 5 章至第 8 章, 第三部分包括第 9 章和第 10 章.

第 1 章从投影技术角度阐释 Ujević 迭代算法, 结合线性子空间理论, 介绍一种求解对称正定线性方程组的“二维双连续投影法”, 搭建定常迭代法和非定常迭代法之间的桥梁.

第 2 章介绍一类基于求解复非 Hermitian 线性方程组的基于短项递推 $\boldsymbol{A}$-双正交理论的 Krylov 子空间方法; 并在第 3 章研究运用此类 Lanczos 双共轭 $\boldsymbol{A}$-标准正交法求解计算电磁学中产生的大型线性方程组.

第 4 章研究求解多右端向量线性系统的块 Krylov 子空间迭代算法.

第 5 章和第 6 章分别研究阶梯矩阵和多项式预处理技术和 Chebyshev 型预处理技术的构造方法.

第 7 章介绍不完全 LU 分解预处理技术研究与应用; 第 8 章介绍复线性系统的求解与预处理技术.

第 9 章介绍开域电磁计算中产生的大型线性方程组求解问题, 着重展示线性代数方程组求解研究在电磁计算中的应用潜力.

第 10 章介绍迭代法和预处理方法在图像复原问题中的应用, 针对图像复原问题中线性代数系统的特殊结构, 讨论适用于大规模线性代数系统的高性能迭代法及预处理方法.

本书由黄廷祝主编. 各章编写者分别为: 荆燕飞 (第 1 章至第 3 章)、孟静 (第 4 章)、李厚彪 (第 5 章和第 6 章)、张勇 (第 7 章)、顾先明 (第 8 章)、李良 (第 9

章)、赵熙乐 (第 10 章).

本书的出版得到电子科技大学研究生院、数学科学学院的大力支持和帮助, 在此表示衷心感谢.

由于作者水平有限, 不足之处在所难免, 敬请同行和广大读者不吝批评指正.

作　者

2015 年 10 月于成都

目　录

第 1 章　二维双连续投影法

众所周知, 许多应用领域均会涉及正定线性系统, 例如, 调和分析、复分析、机械系统的振动理论, 以及包括奇异值分解和线性最小二乘问题数值解在内的矩阵理论等[1]. 目前共轭梯度 (Conjugate gradient, CG) 方法是用于求解对称正定线性方程组的一种有效且成熟的方法[1, 2].

本章基于 Ujević 提出的一种新颖的数值迭代算法[3], 结合投影技术和线性子空间理论, 介绍一种新的求解对称正定线性方程组的算法 ——"二维双连续投影法 (2D-DSPM)" [4, 5], 以便搭建定常迭代法和非定常迭代法之间的桥梁, 进一步加深对这两类方法之间关系的理解.

本章首先从 Krylov 子空间投影技术角度详细分析 Ujević 迭代算法, 并指出这种算法是一种特殊的投影方法, 然后给出 2D-DSPM 的具体推导过程以及这种方法与 Ujević 的迭代算法的理论比较结果, 并通过数值实验验证 2D-DSPM 较 Ujević 迭代算法的优越性, 最后给出本章小结.

1.1　引　　言

2006 年 Ujević 提出了一种新颖的求解正定线性方程组的数值迭代算法. 这种算法可视为 Gauss-Seidel 迭代法的改进修正方法. 事实上, 从科学计算领域广泛使用的投影技术角度考虑, 这种数值算法可以视为"一维双连续投影法 (1D-DSPM)", 而基本的 Gauss-Seidel 迭代法可以视为"一维单连续投影法 (1D-SSPM)".

考虑如下 $n \times n$ 非奇异线性方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1.1}$$

其中, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 为给定的对称正定 (SPD) 矩阵, $\boldsymbol{b} \in \mathbb{R}^n$ 为已知右端向量, $\boldsymbol{x} \in \mathbb{R}^n$ 为未知待求量. 由科学和工程中突飞猛进的科技进步所驱动, 求解此类线性系统的迭代方法发展迅速. 目前, 大多数求解实际问题中大规模线性方程组的迭代技术都会在某种程度上运用投影过程, 投影技术在许多其他科学计算领域也会以各种不同的形式加以运用, 而且可以在抽象的 Hilbert 泛函空间和有限元空间中表示出来.

投影技术是通过保证残量修正值在某种范数意义下达到最小来不断求解单个方程, 从而最终求解整个方程组的过程. 投影技术可以被广泛地应用于求解实际问题, 例如, X 线断层摄影术等产生的超定或者不定线性方程组投影过程的思想是, 在

n 维实数空间 $\mathbb{R}^n$ 的子空间中选取方程组 (1.1) 的近似解. 用 $\mathcal{K}$ 和 $\mathcal{L}$ 分别表示 m 维搜索子空间和约束子空间, $\boldsymbol{x}_0 \in \mathbb{R}^n$ 表示近似解初值. 一般地, 投影到子空间 $\mathcal{K}$ 上并且与子空间 $\mathcal{L}$ 正交的投影方法是通过强加 Petrov-Galerkin 条件来试图寻找方程组 (1.1) 的近似解 $\boldsymbol{x} \in \mathbb{R}^n$. Petrov-Galerkin 条件要求 $\boldsymbol{x}$ 为仿射空间 $\boldsymbol{x}_0 + \mathcal{K}$ 中的元素, 而对应的残量 $(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$ 与 $\mathcal{L}$ 正交. 这种条件用数学语言表述为

$$\text{求 } \boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}, \quad \text{使得} \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \perp \mathcal{L}. \tag{1.2}$$

从这种投影角度出发, Ujević 所提出的求解正定线性方程组的迭代算法可以视为一种特殊的投影方法. 这种方法在每组内循环迭代过程中 $(k = 1, \cdots, n)$, 每一步选取两个一维子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 同时作两次修正. 因此, 这种方法称为“一维双连续投影法 (1D-DSPM)”. 另外, 基本的 Gauss-Seidel 迭代法可以看成通过选取一维子空间 $\mathcal{K} = \mathcal{L} = \text{span}\{\boldsymbol{e}_i\}$ 而求解的投影方法, 其中 $\boldsymbol{e}_i$ 是单位矩阵的第 i 列. 在 Gauss-Seidel 迭代法每组内循环迭代过程中 $(k = 1, \cdots, n)$, 每一步只作一次修正, 因此, Gauss-Seidel 迭代法可以视为 1D-SSPM 通过组合 1D-DSPM 内循环的每一步选取的两个子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 结合投影技术和线性子空间理论, 可以给出一种新的求解方法. 1.3 节将给出此方法的具体推导过程. 在每一组内循环迭代过程中 $(k = 1, \cdots, n)$, 这种方法每一步仍作两次修正. 也就是说, 强加的 Petrov-Galerkin 条件要求每一步在二维子空间 $\mathcal{K}$ 中寻找近似解, 同时要求相应迭代残量与二维子空间 $\mathcal{L}$ 正交. 这里, $\mathcal{K}$ 和 $\mathcal{L}$ 选取为相同的子空间. 我们称这种方法为 2D-DSPM. 理论分析结果表明, 2D-DSPM 较 1D-DSPM 可以更好 (至少一样) 地减少迭代误差. 1.4 节数值实验表明, 2D-DSPM 在大部分实际问题求解过程中收敛速度至少是 1D-DSPM 的 1.5 倍.

本节最后介绍一些本章所用的数学符号. $\boldsymbol{e}_i$ 是单位矩阵的第 i 列. 对于任意非负整数 k, $\boldsymbol{x}_*$, $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1} \in \mathbb{R}^n$ 分别表示线性方程组 (1.1) 的精确解、当前近似解和下一步近似解.

两个向量 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ 的内积定义为 $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$. 对于任意正定矩阵 $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, $\boldsymbol{M}$-内积定义为 $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{M}} = \langle \boldsymbol{M}\boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}$. 对于任意的 $\boldsymbol{x} \in \mathbb{R}^n$, 如果 $\boldsymbol{M}$ 是对称正定矩阵, 则其相应的 $\boldsymbol{M}$-范数定义为 $||\boldsymbol{x}||_{\boldsymbol{M}}^2 = \langle \boldsymbol{M}\boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x} = (\boldsymbol{M}\boldsymbol{x})^{\mathrm{T}}\boldsymbol{x} = \langle \boldsymbol{x}, \boldsymbol{M}\boldsymbol{x} \rangle$.

为了表达和计算的方便与统一, 在每组内循环迭代过程中 $(k = 1, \cdots, n)$, 1D-DSPM 每一步所选用的子空间记为 $\mathcal{K}_1 = \text{span}\{\boldsymbol{v}_1\}$ 和 $\mathcal{K}_2 = \text{span}\{\boldsymbol{v}_2\}$, 其中 $\boldsymbol{0} \neq \boldsymbol{v}_1$, $\boldsymbol{v}_2 \in \mathbb{R}^n$ 且 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 线性无关. 相应地, 2D-DSPM 每一步所选子空间记为 $\mathcal{K} = \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$.

由于所考虑的线性方程组 (1.1) 的系数矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 对称正定, 所以内积简记为

$$a = \langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_1\rangle, \quad c = \langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_2\rangle = \langle \boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_1\rangle, \quad d = \langle \boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_2\rangle,$$

并记

$$p_1 = \langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{v}_1\rangle, \quad p_2 = \langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{v}_2\rangle.$$

1.2 Ujević 迭代算法的投影角度解析

下面首先整理一下 Ujević 迭代算法的推导思想. 通过变量代换和简单符号运算, 1D-DSPM 的原理可以表示为

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_1\boldsymbol{v}_1 + \beta_2\boldsymbol{v}_2, \\ f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) = -\dfrac{p_1^2}{2a} - \dfrac{(cp_1 - ap_2)^2}{2a^2d}, \quad k = 1, \cdots, n, \end{cases} \tag{1.3}$$

式中, $\alpha_1 = -\dfrac{p_1}{a}$, $\beta_2 = \dfrac{cp_1 - ap_2}{ad}$, $f(\boldsymbol{x}) = \dfrac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle - \langle \boldsymbol{b}, \boldsymbol{x}\rangle$.

在每组内循环迭代过程中 ($k = 1, \cdots, n$), 1D-DSPM 每一步选择 $\boldsymbol{v}_1 = \boldsymbol{e}_i$, $\boldsymbol{v}_2 = \boldsymbol{e}_j$, 其中 j 与 i 相关, $j \neq i$, 并同时修正近似解 $\boldsymbol{x}_k$ 的两个分量. 这样, 1D-DSPM 就可以视为 Gauss-Seidel 迭代法的一种改进修正算法. 而且, 1D-DSPM 收敛速度是 Gauss-Seidel 迭代法的 2 倍. 接下来从投影角度对 1D-DSPM 迭代求解过程的每一步进行两步分析研究.

第一步通过选取子空间 $\mathcal{K}_1 = \mathcal{L}_1 = \mathrm{span}\{\boldsymbol{v}_1\}$, $\boldsymbol{x}_0 = \boldsymbol{x}_k$, 式 (1.2) 转化为

$$\text{求}\tilde{\boldsymbol{x}}_{k+1} \in \boldsymbol{x}_k + \mathcal{K}_1, \text{ 使得 } \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}_{k+1} \perp \mathcal{L}_1, \tag{1.4}$$

其中, $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k + \tilde{\alpha}\boldsymbol{v}_1$. 采用内积形式, 式 (1.4) 可表示为

$$\langle \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}_{k+1}, \boldsymbol{v}_1\rangle = 0, \tag{1.5}$$

进一步展开可得

$$\begin{aligned} \langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \tilde{\alpha}\boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_1\rangle &= \langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k, \boldsymbol{v}_1\rangle - \tilde{\alpha}\langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_1\rangle \\ &= -p_1 - \tilde{\alpha}a \\ &= 0, \end{aligned}$$

于是可以求出 $\tilde{\alpha} = -\dfrac{p_1}{a}$, 与式 (1.3) 中的 α_1 相等.

类似地, 下一步通过选取子空间 $\mathcal{K}_2 = \mathcal{L}_2 = \text{span}\{\boldsymbol{v}_2\}$, $\boldsymbol{x}_0 = \tilde{\boldsymbol{x}}_{k+1}$, 式 (1.2) 转化为

$$\text{求}\boldsymbol{x}_{k+1} \in \tilde{\boldsymbol{x}}_{k+1} + \mathcal{K}_2,\ \text{使得}\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} \perp \mathcal{L}_2, \tag{1.6}$$

其中, $\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k+1} + \tilde{\beta}\boldsymbol{v}_2$. 采用内积形式, 式 (1.6) 可表示为

$$\langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}, \boldsymbol{v}_2 \rangle = 0, \tag{1.7}$$

进一步展开可得

$$\begin{aligned}
\left\langle \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}_{k+1} - \tilde{\beta}\boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_2 \right\rangle &= \left\langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \tilde{\alpha}\boldsymbol{A}\boldsymbol{v}_1 - \tilde{\beta}\boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_2 \right\rangle \\
&= \langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k, \boldsymbol{v}_2 \rangle - \tilde{\alpha} \langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_2 \rangle - \tilde{\beta} \langle \boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_2 \rangle \\
&= -p_2 - \tilde{\alpha}c - \tilde{\beta}d \\
&= -p_2 + \frac{cp_1}{a} - \tilde{\beta}d \\
&= 0,
\end{aligned}$$

于是可以求出 $\tilde{\beta} = \dfrac{cp_1 - ap_2}{ad}$, 与式 (1.3) 中的 β_2 相同. 经过式 (1.4) 和式 (1.6) 两步投影操作, 所得方程组 (1.1) 的下一步近似解与采用 Ujević 迭代算法求出的相同.

至此, 基于从投影技术角度的分析, 可以清楚地认识到, 1D-DSPM 属于一种特殊的投影方法.

1.3　2D-DSPM 及其与 Ujević 迭代算法理论比较结果

本节给出 2D-DSPM 的具体推导过程及其与 1D-DSPM 的理论比较结果. 与 1D-DSPM 不同的是, 在每组内循环迭代过程中 ($k = 1, \cdots, n$), 2D-DSPM 每一步迭代选取的子空间为 1D-DSPM 每一步选取的两个子空间 $\mathcal{K}_1$, $\mathcal{K}_2$ 的线性组合. 而 2D-DSPM 在投影过程的每一步仍进行两次修正. 也就是说, 强加的 Petrov-Galerkin 条件要求每一步迭代是在二维子空间 $\mathcal{K} = \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$ 中寻求近似解, 同时要求相应残量与二维子空间 $\mathcal{L} = \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$ 正交. 于是, 式 (1.2) 变为

$$\text{求}\boldsymbol{x}_{k+1} \in \boldsymbol{x}_k + \mathcal{K},\ \text{使得}\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} \perp \mathcal{L}, \tag{1.8}$$

其中, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha\boldsymbol{v}_1 + \beta\boldsymbol{v}_2$. 式 (1.8) 用内积形式可以表示为

$$\begin{cases} \langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}, \boldsymbol{v}_1 \rangle = 0, \\ \langle \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}, \boldsymbol{v}_2 \rangle = 0, \end{cases} \tag{1.9}$$

进一步展开可得

$$
\begin{cases}
\begin{aligned}
\langle \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k-\alpha\boldsymbol{A}\boldsymbol{v}_1-\beta\boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_1\rangle &= \langle \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k,\boldsymbol{v}_1\rangle-\alpha\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_1\right\rangle-\beta\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_1\right\rangle\\
&= -p_1-a\alpha-c\beta\\
&= 0,\\
\langle \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k-\alpha\boldsymbol{A}\boldsymbol{v}_1-\beta\boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\rangle &= \langle \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k,\boldsymbol{v}_2\rangle-\alpha\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\right\rangle-\beta\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\right\rangle\\
&= -p_2-c\alpha-d\beta\\
&= 0.
\end{aligned}
\end{cases}
\tag{1.10}
$$

为了求解式 (1.10) 中的系数 α 和 β, 需要下面的引理.

引理 1.1 定义于 1.1 节中的变量 a,c,d 满足下列不等式组

$$
\begin{cases}
a>0,\\
d>0,\\
ad-c^2>0.
\end{cases}
\tag{1.11}
$$

证明 正如 1.1 节所定义, $a=\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_1\rangle$, $c=\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\rangle=\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_1\rangle$, $d=\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\rangle$, 其中, $\boldsymbol{0}\neq\boldsymbol{v}_1$, $\boldsymbol{v}_2\in\mathbb{R}^n$ 且 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 线性无关. 于是根据向量内积和正定矩阵的性质, 可以容易看出前两个不等式成立. 为了证明第三个不等式成立, 首先将 $\langle \boldsymbol{A}(\boldsymbol{v}_1-\lambda\boldsymbol{v}_2),\boldsymbol{v}_1-\lambda\boldsymbol{v}_2\rangle$ 展开为

$$
\langle \boldsymbol{A}(\boldsymbol{v}_1-\lambda\boldsymbol{v}_2),\boldsymbol{v}_1-\lambda\boldsymbol{v}_2\rangle=\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_1\right\rangle-2\lambda\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\right\rangle+\lambda^2\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\right\rangle,
$$

其中, $\lambda\in\mathbb{R}$. 由于 $\boldsymbol{v}_2\neq\boldsymbol{0}$ 并且 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 线性无关, 令 $\lambda=\dfrac{\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\rangle}{\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\rangle}$, 则根据 $\langle \boldsymbol{A}(\boldsymbol{v}_1-\lambda\boldsymbol{v}_2),\boldsymbol{v}_1-\lambda\boldsymbol{v}_2\rangle>0$ 可得

$$
\begin{aligned}
0<\langle \boldsymbol{A}(\boldsymbol{v}_1-\lambda\boldsymbol{v}_2),\boldsymbol{v}_1-\lambda\boldsymbol{v}_2\rangle &= \left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_1\right\rangle-2\frac{\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\right\rangle^2}{\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\right\rangle}+\frac{\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\right\rangle^2}{\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\right\rangle}\\
&= \left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_1\right\rangle-\frac{\left\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\right\rangle^2}{\left\langle \boldsymbol{A}\boldsymbol{v}_2,\boldsymbol{v}_2\right\rangle}\\
&= a-\frac{c^2}{d},
\end{aligned}
$$

从而可知最后一个不等式成立. □

因此, 通过求解 (1.10) 中的下面两个方程

$$
\begin{cases}
-p_1-a\alpha-c\beta=0,\\
-p_2-c\alpha-d\beta=0,
\end{cases}
$$

可得

$$\begin{cases} \alpha = \dfrac{cp_2 - dp_1}{ad - c^2}, \\ \beta = \dfrac{cp_1 - ap_2}{ad - c^2}. \end{cases} \tag{1.12}$$

现在, 2D-DSPM 可以描述为算法 1.1.

算法 1.1　2D-DSPM

1. 选择方程组 (1.1) 的近似解初值 $\boldsymbol{x}_0 \in \mathbb{R}^n$
2. for $m = 1, 2, \cdots$ do
3. 　$\boldsymbol{x}_1 = \boldsymbol{x}_0$
4. 　for $k = 1, \cdots, n$ do
5. 　　$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{v}_1 + \beta \boldsymbol{v}_2$, 其中, α 和 β 如式 (1.12) 所示
6. 　end do
7. 　$\boldsymbol{x}_0 = \boldsymbol{x}_{n+1}$
8. 　验证收敛条件
9. end do

注意到, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 可以是 n 维实数空间 $\mathbb{R}^n$ 中的任意向量. 但是, 为了有效地求解实际问题, 往往需要在每一步内循环迭代时选取合理的 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$. 定理 1.4 和推论 1.8 将给出 2D-DSPM 和 1D-DSPM 在减少误差效果方面的理论比较. 推论 1.5 将给出针对特定 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 选择的理论结果. 1.4 节将给出与理论结果相对应的数值实验比较结果.

考虑如下优化问题

$$f(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle - \langle \boldsymbol{b}, \boldsymbol{x}\rangle \to \inf. \tag{1.13}$$

在给出 2D-DSPM 和 1D-DSPM 在减少误差效果方面的理论比较之前, 先给出引理 1.2 和定理 1.3. 引理 1.2 指出优化问题 (1.13) 的目标函数 f 的减少量等价于对应误差的减少量. 定理 1.3 给出采用 2D-DSPM 计算时目标函数 $f(\boldsymbol{x}_k)$ 和 $f(\boldsymbol{x}_{k+1})$ 之间的减少量.

引理 1.2　*优化问题 (1.13) 的目标函数 $f(\boldsymbol{x}_k)$ 和 $f(\boldsymbol{x}_{k+1})$ 之间的减少量与 $\boldsymbol{A}$-范数意义下误差 error $= \boldsymbol{x} - \boldsymbol{x}_*$ 的减少量等价.*

证明　结论可由以下推导过程容易证明.

$$\begin{aligned} & \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_*\|_{\boldsymbol{A}} - \|\boldsymbol{x}_k - \boldsymbol{x}_*\|_{\boldsymbol{A}} \\ = & \langle \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{A}\boldsymbol{x}_*, \boldsymbol{x}_{k+1} - \boldsymbol{x}_*\rangle - \langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{A}\boldsymbol{x}_*, \boldsymbol{x}_k - \boldsymbol{x}_*\rangle \\ = & \langle \boldsymbol{A}\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+1}\rangle - 2\langle \boldsymbol{b}, \boldsymbol{x}_{k+1}\rangle - (\langle \boldsymbol{A}\boldsymbol{x}_k, \boldsymbol{x}_k\rangle - 2\langle \boldsymbol{b}, \boldsymbol{x}_k\rangle) \\ = & 2f(\boldsymbol{x}_{k+1}) - 2f(\boldsymbol{x}_k). \end{aligned} \tag{1.14}$$

由式 (1.14) 结论成立. □

定理 1.3 采用 2D-DSPM 计算时目标函数 $f(\boldsymbol{x}_k)$ 和 $f(\boldsymbol{x}_{k+1})$ 之间的减少量为

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) = \frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)}. \tag{1.15}$$

证明 由算法 1.1 可知, 在每组内循环迭代过程中 $(k = 1, \cdots, n)$, 2D-DSPM 每一步的近似解满足 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{v}_1 + \beta \boldsymbol{v}_2$, 代入 $f(\boldsymbol{x}_{k+1})$ 可得

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &= f(\boldsymbol{x}_k + \alpha \boldsymbol{v}_1 + \beta \boldsymbol{v}_2) \\ &= f(\boldsymbol{x}_k) + \alpha \langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{v}_1 \rangle + \beta \langle \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \boldsymbol{v}_2 \rangle \\ &\quad + \frac{1}{2}\alpha^2 \langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_1 \rangle + \alpha\beta \langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_2 \rangle + \frac{1}{2}\beta^2 \langle \boldsymbol{A}\boldsymbol{v}_2, \boldsymbol{v}_2 \rangle \\ &= f(\boldsymbol{x}_k) + p_1\alpha + p_2\beta + \frac{1}{2}a\alpha^2 + c\alpha\beta + \frac{1}{2}d\beta^2. \end{aligned} \tag{1.16}$$

令

$$g(\alpha, \beta) = p_1\alpha + p_2\beta + \frac{1}{2}a\alpha^2 + c\alpha\beta + \frac{1}{2}d\beta^2.$$

为了使 $f(\boldsymbol{x}_k)$ 和 $f(\boldsymbol{x}_{k+1})$ 之间减少量达到最大化, 必须最小化以 α, β 为变量的函数 $g(\alpha, \beta)$. 由极小化的必要条件可得

$$\begin{cases} \dfrac{\partial g}{\partial \alpha} = p_1 + a\alpha + c\beta = 0, \\ \dfrac{\partial g}{\partial \beta} = p_2 + c\alpha + d\beta = 0, \end{cases}$$

其解恰如式 (1.12) 所示. 将 α, β 的表达式代入式 (1.16) 可得

$$f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k) - \frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)},$$

从而定理得证. □

下面, 从理论上比较 2D-DSPM 和 1D-DSPM 在减少误差方面的效果.

定理 1.4 2D-DSPM 比 1D-DSPM 可以更好 (至少一样) 地减少优化问题 (1.13) 中的目标函数 f; 也就是说, 2D-DSPM 比 1D-DSPM 可以更大程度 (至少一样) 地减少误差.

证明 由式 (1.3) 可知, 采用 1D-DSPM 计算时 $f(\boldsymbol{x}_k)$ 和 $f(\boldsymbol{x}_{k+1})$ 之间的减少量为

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) = \frac{p_1^2}{2a} + \frac{(cp_1 - ap_2)^2}{2a^2d}. \tag{1.17}$$

接下来所需要做的就是比较式 (1.15) 和式 (1.17). 用式 (1.15) 减式 (1.17) 可得

$$\frac{dp_1^2+ap_2^2-2cp_1p_2}{2(ad-c^2)}-\left(\frac{p_1^2}{2a}+\frac{(cp_1-ap_2)^2}{2a^2d}\right)=\frac{c^2(cp_1-ap_2)^2}{2a^2d(ad-c^2)}\geqslant 0,$$

这就证明了结论的第一部分, 而第二部分可以根据式 (1.14) 立即得证. □

推论 1.5　*如果 2D-DSPM 每一步内循环迭代中 $\langle \boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{v}_2\rangle=0$ 或 $\langle \boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b}, c\boldsymbol{v}_1-a\boldsymbol{v}_2\rangle=0$, 那么 1D-DSPM 和 2D-DSPM 减少误差的效果相同. 如果 1D-DSPM 和 2D-DSPM 的每一步内循环迭代中 $\langle \boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b},c\boldsymbol{v}_1-a\boldsymbol{v}_2\rangle=0$, 那么 1D-SSPM, 1D-DSPM 和 2D-DSPM 三种方法减少误差的效果相同, 而且 1D-DSPM 和 2D-DSPM 退化为 1D-SSPM, 即退化为 Gauss-Seidel 方法.*

证明　由定理 1.3 和定理 1.4, 可以从以下两方面证明: 一方面, 如果 $\langle \boldsymbol{A}\boldsymbol{v}_1, \boldsymbol{v}_2\rangle=0$, 即 $c=0$, 那么 1D-DSPM 和 2D-DSPM 所得 f 的减少量相同, 均为 $\frac{p_1^2}{2a}+\frac{p_2^2}{2d}$. 另一方面, 如果 $\langle \boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b},c\boldsymbol{v}_1-a\boldsymbol{v}_2\rangle=0$, 即 $cp_1-ap_2=0$, 那么 1D-SSPM, 1D-DSPM 和 2D-DSPM 三种方法所得 f 的减少量相同, 均为 $\frac{p_1^2}{2a}$. 在这种情形下, 1D-DSPM 和 2D-DSPM 就退化为 1D-SSPM, 结论得证. □

注解 1.1　*算法每一步内循环迭代中 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 的不同选择将产生不同的误差减小量. 另外, 误差减小量也依赖于 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 这两个向量之间的关于系数矩阵的相互关系, 以及当前残量和两个向量 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ 之间的关系. 这一点已一定程度地在推论 1.5 中有所反映, 并且在数值实验部分也可观察到.*

根据定理 1.4 和推论 1.5 可以看出, 2D-DSPM 比 1D-DSPM 可以更好 (至少一样) 地减少误差. 在进一步探究式 (1.15) 和式 (1.17) 两式关系之前, 先给出如下引理.

引理 1.6　*如果 $cp_1(cp_1-ap_2)\geqslant 0$ 且 $dp_1\neq cp_2$, 那么下列不等式组成立*

$$0\leqslant \frac{c^2(cp_1-ap_2)^2}{a^2(dp_1-cp_2)^2}<1. \tag{1.18}$$

证明　首先, 由于 $dp_1\neq cp_2$, 所以式 (1.18) 存在并有意义. 于是, 如果 $a^2(dp_1-cp_2)^2-c^2(cp_1-ap_2)^2>0$, 就有式 (1.18) 成立. 事实上, 通过简单计算, 可得

$$a^2(dp_1-cp_2)^2-c^2(cp_1-ap_2)^2=(ad-c^2)((ad+c^2)p_1^2-2acp_1p_2). \tag{1.19}$$

由引理 1.1 及假设条件, 有

$$\begin{cases} ad-c^2>0,\\ (ad+c^2)p_1^2-2acp_1p_2>2c^2p_1^2-2acp_1p_2=2cp_1(cp_1-ap_2)\geqslant 0. \end{cases}$$

由上面不等式组, 可知式 (1.19) 大于零, 于是结论成立. □

定理 1.7 *如果满足引理 1.6 的条件, 那么 2D-DSPM 和 1D-DSPM 在减少优化问题 (1.13) 中的目标函数 f 方面, 具有如下关系*

$$1 \leqslant \frac{\Delta f_{\text{2D-DSPM}}}{\Delta f_{\text{1D-DSPM}}} \leqslant \frac{1}{1-\dfrac{c^2(cp_1-ap_2)^2}{a^2(dp_1-cp_2)^2}}, \tag{1.20}$$

式中, $\Delta f_{\text{2D-DSPM}}$ 和 $\Delta f_{\text{1D-DSPM}}$ 分别表示式 (1.15) 和式 (1.17) 两式.

证明 根据定理 1.4 的证明过程, 可知式 (1.20) 左端不等号显然成立. 右端不等号可证明如下. 首先, 令 $\omega\Delta f_{\text{2D-DSPM}}=\Delta f_{\text{1D-DSPM}}$, 其中 ω 为标量, 具体展开有

$$\omega\frac{dp_1^2+ap_2^2-2cp_1p_2}{2(ad-c^2)}=\frac{p_1^2}{2a}+\frac{(cp_1-ap_2)^2}{2a^2d}.$$

上式变形可得

$$\omega=1-\frac{c^2(cp_1-ap_2)^2}{a^2(d^2p_1^2-2cdp_1p_2+adp_2^2)}.$$

注意到 $ad>c^2$, 于是

$$\omega\geqslant 1-\frac{c^2(cp_1-ap_2)^2}{a^2(dp_1-cp_2)^2}>0,$$

其中, 上面等号情况在推论 1.5 的条件下成立. 因此, 再根据引理 1.6, 即得结论. □

推论 1.8 *2D-DSPM 和 1D-DSPM 在减少误差方面也具有形如式 (1.20) 的关系.*

证明 由引理 1.2 及定理 1.7 直接得证. □

这样, 根据推论 1.8, 由 1D-SSPM 和 1D-DSPM 求解线性方程组 (1.1) 的收敛性就保证了 2D-DSPM 的收敛性.

1.4 数值实验

本节选用两类矩阵实例来比较本章所提出的 2D-DSPM 与 Ujević 迭代算法. 为了采用算法 1.1 迭代求解方程组 (1.1), 首先详细地给出类似于 Ujević 特定方法的一种推广算法.

选择 $\boldsymbol{v}_1=\boldsymbol{e}_i$, $\boldsymbol{v}_2=\boldsymbol{e}_j$, 其中 j 依赖于 i, 但是 $j\neq i$, $i=1,\cdots,n$. 那么算法 1.1 每一步内循环迭代中有

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_i\boldsymbol{e}_i+\beta_j\boldsymbol{e}_j,\quad k=1,\cdots,n,$$

式中