

一种基于对角化的抛物型最优控制问题的 预处理器

高 广

中国地质大学(武汉), 数学与物理学院, 湖北 武汉

收稿日期: 2024年4月29日; 录用日期: 2024年5月22日; 发布日期: 2024年5月31日

摘 要

本文研究了求解抛物型偏微分方程约束的最优控制问题, 利用对角化技巧, 提出了一个新的基于对角化的预处理器, 用于快速求解大型稀疏方程组。数值实验说明了预处理器的良好加速效果和稳定性。

关键词

预处理器, 最优控制, 抛物方程, 对角化

A Preconditioner for Parabolic Optimal Control Problems Based on Diagonalization

Guang Gao

School of Mathematics and Physics, China University of Geosciences (Wuhan), Wuhan Hubei

Received: Apr. 29th, 2024; accepted: May 22nd, 2024; published: May 31st, 2024

Abstract

In this paper, we study the solution of the optimal control problem constrained by parabolic partial differential equations. By using the diagonalization technique, we propose a new preconditioner based on diagonalization to quickly calculate the large sparse equation system. Numerical experiments demonstrate the good acceleration effect and stability of the preconditioner.

Keywords

Preconditioner, Optimal Control, Parabolic Equations, Diagonalization

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

最优控制是指在给定的约束条件下，寻求一个控制，使给定的系统性能指标达到极大值(或极小值)。这类问题广泛存在于社会问题或技术领域中。由于在绝大多数情况下，不能通过解析的方式求解最优控制问题，因此构造有效的数值计算方法成为必要。通常有两种不同的方法来处理偏微分方程约束的最优控制问题。一种是先优化后离散法，另一种是先离散后优化法，参见[1] [2] [3] [4]。本文选择先优化后离散的方法，来求解偏微分方程最优控制问题。

对于包含时间项的偏微分方程最优控制问题，状态方程随时间正向发展，伴随方程随时间反向发展。这意味着需要一次性求解正倒向两个方程耦合的大型代数方程组，因此导致计算成本很高。求解这类问题经常用到的方法是一种所谓的 ParaDiag 算法。该算法的核心步骤通常是将时间离散矩阵近似对角化，得到预处理子，然后进行分解，使得系统可以被一次性求解，以达到提高计算效率的目的，近年来该领域有许多相关的文献。这些文献中，ParaDiag 算法大致被分为两类：① ParaDiag-I：使用直接法进行求解。这类方法最早由 Maday [5]等人提出，为了保证得到时间离散化矩阵可以很好地对角化，该方法会采取变化的时间步长或是使用混合的时间离散格式[6]。② ParaDiag-II：使用迭代法进行求解。该类方法允许通过强制修改系数矩阵来使得它可以对角化，近年来被许多文献所研究[7] [8] [9] [10]。

本文结构如下：第 2 节引入研究的抛物型方程约束的最优控制问题，对该问题进行时空间方向的离散。然后提出一种预处理子，用于加速计算大型稀疏线性方程组；第 3 节是数值实验，说明提出的预处理子的良好性能；第 4 节为全文的总结。

2. 离散格式与预处理

2.1. 全离散格式

考虑如下跟踪型分布式控制问题：

$$\min J(y, u) := \frac{1}{2} \|y - g\|_{L^2(\Omega \times (0, T))}^2 + \frac{\gamma}{2} \|u\|_{L^2(\Omega \times (0, T))}^2, \quad (1)$$

使得

$$\begin{cases} y_t - Ly = u + f, & (x, t) \in \Omega \times [0, T] \\ y(x, t) = 0, & (x, t) \in \partial\Omega \times [0, T] \\ y(x, 0) = y_0, & x \in \Omega \end{cases} \quad (2)$$

其中 L 为空间算子，一般由拉普拉斯算子给出。 y 是状态变量， g 为目标函数， u 为控制变量， $\gamma > 0$ 为正则化参数。

利用一阶最优性条件，消去控制变量 u ，得到如下简化的 KKT 系统[11]

$$\begin{bmatrix} L_1 & -\gamma^{-1}I \\ I & L_2 \end{bmatrix} \begin{bmatrix} y \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \quad (3)$$

其中

$$\begin{cases} L_1 y := (\partial_t - L)y, & y(x, 0) = y_0, \\ L_2 p := (-\partial_t - L)p, & p(x, T) = 0. \end{cases} \quad (4)$$

为了求解系统(3), 对于空间离散化, 采用有限差分格式(或有限元法, 有限体积法等)。本文更加关注对时间方向离散化后的矩阵进行预处理, 因此不论使用上述的何种方法对空间进行离散化, 都可以利用空间离散矩阵 L_h 代替具体的格式。即

$$\begin{cases} y_t = f + L_h y - \gamma^{-1} p, \\ -p_t = g + L_h p - y. \end{cases}$$

时间上采用二阶精度的 Crank-Nicolson 离散格式, 于是得到离散 KKT 系统

$$\begin{bmatrix} B_1 \otimes I_x - \frac{\Delta t}{2} (B_2 \otimes L_h) & -\frac{\Delta t}{2\sqrt{\gamma}} (B_2^T \otimes I_x) \\ \frac{\Delta t}{2\sqrt{\gamma}} (B_2 \otimes I_x) & B_1^T \otimes I_x - \frac{\Delta t}{2} (B_2^T \otimes L_h) \end{bmatrix} \begin{bmatrix} \hat{Y} \\ P \end{bmatrix} = \begin{bmatrix} \hat{F} \\ G \end{bmatrix}, \quad (5)$$

其中

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_t} \end{bmatrix}, P = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{N_t-1} \end{bmatrix}, B_1 = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{bmatrix},$$

$$F = \frac{\Delta t}{2} \begin{bmatrix} 2y_0/\Delta t + L_h y_0 + (f_0 + f_1) \\ f_1 + f_2 \\ \vdots \\ f_{N_t-1} + f_{N_t} \end{bmatrix}, G = \frac{\Delta t}{2} \begin{bmatrix} -2y_0/\Delta t + (g_0 + g_1) \\ g_1 + g_2 \\ \vdots \\ g_{N_t-1} + g_{N_t} \end{bmatrix}.$$

其中 I_t 是 $N_t \times N_t$ 的单位矩阵, I_x 为 $N_x \times N_x$ 的单位矩阵, N_x 表示空间离散剖分的网格数。“ \otimes ”表示克罗内克积, 是张量积的特殊形式。

2.2. 预处理

本节使用 ParaDiag-II 方法处理大型代数系统(5), 通过对矩阵 B_1, B_2 进行修改, 使得它们成为 α -循环矩阵。众所周知, 循环矩阵都可以进行相似对角化, 具体步骤参见[12], 因此在这一类方法中, 基于对角化的预处理子被引入, 以达到加速计算的目的。 α -循环矩阵结构如下:

$$C_1 = \begin{bmatrix} 1 & & & & -\alpha \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, C_2 = \begin{bmatrix} 1 & & & & \alpha \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{bmatrix}, \quad (6)$$

$\alpha \in [-1, 1]$ 是一个常数。有关 α -循环矩阵的介绍可以参见[8], α -循环矩阵可以进行对角化分解

$$C_1 = \Lambda^{-1} \mathbb{F}^* D_1 \mathbb{F} \Lambda, \quad D_1 = \text{diag}(\lambda_{1,1}, \lambda_{1,2}, \dots, \lambda_{1,N_t}),$$

$$C_2 = \Lambda^{-1} \mathbb{F}^* D_2 \mathbb{F} \Lambda, \quad D_2 = \text{diag}(\lambda_{2,1}, \lambda_{2,2}, \dots, \lambda_{2,N_t}),$$

其中 $\Lambda = \text{diag}\left(1, \alpha^{\frac{1}{N_t}}, \alpha^{\frac{2}{N_t}}, \dots, \alpha^{\frac{N_t-1}{N_t}}\right)$, \mathbb{F} 为离散傅立叶变换矩阵, 两个对角矩阵 D_1, D_2 的特征值分别为

$$\lambda_{1,n} = 1 - \alpha^{\frac{1}{N_t}} \omega^{n-1}, \quad \lambda_{2,n} = 1 + \alpha^{\frac{1}{N_t}} \omega^{n-1}, \quad n = 1, 2, \dots, N_t. \quad (7)$$

其中 $\omega = e^{-2\pi i/N_t}$, 容易知道 \mathbb{F} 是一个酉矩阵, 即 $\mathbb{F}^* \mathbb{F} = I$, 这意味着 \mathbb{F} 的逆矩阵与其共轭转置矩阵是相等的, 观察(7)可知 $\bar{\lambda}_{1,n} = \lambda_{2,n}$. 利用[7]的结论, 可以得到 \tilde{P} 的分解

$$\tilde{P} = (V \otimes I_x) \left(D_3 \otimes I_x - \frac{\Delta t}{2} \begin{bmatrix} I_t & \\ & I_t \end{bmatrix} \otimes L_h \right) (V^{-1} \otimes I_x). \quad (8)$$

其中

$$V = \begin{bmatrix} \Lambda^{-1} \mathbb{F}^* & \\ & \Lambda^{-1} \mathbb{F}^* \end{bmatrix} \begin{bmatrix} I_t & S_2 \\ S_1 & I_t \end{bmatrix}, \quad S = \begin{bmatrix} I_t & S_2 \\ S_1 & I_t \end{bmatrix},$$

$$D_3 = \begin{bmatrix} D_2^{-1} D_1 - \frac{\Delta t}{2\sqrt{\gamma}} I_t S_1 & \\ & D_2 D_1^{-1} + \frac{\Delta t}{2\sqrt{\gamma}} I_t S_2 \end{bmatrix}.$$

$$S_1 = -\frac{\sqrt{\gamma}}{\Delta t} I_t \left(D_2 D_1^{-1} - D_2^{-1} D_1 + \sqrt{(D_2 D_1^{-1} - D_2^{-1} D_1)^2 - \frac{\Delta t^2}{\gamma}} I_t \right),$$

$$S_2 = -\frac{\sqrt{\gamma}}{\Delta t} I_t \left(D_2 D_1^{-1} - D_2^{-1} D_1 + \sqrt{(D_2 D_1^{-1} - D_2^{-1} D_1)^2 + \frac{\Delta t^2}{\gamma}} I_t \right).$$

由[13]的分析可以知道, 要成功地执行类似[9]的对角化分解, 矩阵 $C_{1,2}, C_{1,2}^*$ 必须同时都可以对角化, 即

$$\Lambda^{-1} = \Lambda^*.$$

该等式和 $|\alpha| = 1$ 是等价的, 这个限制条件使得 \tilde{P} 的形式有很大的局限性. 和[13]中的选择类似, 在本文后续的讨论中, 取 $\alpha = -1$, 此时 $C_{1,2}$ 又被称为负循环矩阵.

下面的步骤(1)~(3)给出了求解式(8)的方法.

为计算 $X_2 = \tilde{P}^{-1} r$, 有

$$\begin{aligned} \text{(a)} \quad & X_1 = (V^{-1} \otimes I_x) r, \\ \text{(b)} \quad & \left(d_n I_x - \frac{\Delta t}{2} L_h \right) w_n = X_{1,n}, \quad n = 1, 2, \dots, 2N_t, \\ \text{(c)} \quad & (V \otimes I_x) w = X_2. \end{aligned} \quad (9)$$

其中 r 表示任意的输入向量,

$$X_1 = [X_{1,1}^T, X_{1,2}^T, \dots, X_{1,2N_t}^T]^T, \quad w = [w_1^T, w_2^T, \dots, w_{2N_t}^T]^T.$$

d_n 表示对角矩阵 D_3 的第 n 行元素. 式(9)一般被称为三步对角化技巧[6]. 该方法的好处在于: 得益于矩阵 V 的特殊结构, 步骤(a)和步骤(c)可以通过快速傅立叶变换进行高效计算; 步骤(b)由 $2N$ 个独立的线性系统构成, 可以用高度并行的方式求解[14].

根据[15]的分析, 对角化步骤(a)和(c)会产生较大的舍入误差, 定理 2.1 给出了适用于本文研究问题的舍入误差分析。

定理 2.1 令 $\lambda \in \sigma(L_n)$ 为空间离散矩阵 L_n 的任意特征值, 假设(8)的步骤(b)是通过直接法求解的(例如, LU 分解法)。设机器精度为 ε , 那么舍入误差为

$$\frac{\|\hat{\eta}_\lambda - \tilde{\eta}_\lambda\|_2}{\|\hat{\eta}_\lambda\|_2} \leq \varepsilon(4N_t + 1)\text{Cond}_2(\Sigma(\lambda))(\text{Cond}_2(S))^2 + O(\varepsilon^2). \quad (10)$$

其中 $\hat{\eta}_\lambda$ 为如下方程(11)中对任一输入向量 \hat{r}_λ 的精确解, $\tilde{\eta}_\lambda$ 为方程(11)的数值近似解。

$$\tilde{P}_\lambda \hat{\eta}_\lambda := \left(\begin{array}{cc} C_2^{-1}C_1 & -\frac{\Delta t}{2\sqrt{\gamma}} \\ \frac{\Delta t}{2\sqrt{\gamma}} & (C_2^{-1}C_1)^\top \end{array} - \frac{\Delta t}{2} \begin{bmatrix} I_t & \\ & I_t \end{bmatrix} \right) \hat{\eta}_\lambda = \hat{r}_\lambda, \quad (11)$$

证明: 对于方程(11), 根据[7]的向后误差分析, 通过对角化技巧(2.4)得到的解满足如下的扰动系统:

$$(V + \delta V_1)g_\lambda = \hat{r}_\lambda, (\Sigma(\lambda) + \delta\Sigma)w_\lambda = g_\lambda, (V^{-1} + \delta V_2)\tilde{\eta}_\lambda = w_\lambda,$$

其中 δV_1 , $\delta\Sigma$ 和 δV_2 表示矩阵 V , D_3 和 V^{-1} 的微小扰动。由[7], 可得

$$\begin{aligned} \|\delta V_1\|_2 &\leq 2\varepsilon N \|V\|_2 + O(\varepsilon^2), \\ \|\delta V_2\|_2 &\leq 2\varepsilon N \|V^{-1}\|_2 + O(\varepsilon^2), \\ \|\delta\Sigma(\lambda)\|_2 &\leq \varepsilon \|\Sigma(\lambda)\|_2 + O(\varepsilon^2). \end{aligned} \quad (12)$$

最后一个不等式成立的原因是 $\Sigma(\lambda)$ 是一个对角矩阵。注意到通过对角化解方程(11)等价于精确求解 $(\tilde{P}_\lambda + \delta\tilde{P}_\lambda)\tilde{\eta}_\lambda = \hat{r}_\lambda$, 这里 $\delta\tilde{P}_\lambda$ 表示一个合适的扰动。实际上, 有

$$(V + \delta V_1)(\Sigma(\lambda) + \delta\Sigma(\lambda))(V^{-1} + \delta V_2)\tilde{\eta}_\lambda = \hat{r}_\lambda, \quad (13)$$

结合(12)式和(13)可以对 $\delta\tilde{P}_\lambda$ 做出如下的估计:

$$\|\delta\tilde{P}_\lambda\|_2 \leq \varepsilon(4N_t + 1)\|V\|_2\|V^{-1}\|_2\|\Sigma(\lambda)\|_2 + O(\varepsilon^2). \quad (14)$$

根据[16], 下列不等式成立

$$\frac{\|\hat{\eta}_\lambda - \tilde{\eta}_\lambda\|_2}{\|\hat{\eta}_\lambda\|_2} \leq \text{Cond}_2(\tilde{P}_\lambda) \frac{\|\delta\tilde{P}_\lambda\|_2}{\|\tilde{P}_\lambda\|_2},$$

结合(14)可以得到:

$$\begin{aligned} \frac{\|\hat{\eta}_\lambda - \tilde{\eta}_\lambda\|_2}{\|\hat{\eta}_\lambda\|_2} &\leq \varepsilon(4N_t + 1)\|V\|_2\|V^{-1}\|_2\|\tilde{P}_\lambda^{-1}\|_2\|\Sigma(\lambda)\|_2 + O(\varepsilon^2) \\ &= \varepsilon(4N_t + 1)\text{Cond}_2(V)\|\tilde{P}_\lambda^{-1}\|_2\|\Sigma(\lambda)\|_2 + O(\varepsilon^2). \end{aligned}$$

由于 $V = \begin{bmatrix} \Lambda_\alpha^{-1}\mathbb{F}^* \\ \Lambda_\alpha^{-1}\mathbb{F}^* \end{bmatrix} S$, 且 $\tilde{P}_\lambda = V\Sigma(\lambda)V^{-1}$, 结合 \mathbb{F} 是酉矩阵, 可以得到

$$\|V\|_2 \leq \|\Lambda_\alpha^{-1}\|_2\|\mathbb{F}^*\|_2\|S\|_2 \leq \max\left\{1, |\alpha|^{\frac{N_t-1}{N_t}}\right\} \frac{1}{\sqrt{N_t}}\|S\|_2,$$

同样地, 有

$$\|V^{-1}\|_2 \leq \|\Lambda_\alpha\|_2 \|\mathbb{F}\|_2 \|S^{-1}\|_2 \leq \max\left\{1, |\alpha|^{-\frac{N_t-1}{N_t}}\right\} \frac{1}{\sqrt{N_t}} \|S^{-1}\|_2,$$

于是

$$\text{Cond}_2(V) \leq \text{Cond}_2(S),$$

以及

$$\|\tilde{P}_\lambda^{-1}\|_2 \leq \|S \Sigma^{-1}(\lambda) S^{-1}\|_2 \leq \text{Cond}_2(S) \|\Sigma^{-1}(\lambda)\|_2.$$

综合上述讨论, 便得到了不等式(10). ■

上述定理表明, 由(9)带来的舍入误差主要与三个因素有关, 即时间离散的节点数 N_t , 矩阵 S 的条件数以及不可避免的, 对角矩阵 $\Sigma(\lambda)$ 的条件数。而(10)中 $\Sigma(\lambda)$ 的条件数这一因素无法进一步改进。于是本文考虑对 \tilde{P} 进行修改, 以使得矩阵 S 的条件数尽可能接近于或是等于一个网格无关的常数。

容易知道, 消除 $\text{Cond}_2(S)$ 等价于令下列矩阵为零,

$$D_2^{-1}D_1 - D_1^{-1}D_2.$$

用特征值来表征矩阵 $D_2^{-1}D_1$ 和 $D_1^{-1}D_2$, 即

$$\begin{cases} \lambda(D_2^{-1}D_1) = \begin{matrix} \lambda_{1,n} \\ \lambda_{2,n} \end{matrix} = \frac{1 - \alpha^{\frac{1}{N_t}} \omega^{n-1}}{1 + \alpha^{\frac{1}{N_t}} \omega^{n-1}}, \\ \lambda(D_1^{-1}D_2) = \begin{matrix} \lambda_{2,n} \\ \lambda_{1,n} \end{matrix} = \frac{1 + \alpha^{\frac{1}{N_t}} \omega^{n-1}}{1 - \alpha^{\frac{1}{N_t}} \omega^{n-1}}, \end{cases} \quad n = 1, 2, \dots, N_t. \quad (15)$$

取

$$\rho_n = \frac{-2\alpha^{\frac{1}{N_t}}}{\omega^{-(n-1)} - \alpha^{\frac{2}{N_t}} \omega^{n-1}}, \quad \rho = \text{diag}(\rho_1, \rho_2, \dots, \rho_{N_t}),$$

于是有

$$(D_2^{-1}D_1 - \rho I_t) - (D_1^{-1}D_2 + \rho I_t) = O. \quad (16)$$

因此, 可以用矩阵 $(D_2^{-1}D_1 - \rho I_t)$ 代替 $D_2^{-1}D_1$, 类似地, 用矩阵 $(D_1^{-1}D_2 + \rho I_t)$ 代替 $D_1^{-1}D_2$ 。如此一来, 就可以得到

$$\begin{cases} \tilde{S}_1 = \frac{1}{2} \cdot \frac{-2\sqrt{\gamma}}{\Delta t} \cdot \frac{2i\Delta t}{2\sqrt{\gamma}} I_t = -iI_t, \\ \tilde{S}_2 = -\frac{1}{2} \cdot \frac{2\sqrt{\gamma}}{\Delta t} \cdot \frac{2i\Delta t}{2\sqrt{\gamma}} I_t = -iI_t, \end{cases} \quad (17)$$

于是

$$\tilde{S}\tilde{S}^* = \begin{bmatrix} I_t & \tilde{S}_2 \\ \tilde{S}_1 & I_t \end{bmatrix} \begin{bmatrix} I_t & \tilde{S}_1^* \\ \tilde{S}_2^* & I_t \end{bmatrix} = 2 \begin{bmatrix} I_t & \\ & I_t \end{bmatrix}.$$

矩阵 \tilde{S} 的条件数为

$$\text{Cond}_2(\tilde{S}) := \|S\|_2 \|S^{-1}\|_2 = \sqrt{2} \left\| \frac{1}{\sqrt{2}} \tilde{S} \right\|_2 \left\| \frac{1}{\sqrt{2}} \tilde{S}^* \right\|_2 = \left\| \frac{1}{\sqrt{2}} \tilde{S} \right\|_2 \left\| \frac{1}{\sqrt{2}} \tilde{S}^* \right\|_2 = 1.$$

这样一来，便消除了和舍入误差有关的 $\text{Cond}_2(S)$ 这一因素，导出了预处理子 \hat{P} ：
令

$$\tilde{D}_3 = \begin{bmatrix} D_2^{-1} D_1 - \rho I_t - \frac{\Delta t}{2\sqrt{\gamma}} I_t \tilde{S}_1 & \\ & D_2 D_1^{-1} + \rho I_t + \frac{\Delta t}{2\sqrt{\gamma}} I_t \tilde{S}_2 \end{bmatrix},$$

预处理子 \hat{P} 可以表示为

$$\hat{P} = (V \otimes I_x) \left(\tilde{D}_3 \otimes I_x - \frac{\Delta t}{2} \begin{bmatrix} I_t & \\ & I_t \end{bmatrix} \otimes L_h \right) (V^{-1} \otimes I_x). \quad (18)$$

值得注意的是，在消去舍入误差中 $\text{Cond}_2(S)$ 这一项时， $\text{Cond}_2(\Sigma(\lambda))$ 同样会受到一定的影响。令

$$\tilde{\Sigma}(\lambda) = \tilde{D}_3 - \frac{\lambda \Delta t}{2} \begin{bmatrix} I_t & \\ & I_t \end{bmatrix},$$

因为 $\lambda \in \sigma(L_h)$ 是实对称矩阵的特征值，故 $\lambda = \bar{\lambda}$ ，可以使用矩阵的奇异值来估计 $\tilde{\Sigma}(\lambda)$ 以及 $\Sigma(\lambda)$ 的条件数，即

$$\text{Cond}_2(\Sigma(\lambda)) = \frac{\sigma_{\max}(\Sigma(\lambda))}{\sigma_{\min}(\Sigma(\lambda))} = \sqrt{\frac{\lambda_{\max}(\Sigma(\lambda)\Sigma^*(\lambda))}{\lambda_{\min}(\Sigma(\lambda)\Sigma^*(\lambda))}},$$

同样有

$$\text{Cond}_2(\tilde{\Sigma}(\lambda)) = \frac{\sigma_{\max}(\tilde{\Sigma}(\lambda))}{\sigma_{\min}(\tilde{\Sigma}(\lambda))} = \sqrt{\frac{\lambda_{\max}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))}{\lambda_{\min}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))}}.$$

其中 $\sigma_{\max}(\bullet), \sigma_{\min}(\bullet)$ 分别表示最大奇异值和最小奇异值。

观察 $\Sigma(\lambda)$ 可知，它也是一个对角矩阵，因此特征值可以被对角线元素表征。考虑矩阵 $\Sigma(\lambda), \tilde{\Sigma}(\lambda)$ 的前 N_t 行，有

$$\begin{aligned} \lambda_n(\Sigma(\lambda)\Sigma^*(\lambda)) &= \left(\lambda_n(D_2^{-1}D_1) - \frac{\Delta t}{2\sqrt{\gamma}} \lambda_n(S_1) - \frac{\lambda \Delta t}{2} \right) \left(\lambda_n(D_1^{-1}D_2) - \frac{\Delta t}{2\sqrt{\gamma}} \lambda_n^*(S_1) - \frac{\lambda \Delta t}{2} \right), \\ \lambda_n(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda)) &= \left(\lambda_n(D_2^{-1}D_1) + \frac{i\Delta t}{2\sqrt{\gamma}} - \frac{\lambda \Delta t}{2} - \rho_n \right) \left(\lambda_n(D_1^{-1}D_2) - \frac{i\Delta t}{2\sqrt{\gamma}} - \frac{\lambda \Delta t}{2} - \rho_n^* \right), \end{aligned}$$

$\lambda_n(\bullet)$ 表示第 n 个特征值， $n = 1, 2, \dots, N_t$ 。

令

$$\Delta \kappa = \text{Cond}_2^2(\tilde{\Sigma}(\lambda)) - \text{Cond}_2^2(\Sigma(\lambda)),$$

若

$$\frac{\lambda_{\min}(\Sigma(\lambda)\Sigma^*(\lambda))}{\lambda_{\min}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))} < \sqrt{\frac{\lambda_{\max}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))}{\lambda_{\max}(\Sigma(\lambda)\Sigma^*(\lambda))}},$$

则有如下的不等式估计

$$\Delta\kappa = \frac{\left| \frac{\lambda_{\max}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))}{\lambda_{\min}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))} - \frac{\lambda_{\max}(\Sigma(\lambda)\Sigma^*(\lambda))}{\lambda_{\min}(\Sigma(\lambda)\Sigma^*(\lambda))} \right|}{\left| \frac{\lambda_{\max}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))}{\lambda_{\min}(\tilde{\Sigma}(\lambda)\tilde{\Sigma}^*(\lambda))} - \frac{\lambda_{\max}(\Sigma(\lambda)\Sigma^*(\lambda))}{\lambda_{\min}(\Sigma(\lambda)\Sigma^*(\lambda))} \right|},$$

经过一系列冗杂但简单的运算可以得到

$$\Delta\kappa < \left| \frac{1 + \alpha \frac{2}{N_t} \omega^{2(n-1)}}{1 - \alpha \frac{2}{N_t} \omega^{2(n-1)}} - i \frac{\Delta t}{\sqrt{\gamma}} + \frac{\Delta t^2}{4\gamma} \right| \quad (19)$$

$$< \frac{\Delta t^2}{4\gamma} + c \left(\alpha, \frac{\Delta t}{\sqrt{\gamma}} \right).$$

其中 $c \left(\alpha, \frac{\Delta t}{\sqrt{\gamma}} \right)$ 是一个与 α 和 $\frac{\Delta t}{\sqrt{\gamma}}$ 有关的正数。于是可以知道, 修改 \tilde{P} 确实会增加 $\text{Cond}_2(\Sigma(\lambda))$, 但当时间方向的网格步长的平方 Δt^2 和正则化参数 γ 保持在同一数量级时, $\Delta\kappa$ 会有一个比较稳定的上界。

3. 数值算例

本节将使用带预处理子的 GMRES 方法来求解给定的抛物型方程最优控制问题。设置停机标准为 $\varepsilon = 10^{-6}$, 采用的是右预处理 GMRES 方法, 迭代的最大步数设置为 $\text{IterMax} = 400$ 。空间离散算子 L_h 由二阶中心差分格式导出, 它是一个实对称矩阵。本节的实验目的是, 针对具体的问题, 在不同的正则化参数 γ 以及时空间网格大小的选取下, 获得程序输出的结果。输出的结果包括了 GMRES 求解器完成计算所需要的迭代次数, 记为 Iter 、求解过程中计算机耗费的时间(单位为 s), 记为 CPU 、状态变量的误差估计范数 e_y 。此外, 作为对比, 将直接使用 GMRES 求解器而非使用预处理子来计算同样的算例。

例 1 给出了一维情况下抛物型 PDE 最优控制问题, 数值结果如表 1 所示。 $N_t = 64, N_x = 64, \gamma = 10^{-4}$ 时真解和数值解的对比图如图 1 和图 2 所示。

例 1 令空间域 $\Omega = (0,1)$, 时间域 $T = (0,1)$, 方程组的解析解等已知条件如下:

$$\begin{cases} y = \sin(\pi x) \cos(\pi t), \\ p = (t-1)^2 \sin(\pi x), \\ f = -\pi \sin(\pi x) \sin(\pi t) - \pi^2 \sin(\pi x) \cos(\pi t) - (t-1)^2 \sin(\pi x) / \gamma, \\ g = 2(t-1) \sin(\pi x) + \pi^2 \sin(\pi x) (t-1)^2 + \sin(\pi x) \cos(\pi t). \end{cases} \quad (20)$$

通过表 1, 可以看到, 对于 GMRES 方法所需的迭代次数, 无论正则化参数 γ 取到何值, 预处理过程都会大大减少迭代次数, 使得 GMRES 求解过程的迭代次数保持在一个很少的范围。对于计算时间, 如果正则化参数 γ 的取值不过分的小, 即 $\gamma \geq 10^{-4}$ 时, 不使用预处理方法的 GMRES 求解器所耗费的计算成本都远远大于经过预处理后的 GMRES 求解器, 而在 $\gamma = 10^{-6}$ 以及 $\gamma = 10^{-8}$ 的情况下, 使用预处理子与否对计算耗费的时间影响并没有显著差异, 甚至网格较粗的情况下不使用预处理的 GMRES 方法在时间成本上稍显优势。对于此现象的解释是, 正则化参数 γ 与时空间网格步长的比 $\Delta t / \Delta x^2$ 之间的数量级有差异。如果增加网格大小, 以使得 $\sqrt{\gamma}$ 和 $\Delta t / \Delta x^2$ 尽量接近, 那么上述现象便不会再发生。为尝试验证这一解释,

在 $\gamma = 10^{-6}$ 和 $\gamma = 10^{-8}$ 的情况下使用了更加细化的网格，大小为 500×500 ，作为追加的实验。如表 1 中 Add1 和 Add2 的两行数据所示，当 $\sqrt{\gamma}$ 和 $\Delta t/\Delta x^2$ 比较接近时，施加了预处理过程的 GMRES 方法的数值表现明显更为优秀，即 Add1 中的结果；而 $\sqrt{\gamma}$ 和 $\Delta t/\Delta x^2$ 不那么接近时，两者在计算时间上的差异变得较小，即 Add2 中的结果。由此可以得到实验结果一定程度上与上述解释相符合。

Table 1. The number of iterations of the GMRES method in Example 1, the CPU time, and the error before and after using the preconditioner

表 1. 使用预处理子前后，例 1 的 GMRES 方法迭代所需的次数、CPU 耗费时间以及误差

γ	(N_t, N_x)	使用预处理子			未使用预处理子		
		Iter	CPU	e_y	Iter	CPU	e_y
1	(64, 64)	3	0.07	6.2e-01	209	0.56	6.2e-01
	(128, 128)	3	0.28	6.3e-01	>400	—	—
	(256, 256)	3	1.71	6.4e-01	>400	—	—
10^{-2}	(64, 64)	4	0.08	7.5e-01	214	0.58	7.5e-01
	(128, 128)	4	0.35	7.6e-01	>400	—	—
	(256, 256)	4	1.91	7.6e-01	>400	—	—
10^{-4}	(64, 64)	5	0.13	8.8e-01	186	0.44	8.8e-01
	(128, 128)	5	0.45	8.9e-01	400	6.10	8.9e-01
	(256, 256)	6	2.34	8.9e-01	>400	—	—
10^{-6}	(64, 64)	3	0.08	7.2e-01	76	0.08	7.2e-01
	(128, 128)	4	0.34	7.2e-01	149	0.91	7.2e-01
	(256, 256)	5	2.15	7.3e-01	302	11.30	7.3e-01
Add1	(500, 500)	6	26.95	7.3e-01	>400	—	—
	(64, 64)	3	0.07	7.0e-01	30	0.02	7.0e-01
10^{-8}	(128, 128)	3	0.28	7.0e-01	50	0.16	7.0e-01
	(256, 256)	3	1.66	7.1e-01	87	1.75	7.1e-01
Add2	(500, 500)	3	23.86	7.1e-01	162	36.95	7.1e-01

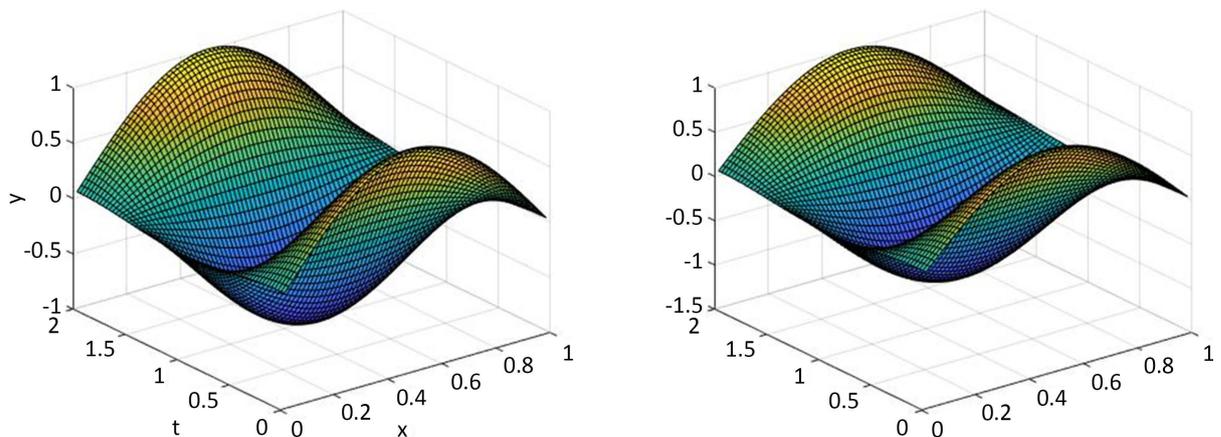


Figure 1. Comparison of the real solution (left) and the numerical solution (right) of y
图 1. y 的真实解(左)和数值解(右)对比图

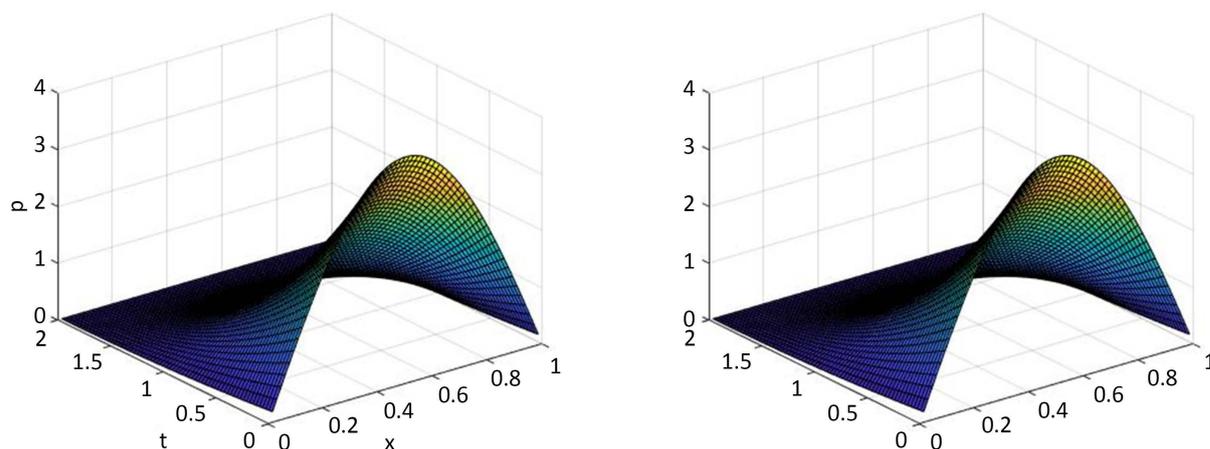


Figure 2. Comparison of the real solution (left) and the numerical solution (right) of p
图 2. p 的真实解(左)和数值解(右)对比图

4. 总结

本文研究了一种应用于初值型抛物型方程最优控制问题的预处理子, 用于快速求解给定的模型问题。首先, 给出了受抛物型方程约束的控制问题及初值条件, 得到 KKT 系统; 然后在时间方向上采用 Crank-Nicolson 格式, 空间方向上使用离散拉普拉斯算子表示空间离散后的格式, 得到了一个大型线性方程组。对离散后得到的大型稀疏代数系统, 利用三步对角化技巧快速求解。为了消除对角化步骤中矩阵 S 的条件数这一因素, 提出了一个新的预处理子。本文采用 ParaDiag-II 方案时, 使用了广义极小残量法 (GMRES) 来作为迭代求解器, 并分析了应用预处理子时, GMRES 方法的收敛性。最后的数值实验, 给出了一维的具体数值算例, 数值结果表明提出的新的预处理子确实有较为优秀的加速效果和稳定性。

致 谢

作者非常感谢各位老师给出的宝贵意见和建议。

参考文献

- [1] Kunisch, K. and Rund, A. (2015) Time Optimal Control of the Monodomain Model in Cardiac Electrophysiology. *IMA Journal of Applied Mathematics*, **80**, 1664-1683. <https://doi.org/10.1093/imamat/hxv010>
- [2] Bryson, A.E. (1996) Optimal control-1950 to 1985. *IEEE Control Systems Magazine*, **16**, 26-33. <https://doi.org/10.1109/37.506395>
- [3] Sargent, R.W.H. (2000) Optimal Control. *Journal of Computational and Applied Mathematics*, **124**, 361-371. [https://doi.org/10.1016/S0377-0427\(00\)00418-0](https://doi.org/10.1016/S0377-0427(00)00418-0)
- [4] Liu, J. and Wang, Z. (2019) Non-Commutative Discretize-Then-Optimize Algorithms for Elliptic PDE-Constrained Optimal Control Problems. *Journal of Computational and Applied Mathematics*, **362**, 596-613. <https://doi.org/10.1016/j.cam.2018.07.028>
- [5] Maday, Y. and Rønquist, E.M. (2008) Parallelization in Time through Tensor-Product Space-Time Solvers. *Comptes Rendus. Mathématique*, **346**, 113-118. <https://doi.org/10.1016/j.crma.2007.09.012>
- [6] Gander, M.J., Liu, J., Wu, S.L., et al. (2020) Paradiag: Parallel-in-Time Algorithms Based on the Diagonalization Technique. arXiv: 2005.09158.
- [7] Golub, G.H., Van Loan, C.F. (2013) Matrix Computations. JHU Press, Baltimore. <https://doi.org/10.56021/9781421407944>
- [8] Saad, Y. (2003) Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia. <https://doi.org/10.1137/1.9780898718003>
- [9] Wu, S.L. and Zhou, T. (2020) Diagonalization-Based Parallel-in-Time Algorithms for Parabolic PDE-Constrained Op-

timization Problems. *ESAIM: Control, Optimisation and Calculus of Variations*, **26**, 88.

<https://doi.org/10.1051/cocv/2020012>

- [10] Liu, J., Wu, S.L. (2020) A Fast Block α -Circulant Preconditioner for All-at-Once Systems from Wave Equations. *SIAM Journal on Matrix Analysis and Applications*, **41**, 1912-1943. <https://doi.org/10.1137/19M1309869>
- [11] Lin, X.L. and Wu, S.L. (2021) A Parallel-in-Time Preconditioner for Crank-Nicolson Discretization of a Parabolic Optimal Control Problem. arXiv: 2109.12524.
- [12] Gander, M.J. and Palitta, D. (2024) A New Paradiag Time-Parallel Time Integration Method. *SIAM Journal on Scientific Computing*, **46**, A697-A718. <https://doi.org/10.1137/23M1568028>
- [13] Bouillon, A., Samaey, G. and Meerbergen, K. (2023) On Generalized Preconditioners for Time-Parallel Parabolic Optimal Control. arXiv: 2302.06406.
- [14] McDonald, E., Pestana, J. and Wathen, A. (2018) Preconditioning and Iterative Solution of All-at-Once Systems for Evolutionary Partial Differential Equations. *SIAM Journal on Scientific Computing*, **40**, A1012-A1033. <https://doi.org/10.1137/16M1062016>
- [15] Wu, S.L., Zhou, T. and Zhou, Z. (2022) A Uniform Spectral Analysis for a Preconditioned All-at-Once System from First-Order and Second-Order Evolutionary Problems. *SIAM Journal on Matrix Analysis and Applications*, **43**, 1331-1353. <https://doi.org/10.1137/21M145358X>
- [16] Liu, J., Wang, X.S., Wu, S.L., *et al.* (2022) A Well-Conditioned Direct PinT Algorithm for First-and Second-Order Evolutionary Equations. *Advances in Computational Mathematics*, **48**, 16. <https://doi.org/10.1007/s10444-022-09928-4>