Journal of Computational and Applied Mathematics 235 (2011) 5078–5094

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A Schwarz domain decomposition method with gradient projection for optimal control governed by elliptic partial differential equations

Huibin Chang, Danping Yang *

Department of Mathematics, East China Normal University, Shanghai, 200241, PR China

ARTICLE INFO

Article history:
Received 29 January 2010
Received in revised form 21 April 2011

MSC:
49M15
49M25
65N30
65N55

Keywords:
Distributed optimal control
Elliptic partial differential equation
Domain decomposition
Schwarz alternating method
Gradient projection algorithm
Geometric convergence rate

ABSTRACT

A domain decomposition method (DDM) is presented to solve the distributed optimal control problem. The optimal control problem essentially couples an elliptic partial differential equation with respect to the state variable and a variational inequality with respect to the constrained control variable. The proposed algorithm, called SA–GP algorithm, consists of two iterative stages. In the inner loops, the Schwarz alternating method (SA) is applied to solve the state and co-state variables, and in the outer loops the gradient projection algorithm (GP) is adopted to obtain the control variable. Convergence of iterations depends on both the outer and the inner loops, which are coupled and affected by each other. In the classical iteration algorithms, a given tolerance would be reached after sufficiently many iteration steps, but more iterations lead to huge computational cost. For solving constrained optimal control problems, most of the computational cost is used to solve PDEs. In this paper, a proposed iterative number independent of the tolerance is used in the inner loops so as to save a lot of computational cost. The convergence rate of L^2-error of control variable is derived. Also the analysis on how to choose the proposed iteration number in the inner loops is given. Some numerical experiments are performed to verify the theoretical results.

1. Introduction

The optimal control problems (OCPs) that are governed by partial differential equations (PDEs) arise naturally in many areas of science and engineering. Generally, these problems result in large scale systems. The size, complexity and high-dimensional nature of PDE-governed optimal control problems present significant challenges for general-purpose numerical algorithms. These features often require iterative solvers, preconditions, and parallel implementations. As one of the most fast and efficient numerical methods for solving the large scale PDE systems, domain decomposition methods based upon overlapping and non-overlapping have been widely studied. The great advantage of domain decomposition methods is to provide bases of parallel and fast computation. There has been so extensive research on domain decomposition algorithms for PDEs in the scientific literature that it is simply impossible to give a very brief review here and to list related references. Refs. [1–5] provided a good survey of the fields and detailed references.

In contrast to the large body of work on parallel and domain decomposition methods for PDEs, few have been published on parallel algorithms for OCPs governed by PDEs. Generally speaking, those domain decomposition methods which are efficient for PDEs and their related analyses are not directly applied to OCPs governed by PDEs due to some special computational and theoretical difficulties. In unconstrained cases, OCPs could be transformed to nonsymmetric or indefinite PDE systems. Several precondition methods and domain decomposition methods were proposed, for example, [6–22] for

* Corresponding author. Tel.: +86 21 52100198; fax: +86 21 54342609.
E-mail addresses: juyuefo@student.ecnu.edu.cn (H. Chang), dpyang@math.ecnu.edu.cn (D. Yang).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.04.037

problems governed by elliptic PDEs and [23–26] for problems governed by parabolic PDEs. In constrained cases, OCPs are the coupled systems of PDEs with respect to the state and co-state variables and the inequalities with respect to the control variable. There are few researches on DDMs for the constrained optimal control problems. In [9–11], non-overlapping domain decomposition algorithms were proposed and analyzed, in which a global system was decomposed into several local sub-systems connected with Robin–Robin boundary conditions defined on inner boundaries of sub-domains. In [12,20], additive or alternating Schwarz methods based upon overlapping domain decompositions were proposed to solve PDEs of the state variable. To realize highly efficient computation for solving nonlinear, nonsymmetric or indefinite PDE systems, decoupling of the variables was used. Thus the iterations of two stages were introduced. One for solving the control variables, which was called as outer loops, while the other for solving PDEs with respect to the state variable and the co-state variable, which was called as the inner loops. In [12], Bounaïm applied this idea to a boundary control problem, where additive or alternating Schwarz methods were used, as solver or preconditioner, to solve PDEs at each step of the outer loops. He performed some numerical experiments and compared numerical results but did not give any theoretical analysis. In his numerical experiments, iterations in the inner loops were controlled by a given tolerance.

Up to now, most of existing researches focused on unconstrained problems. But we concern with the parallel algorithms for solving the optimal control problem governed by PDEs with control constraints. For the decoupling algorithms, one desires to use iteration algorithms with fast convergence rate in both of the inner and outer loops, so that total computational cost is much less. The purpose of this article is to analyze how to choose suitable iteration number in inner loops. As an example, we study the SA–GP algorithm similarly to the idea given in [12]. The gradient projection algorithm (GP), as the outer loops, is used to solve the variational inequality with respect to the control variable, and Schwarz alternating methods (SA) based on an overlapping domain decomposition are used to solve PDEs coupled with the state and the co-state variables as the inner loops. For the classical Schwarz alternating algorithm and its analyses, let us rapidly mention [27–29,1–3]. Iteration errors and computational cost in SA–GP come from both of the outer loops and the inner loops, which are coupled and affected by each other. It follows from Lions' theory that the iteration error in the inner loops arising from Schwarz alternating method would reach any given tolerance if the iteration number is sufficiently large. That is to say, the iterative number depends on the given tolerance. The smaller is the tolerance, the larger is the iterative number. On one hand, to ensure the convergence of the outer loops, more iterations in the inner loops are needed. Thus that leads to the huge total computational cost. However, that may be a great waste because the error in the outer loops may be still large. On the other hand, the outer iteration will diverge if few iterations are adopted in the inner loops. By reasons presented above it is critical to determine suitable iteration number in the inner loops. We will analyze how to choose suitable iteration number. This number is determined by the compressible factor of the outer loops and other parameters about the domain, but independent of the control tolerance in the inner loops. There is no need to stop the inner iterations until the error of the inner iteration reaches the given tolerance. The proposed algorithm guarantees convergence with lower computation cost.

The article is organized as follows. We first give the model problems in Section 2 and then state SA–GP algorithms for both of the continuous model and the discrete problem in Section 3. The most important part is Section 4, in which the convergence rate of SA–GP algorithm is analyzed and derived. Particularly, we discuss how to choose suitable iteration number in the inner loops and how to estimate some important parameters, which are useful in practical computation. Finally, in Section 5, some numerical experiments are performed to verify the theoretical results. In these tests, a given tolerance is used to control errors in the outer iteration. We adopt two methods to control inner iterations at each outer loop. One is a proposed inner iterative number given in Section 4. The other is a given tolerance to control the inner iteration as usual. The numerical results show that the total iteration number of the first method is much less than that of the second, saving almost one half of computational cost.

2. Model problem and optimality condition

Let Ω be a bounded open domain in $\mathbb{R}^d$, for $1 \le d \le 3$, with a boundary Γ. Define an objective functional

$$\mathcal{J}(u, y) = \frac{1}{2}\int_\Omega (y - y_d)^2 + \frac{\alpha}{2}\int_\Omega u^2,$$

where α is a positive constant. We investigate the following distributed optimal control problem:

$$\begin{cases} \min\limits_{u\in\mathcal{U}_{\rm ad}} & \mathcal{J}(u, y) \\ \text{s.t.} & -\Delta y = u + f \quad \text{in } \Omega,\ y = 0 \text{ on } \partial\Omega, \end{cases} \tag{2.1}$$

where $\mathcal{U}_{\rm ad}$ is a closed convex subset of $U \triangleq L^2(\Omega)$, u is the control variable, $y \in V \triangleq H_0^1(\Omega)$ is the state variable and y_d is the desired state given in $L^2(\Omega)$. By the standard theory of the optimal control, for example, see [30], it is well known that a pair $(u, y) \in U \times V$ is a solution if and only if there exists a co-state variable $p \in V$ such that

$$\begin{cases} -\Delta y = u + f & \text{in } \Omega, & \text{(state equation)} \\ -\Delta p = y - y_d & \text{in } \Omega, & \text{(co-state equation)} \\ \displaystyle\int_\Omega (p + \alpha u)(v - u)\mathrm{d}x \ge 0, & \forall v \in \mathcal{U}_{\rm ad} & \text{(optimality condition).} \end{cases} \tag{2.2}$$

Define operator $\mathcal{Q} : V \to \mathcal{U}_{\text{ad}}$ such that for each w in V,

$$\int_\Omega (\mathcal{Q}w - w)(v - \mathcal{Q}w) \geq 0, \quad \forall v \in \mathcal{U}_{\text{ad}}.$$

Then the optimality condition reduces to $\alpha u = -\mathcal{Q}p$. In order to get the weak formulation of the optimal control problem, denote the L^2-inner product in $L^2(\Omega)$ by

$$(v, w) = \int_\Omega vw, \quad \forall v, w \in L^2(\Omega)$$

and the bi-linear form

$$a(w, v) \triangleq \int_\Omega \nabla w \cdot \nabla v, \quad \forall v, w \in H_0^1(\Omega)$$

with the corresponding norms and semi-norms as follow:

$$\|v\|_0 \triangleq \sqrt{(v, v)}, \qquad |v|_1 \triangleq \sqrt{a(v, v)}, \qquad \|v\|_1 \triangleq \sqrt{a(v, v) + (v, v)}.$$

The weak form of the optimal control problem (2.1) reads:

$$\begin{cases} \min\limits_{u \in \mathcal{U}_{\text{ad}}} & \mathcal{J}(u, y) \\ \text{s.t.} & a(y, w) = (u + f, w), \quad \forall w \in V. \end{cases} \tag{2.3}$$

The equivalent optimality condition is

$$\begin{cases} a(y, w) = (u + f, w), & \forall w \in V, \\ a(q, p) = (y - y_d, q), & \forall q \in V, \\ (p + \alpha u, v - u) \geq 0, & \forall v \in \mathcal{U}_{\text{ad}}. \end{cases} \tag{2.4}$$

In next sections, we will discuss a domain decomposition algorithm for the problem (2.4) and its convergence rate and then perform some numerical tests to verify the theoretical results and illustrate the efficiency of the algorithm.

3. SA–GP algorithm

To solve the optimality system by DDM, we adopt a decoupled method: the outer iterations are used to solve the inequality with respect to the control; the inner iterations are used to treat PDEs with respect to the state and co-state variables. There are many choices of iterative solvers in outer loops and inner loops. Here we adopt the gradient projection algorithm as outer iterations and Schwarz alternating procedures as inner iterations. It is well known that the coloring technique could be used to classify sub-domains into different groups with different colors such that in the same group sub-domains are disjoint each other. So subproblems defined in sub-domains of same color could be solved in parallel. In Section 3.1, we give some preliminary results which will be used in the algorithms. Further, we will state SA–GP algorithms in continuous and discrete cases in Sections 3.2 and 3.3.

3.1. Preliminary results

First of all, let us recall Schwarz alternating algorithms for solving the following variational problem: seek $y \in V$ such that

$$a(y, v) = (g, v), \quad \forall v \in V. \tag{3.1}$$

Construct an initial partition of Ω with non-overlapping sub-domains $\{\widetilde{\Omega}_j\}_{j=1}^J$, then extend each sub-domain $\widetilde{\Omega}_j$ to a larger region $\Omega_j \supset \widetilde{\Omega}_j$, where $\{\Omega_j\}_{j=1}^J$ are overlapping sub-domains of Ω such that $\Omega = \bigcup_{1 \leq j \leq J} \Omega_j$. The overlapping degree δ of the overlapping domain decomposition is defined as $\delta = \min_{1 \leq j \leq J} \{\delta_j\}$, where $\delta_j \triangleq \text{dist}(\partial\Omega_j \setminus \partial\Omega, \partial\widetilde{\Omega}_j \setminus \partial\Omega)$. Assume $\delta > 0$. Define $V_j = \{v \in H_0^1(\Omega);\ v = 0 \text{ in } \Omega \setminus \Omega_j\}$ for $1 \leq j \leq J$. The classical Schwarz alternating algorithm reads:

- Step 1. *Give a tolerance $\epsilon > 0$ and an initial approximation $y_0 \in V$. Set $k = 0$.*
- Step 2. *For $j = 1, \ldots, J$, successively solve $\tilde{y}_{k+\frac{j}{J}}$ such that*

$$a\left(\tilde{y}_{k+\frac{j}{J}}, w\right) = (g, w), \quad \forall w \in V_j, \qquad \tilde{y}_{k+\frac{j}{J}}\Big|_{\partial\Omega_j} = y_{k+\frac{j-1}{J}}, \tag{3.2}$$

and define

$$y_{k+\frac{j}{J}} \triangleq \begin{cases} \tilde{y}_{k+\frac{j}{J}} & \text{in } \Omega_j, \\ y_{k+\frac{j-1}{J}} & \text{in } \Omega \setminus \Omega_j. \end{cases} \tag{3.3}$$

- Step 3. *If $|y_{k+1} - y_k|_1 > \epsilon$, set $k := k + 1$ and then return to Step 2 to start new iteration; else output y_{k+1} and stop the iteration.*

It follows from Lions' theory [27] that there exists a positive constant $0 \le \gamma < 1$ such that

$$|y - y_k|_1 \le \gamma^k |y - y_0|_1. \tag{3.4}$$

(3.4) shows that the solution y_k of the Schwarz alternating algorithm is convergent geometrically as k tends to infinity. In practical computing, the iterations are control by a given tolerance ϵ.

Secondly, we need a Poincaré constant such that

$$\|v\|_0 \le C_\Omega |v|_1, \quad \forall v \in H_0^1(\Omega). \tag{3.5}$$

The parameters γ and C_Ω will play important roles in the our algorithm. It will be discuss how to estimate γ and C_Ω in Section 4.2.

3.2. SA–GP algorithm

In our algorithm, we need to determine the iterative parameter ρ in the outer loops and the iterative number in the inner loops. To this end, define some constants as

$$\begin{aligned}
&\underline{\rho} = \frac{2}{2\alpha + C_\Omega^4}, \qquad \bar{\rho} = \frac{2}{\alpha + C_\Omega^4},\\
&\alpha_1 = \begin{cases} |1 - \alpha\rho|, & 0 < \rho < \underline{\rho},\\ |1 - (\alpha + C_\Omega^4)\rho|, & \underline{\rho} \le \rho < \bar{\rho}, \end{cases}\\
&\alpha_2^2 = \left(\rho C_\Omega^4 + 1 - \alpha\rho\right)\rho,\\
&\alpha_3^2 = \max\{\alpha_2(\alpha_1 + \alpha_2), \alpha_2 C_\Omega^2(\rho + \alpha_2)\},\\
&\alpha_4^2 = C_\Omega^2(3\gamma\rho^2 + \alpha_2(\alpha_1 + 2\alpha_2 + \rho)),
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
&\alpha_5 = \max\left\{\frac{1}{2} + C_\Omega^4(\rho + \alpha_2), \frac{3}{2}C_\Omega^2\right\},\\
&\alpha_6 = C_\Omega^3(1 + \alpha_1 + \alpha_2),\\
&\alpha_7 = \max\{1, (\rho + \alpha_2)C_\Omega^2\}
\end{aligned} \tag{3.7}$$

and

$$\begin{aligned}
&a_1 = \alpha_5 + \frac{1}{2}\alpha_7, \qquad b_1 = \alpha_3 + \alpha_4\alpha_5,\\
&a_2 = \frac{1}{2}C_\Omega(1 + \alpha_1 + \alpha_2) + \alpha_6, \qquad b_2 = \alpha_3 + \alpha_4\alpha_6.
\end{aligned} \tag{3.8}$$

Furthermore, let r_1 be the positive root of the quadratic equation $a_1x^2 + b_1x - \alpha_1 = 0$ and r_2 be the positive root of the quadratic equation $a_2x^2 + b_2x - (1 - \alpha_1) = 0$. Define

$$M(\alpha, \gamma, C_\Omega, \rho) = \log_\gamma\left(\sqrt{\min\{r_1^2, r_2^2\} + 1} - 1\right). \tag{3.9}$$

Now we can state the following Schwarz-type gradient projection algorithm.

SA–GP algorithm

- Step 1. *Give an initial approximation* (u_0, y_0, p_0) *and a tolerance* $\epsilon > 0$. *Set* $k = 0$.
- Step 2. *Take* $0 < \rho_k < \bar{\rho}$ *as the iterative parameter in the k-th outer loop and the inner iteration number* $M_k \ge M(\alpha, \gamma, C_\Omega, \rho_k)$ *and* $N_k \ge M(\alpha, \gamma, C_\Omega, \rho_k)$ *for the state and co-state variables respectively. Set* $m = 0$ *and* $y_k^0 \triangleq y_k$, *then compute* y_{k+1} *as follow.*

  (a) For $j = 1, \ldots, J$, *successively solve* $\tilde{y}_k^{m+\frac{j}{J}}$ *such that*

  $$a\left(\tilde{y}_k^{m+\frac{j}{J}}, w\right) = (u_k + f, w), \quad \forall w \in V_j, \qquad \tilde{y}_k^{m+\frac{j}{J}}\Big|_{\partial\Omega_j} = y_k^{m+\frac{j-1}{J}} \tag{3.10}$$

  and define

  $$y_k^{m+\frac{j}{J}} \triangleq \begin{cases} \tilde{y}_k^{m+\frac{j}{J}} & \text{in } \Omega_j,\\ y_k^{m+\frac{j-1}{J}} & \text{in } \Omega \setminus \Omega_j. \end{cases} \tag{3.11}$$

  (b) *Set* $m := m + 1$. *If* $m < M_k$ *and* $|y_k^{m+1} - y_k^m|_1 > \epsilon$, *then return to* (a); *else define*

  $$y_{k+1} \triangleq y_k^{m+1} \quad \text{in } \Omega, \tag{3.12}$$

  then goto Step 3.

- Step 3. *Set* $m = 0$ *and* $p_k^0 \triangleq p_k$ *and then compute* p_{k+1} *as follow.*

  (c) *For* $j = 1, \ldots, J$, *successively solve* $\tilde{p}_k^{m+\frac{j}{J}}$ *such that*

$$a\left(q, \tilde{p}_k^{m+\frac{j}{J}}\right) = (y_{k+1} - y_d, q), \quad \forall q \in V_j, \qquad \tilde{p}_k^{m+\frac{j}{J}}\Big|_{\partial\Omega_j} = p_k^{m+\frac{j-1}{J}}, \tag{3.13}$$

  then define

$$p_k^{m+\frac{j}{J}} \triangleq \begin{cases} \tilde{p}_k^{m+\frac{j}{J}} & \text{in } \Omega_j, \\ p_k^{m+\frac{j-1}{J}} & \text{in } \Omega \setminus \Omega_j. \end{cases} \tag{3.14}$$

  (d) *Set* $m := m + 1$. *If* $m < N_k$ *and* $|p_k^{m+1} - p_k^m|_1 > \epsilon$, *then return to* (c); *else define*

$$p_{k+1} \triangleq p_k^{m+1} \quad \text{in } \Omega. \tag{3.15}$$

- Step 4. *Update* u_{k+1} *from* u_k *as follow*: *define*

$$\tilde{u}_{k+1} = u_k - \rho_k(\alpha u_k + p_{k+1}), \tag{3.16}$$

  then project $\tilde{u}_{k+1}$ *into* $\mathcal{U}_{\text{ad}}$:

$$u_{k+1} = \mathcal{Q}\tilde{u}_{k+1}. \tag{3.17}$$

- Step 5. *If* $\|u_{k+1} - u_k\|_0 > \epsilon$, *set* $k := k + 1$ *and then return to Step* 2 *to start new iteration; else define and output*

$$u = u_{k+1}, \qquad y = y_{k+1}, \qquad p = p_{k+1}, \tag{3.18}$$

  then stop the iteration.

Similarly, we can define the discrete SA–GP algorithm in the next subsection.

3.3. Discrete schemes

We introduce a finite element discrete scheme of the optimal control problems. (Ref. [31–34].) There are two approaches to construct a finite element partition of the domain Ω. The first is to make a partition in a given overlapping domain decomposition, i.e., first make an overlapping domain decomposition $\Omega = \bigcup_{j=1}^{J} \Omega_j$, which satisfies the conditions in Section 3.2, and then construct a triangulation $\mathcal{T}^h$, in which $\partial\Omega_i$ does not cut through any element in $\mathcal{T}^h$. The second is to form an overlapping domain decomposition in a given finite element partition. Let $\mathcal{T}^h$ be a quasi-regular triangulation of Ω. Then decompose $\mathcal{T}_h$ into an initial non-overlapping decomposition $\mathcal{T}^h = \bigcup_{j=1}^{J} \tilde{\mathcal{T}}_j^h$. Then extend each sub-triangulation $\tilde{\mathcal{T}}_j^h$ to a larger triangulation $\mathcal{T}_j^h$ such that $\{\Omega_j = \mathcal{T}_j^h\}_{j=1}^{J}$ form a domain decomposition of Ω satisfying the conditions given in Section 3.2. This could be done by repeatedly adding a layer of elements starting from $\tilde{\mathcal{T}}_j^h$ for $1 \le j \le J$.

Let $\mathcal{T}^h$ be a quasi-regular triangulation of Ω with the maximum mesh size $h \triangleq \max_{\tau \in \mathcal{T}^h}\{\text{diam}(\tau)\}$ and $\mathcal{T}^h = \{\mathcal{T}_j^h\}_{j=1}^{J}$ be an overlapping decomposition. Associated with $\mathcal{T}_h$ is a finite dimensional subspace $V^h = \{w_h \in H_0^1(\Omega) : w_h|_\tau$ is a polynomial of degree lesser than and equal to r $(r \ge 1)$ for each $\tau \in \mathcal{T}^h\} \subset V$.

Let $\mathcal{T}_U^h$ be another quasi-regular triangulation of Ω with the maximum mesh size $h_U \triangleq \max_{\tau_U \in \mathcal{T}_U^h}\{\text{diam}(\tau_U)\}$. Associated with $\mathcal{T}_U^h$ is a finite dimensional subspace $U^h := \{v_h \in L^2(\Omega) : v_h|_{\tau_U}$ is a polynomial of degree lesser than and equal to k $(k \ge 0)$ for each $\tau_U \in \mathcal{T}_U^h\} \subset U^h$.

The discrete optimal control problem reads:

$$\begin{cases} \min\limits_{v_h \in \mathcal{U}_{\text{ad}} \cap U^h} & \mathcal{J}(v_h, z_h) \\ \text{s.t.} & a(z_h, w_h) = (v_h + f, w_h), \quad \forall w_h \in V^h. \end{cases} \tag{3.19}$$

The equivalent optimality conditions is as follows:

$$(\text{OCP})^h \begin{cases} a(y_h, w_h) = (u_h + f, w_h), & \forall w_h \in V^h, \\ a(p_h, q_h) = (y_h - y_d, q_h), & \forall q_h \in V^h, \\ \displaystyle\int_\Omega (u_h + p_h)(v_h - u_h) \ge 0, & \forall\, v_h \in \mathcal{U}_{\text{ad}} \cap U^h. \end{cases} \tag{3.20}$$

Let $V_j^h = \{w_h \in V^h;\ v_h = 0 \text{ in } \tau \notin \mathcal{T}_j^h\}$ for $1 \le j \le J$. Define $\mathcal{Q}^h : V^h \to \mathcal{U}_{\text{ad}} \bigcap U^h$ such that for each $q_h \in V_h$

$$\int_\Omega (\mathcal{Q}^h q_h - q_h)(v_h - \mathcal{Q}^h q_h) \ge 0, \quad \forall v_h \in \mathcal{U}_{\text{ad}} \bigcap U^h.$$

We get $\alpha u_h = -\mathcal{Q}^h p_h$. The discrete Schwarz-type gradient projection algorithm reads:

Discrete SA–GP algorithm

- Step 1. *Give an initial approximation* $(u_{h,0}, y_{h,0}, p_{h,0})$ *and a tolerance* $\epsilon > 0$. *Set* $k = 0$.
- Step 2. *Take* $0 < \rho_k < \bar{\rho}$ *as iterative parameter in the k-th outer loop and the inner iteration number* $M_k \geq M(\alpha, \gamma, C_\Omega, \rho_k)$ *and* $N_k \geq M(\alpha, \gamma, C_\Omega, \rho_k)$ *for the state and co-state variables respectively. Set* $m = 0$ *and* $y^0_{h,k} \triangleq y_{h,k}$ *and then compute* $y_{h,k+1}$ *as follows.*

  (a) *For* $j = 1, \ldots, J$, *successively solve* $\tilde{y}^{m+\frac{j}{J}}_{h,k}$ *such that*
$$a\left(\tilde{y}^{m+\frac{j}{J}}_{h,k}, w\right) = (u_{h,k} + f, w_h), \quad \forall w_h \in V^h_j, \qquad \tilde{y}^{m+\frac{j}{J}}_{h,k} - y^{m+\frac{j-1}{J}}_{h,k} \in V^h_j, \tag{3.21}$$
  and define
$$y^{m+\frac{j}{J}}_{h,k} \triangleq \begin{cases} \tilde{y}^{m+\frac{j}{J}}_{h,k} & \text{in } \Omega_j, \\ y^{m+\frac{j-1}{J}}_{h,k} & \text{in } \Omega \setminus \Omega_j. \end{cases} \tag{3.22}$$
  (b) *Set* $m := m + 1$. *If* $m < M_k$ *and* $|y^{m+1}_{h,k} - y^m_{h,k}|_1 > \epsilon$, *then return to* (a); *else define*
$$y_{h,k+1} \triangleq y^{m+1}_{h,k} \quad \text{in } \Omega, \tag{3.23}$$
  then goto Step 3.
- Step 3. *Set* $m = 0$ *and* $p^0_{h,k} \triangleq p_{h,k}$ *and then compute* $p_{h,k+1}$ *as follow.*

  (c) *For* $j = 1, \ldots, J$, *successively solve* $\tilde{p}^{m+\frac{j}{J}}_{h,k}$ *such that*
$$a\left(q_h, \tilde{p}^{m+\frac{j}{J}}_{h,k}\right) = (y_{h,k+1} - y_d, q_h), \quad \forall q_h \in V^h_j, \qquad \tilde{p}^{m+\frac{j}{J}}_{h,k} - p^{m+\frac{j-1}{J}}_{h,k} \in V^h_j, \tag{3.24}$$
  and then define
$$p^{m+\frac{j}{J}}_{h,k} \triangleq \begin{cases} \tilde{p}^{m+\frac{j}{J}}_{h,k} & \text{in } \Omega_j, \\ p^{m+\frac{j-1}{J}}_{h,k} & \text{in } \Omega \setminus \Omega_j. \end{cases} \tag{3.25}$$
  (d) *Set* $m := m + 1$. *If* $m < N_k$ *and* $|p^{m+1}_{h,k} - p^m_{h,k}|_1 > \epsilon$, *then return to* (c); *else define*
$$p_{h,k+1} \triangleq p^{m+1}_{h,k} \quad \text{in } \Omega. \tag{3.26}$$
- Step 4. *Update* $u_{h,k+1}$ *from* $u_{h,k}$ *as follow*: *define*
$$\tilde{u}_{h,k+1} = u_{h,k} - \rho_k(\alpha u_{h,k} + p_{h,k+1}) \tag{3.27}$$
  and then project $\tilde{u}_{h,k+1}$ *into* $\mathcal{U}_{\text{ad}} \bigcap U^h$,
$$u_{h,k+1} = \mathcal{Q}^h \tilde{u}_{h,k+1}. \tag{3.28}$$
- Step 5. *If* $\|u_{h,k+1} - u_{h,k}\|_0 > \epsilon$, *set* $k := k + 1$ *and then return to Step* 2 *to start new iteration*; *else define and output*
$$u_h = u_{h_k+1}, \qquad y_h = y_{h,k+1}, \qquad p_h = p_{h,k+1}, \tag{3.29}$$
  and then stop the iteration.

Remark 3.1. In SA–GP algorithms, Steps 2 and 3 are the inner loops for the state and co-state variables, while Step 4 is the outer loops for the control variable. It is well known that if the iterative pairs (y^m_k, p^m_k) and $(y^m_{h,k}, p^m_{h,k})$ sufficiently approximate their limitations with respect to m, the convergence of iterative solutions is determined by the parameter ρ_k of the outer loops and the iterative solutions are geometrically convergent. By repeated iterating, the error resulting from the inner loops gradually vanishes but the computational cost becomes higher so that the total computation cost also becomes higher. However, one wants to spend lower computational cost at each outer loop and obtains pretty fast convergence rate at the same time. Total computational cost comes from two stages, the outer loops and the inner loops. It is easily observed that the most computation work is done in Schwarz alternating schemes in the inner loops. Thus reasonable decrease of the iteration number in the inner loops will save lots of total computational cost. In the classical Schwarz alternating scheme, one stops the iteration procedures when the errors is within the given tolerance. Many iterative steps are needed to reach the tolerance, specially in the case that initial approximation is far away from the exact solution. These facts will be shown in the numerical tests in the last section. In our algorithm, the different point is that the stop criterion in the inner loops is some given iteration number independent of the tolerance, which is not too large. That means it is unnecessary to make the iterative solutions be sufficiently closed to the exact solutions in the inner loops. As a result, the computational cost is reduced greatly, while the algorithm is still geometrically convergent that is guaranteed by Theorem 4.1.

Remark 3.2. In SA–GP algorithms, the inner iteration numbers M_k and N_k play important roles, which depend on the parameters α, γ, C_Ω and ρ_k. Our analyses show that one should take $M_k = N_k$. We will discuss how to choose a better iteration parameter ρ_k and to estimate γ and C_Ω in Section 4.2 so that one can get M_k and N_k by using (3.9).

4. Analysis of convergence

In this section, we will analyze the convergence rate of the SA–GP algorithm. For the sake of simplicity and without losing generality, we assume $\rho_k = \rho$ for $k = 1, 2, \ldots$. It is useful to determine minimum iteration number M and N in the inner loops. As the SA–GP algorithms for the continuous problem and the discrete problem have the same weak form, we will just analyze the convergence rate of SA–GP algorithm for the continuous problem. In Section 4.1, we state the convergence theorem, which is the main result in this section. Then, in Section 4.2, we discuss how to determine some important constants, which are necessary to practical applications. Finally, in Section 4.3, we prove Theorem 4.1 given in Section 4.1.

4.1. Convergence theorem

In order to prove convergence of the algorithm, introduce the projection operators $\mathcal{P}_j : V \to V_j$ such that

$$a(\mathcal{P}_j v, w) = a(v, w), \quad \forall v \in V,\ w \in V_j$$

for $j = 1, \ldots, J$. Furthermore, for any $v \in L^2(\Omega)$, we introduce the auxiliary problem: $y(v) \in V$ and $p(v) \in V$ such that

$$\begin{aligned} a(y(v), w) &= (v + f, w), \quad \forall w \in V, \\ a(q, p(v)) &= (y(v) - y_d, q), \quad \forall q \in V. \end{aligned} \tag{4.1}$$

Correspondingly, introduce an inverse elliptic operator $\mathcal{S} : U \to V$ such that

$$a(\mathcal{S}f, v) = (f, v), \quad \forall v \in V \text{ and } f \in U.$$

Define operator

$$\mathcal{E}_J \triangleq (\mathcal{I} - \mathcal{P}_J)(\mathcal{I} - \mathcal{P}_{J-1}) \cdots (\mathcal{I} - \mathcal{P}_1),$$

and its compressibility factor as

$$\gamma = \|\mathcal{E}_J\| \triangleq \sup_{w \in H_0^1(\Omega),\ |w|_1 = 1} |\mathcal{E}_J w|_1. \tag{4.2}$$

It follows from Lions' theory [27] that if $V = V_1 + V_2 + \cdots + V_J$, then

$$0 \le \gamma < 1. \tag{4.3}$$

It is clear that (y_k, p_k) converges to $(y(u_k), p(u_k))$ as k tends to infinity. However, it needs too many iterations to reach the end in each inner loop. One desires to use much less inner iterations. We will analyze convergence property of SA–GP algorithm. By using Poincaré inequality, it is easily obtained that

$$|\mathcal{S}f|_1 \le C_\Omega \|f\|_0. \tag{4.4}$$

The following theorem is the main result of this article, which characterizes the iteration property of the approximation solutions.

Theorem 4.1. *Let $(u^*, y^*, p^*) \in \mathcal{U}_{\text{ad}} \times V \times V$ be the solution of the model problem and $(u_k, y_k, p_k) \in \mathcal{U}_{\text{ad}} \times V \times V$ be given by the algorithm SA–GP for $0 \le k < \infty$. Let $\delta_{\gamma,M,N} = \gamma^M + \gamma^{M+N} + \gamma^N$. Then $0 < \alpha_1 < 1$ for $0 < \rho < \bar{\rho}$. There exist some constants c_1, c_2, λ_1 and λ_2 satisfying $|\lambda_1| < 1$ and $|\lambda_2| < 1$ such that*

$$\|u_k - u^*\|_0 + |y_k - y^*|_1 + |p_k - p^*|_1 \le c_1 \lambda_1^k + c_2 \lambda_2^k, \tag{4.5}$$

provided

$$\beta_1(M, N)\delta_{\gamma,M,N}^{\frac{1}{2}} \le \alpha_1, \qquad \alpha_1^*(M, N) < 1 - \alpha_1, \tag{4.6}$$

where

$$\begin{aligned} \beta_1(M, N) &= \alpha_3 + \alpha_4\alpha_5 + \left(\alpha_5 + \frac{1}{2}\alpha_7\right)\delta_{\gamma,M,N}^{\frac{1}{2}}, \\ \alpha_1^*(M, N) &= \beta_2(M, N)\,\delta_{\gamma,M,N}^{\frac{1}{2}}, \\ \beta_2(M, N) &= \alpha_3 + \alpha_4\alpha_6 + \left(\frac{1}{2}C_\Omega(1 + \alpha_1 + \alpha_2) + \alpha_6\right)\delta_{\gamma,M,N}^{\frac{1}{2}}, \end{aligned} \tag{4.7}$$

and $0 < \lambda_1 < 1, -1 < \lambda_2 < 0$ are the roots of equation

$$\lambda^2 - \alpha_1\lambda - \alpha_1^*(M, N) = 0. \tag{4.8}$$

4.2. Choice of some important parameters

Some parameters in Theorem 4.1 are important in both practical applications and theoretical analysis. In this subsection, we will discuss how to determine these parameters.

4.2.1. Descent step ρ

In the analysis, other parameters depend on the descent step of projection gradient methods ρ. How can we choose ρ to get much better convergence rates? One knows that the SA–GP algorithm is an approximation of gradient projection methods without domain decomposition. Thus if we choose suitable descent step of gradient projection methods without DDM, then the step is also the better choice for SA–GP algorithm. Similar to the proof of Lemma 4.5, one has got the convergence of gradient projection methods. The convergence rate is just α_1. The proposed step ρ^* is $\rho^* = \min_{0<\rho\le\overline{\rho}} \alpha_1$ (see (3.6)). Thus we choose

$$\rho^* = \underline{\rho} = \frac{2}{2\alpha + C_\Omega^4}. \tag{4.9}$$

4.2.2. Iteration number M

To save computational cost, the most important is how to determine optimal iteration number at each inner loop. However, it is very difficult to obtain the number. Thus we will give an estimated number with respect to M. In addition, define

$$a_1 = \alpha_5 + \frac{1}{2}\alpha_7, \qquad b_1 = \alpha_3 + \alpha_4\alpha_5,$$
$$a_2 = \frac{1}{2}C_\Omega(1 + \alpha_1 + \alpha_2) + \alpha_6, \qquad b_2 = \alpha_3 + \alpha_4\alpha_6.$$

r_1 is the positive root of the quadratic equation $a_1x^2 + b_1x - \alpha_1 = 0$ while r_2 is the positive root of the quadratic equation $a_2x^2 + b_2x - (1 - \alpha_1) = 0$.

If

$$M \ge \log_\gamma\left(\sqrt{\min\{r_1^2, r_2^2\} + 1} - 1\right), \tag{4.10}$$

then the condition (4.6) holds so that the algorithm SA–GP algorithm is geometrically convergent.

4.2.3. Convergence factor of Schwarz algorithm

We provide two approaches. One is the theoretical estimation, the other is the numerical estimation.

The first approach is the theoretical estimation. For example, see [27,34]. It is based on the well known result as follows. Let $V = V_1 + V_2 + \cdots + V_J$. If there exist a linear decomposition form $v = v_1 + v_2 + \cdots + v_J$ for each $v \in V$, where $v_i \in V_i$ for $1 \le i \le J$, and a constant $C_0 > 0$ such that

$$|v_1|_1^2 + |v_2|_1^2 + \cdots + |v_J|_1^2 \le C_0^2|v|_1^2, \quad \forall v \in V,$$

then there holds

$$\gamma = \sqrt{1 - \frac{1}{C_0^2}} \tag{4.11}$$

where C_0 relies on the overlapping degree δ of domain decomposition and the number of sub-domains.

Another approach in practical computations is to use numerical results to estimate γ. For example, let y_0^1, y_0^2 and y_0^3 be given by successively Schwarz iterations. Then $\tilde{\gamma} = \frac{\varepsilon_3}{\varepsilon_2}$ is the applicable estimation of γ, where $\varepsilon_k = |y_{h,0}^k - y_{h,0}^{k-1}|_1$.

4.2.4. Poincaré constant

In the convergence analysis, Poincaré constant C_Ω is the important parameter. For an unit square of two dimension, the best choice is $C_\Omega = \frac{1}{\sqrt{2}\pi}$ given in [35]. For rectangular region Ω, $C_\Omega = \frac{|\Omega|}{\pi\sqrt{\mathrm{diam}(\Omega)}}$, where $\mathrm{diam}(\Omega)$ is the diameter of Ω and $|\Omega|$ is the volume of Ω. If Ω is a bounded domain of d-dimension, then $C_\Omega = \sqrt{\frac{|\Omega|}{\omega_d}}$ in [36] where ω_d is the volume of unit ball in $\mathbb{R}^d$.

4.3. Proof of convergence theorem

The proof of Theorem 4.1 is completed by the following lemmas.

Lemma 4.1. *Let y_k^m be given by the algorithm SA–GP for $1 \le m \le M$ and $0 \le k < \infty$. There holds the following relationship:*

$$y_k^{m+1} - y(u_k) = \mathcal{E}_J\left(y_k^m - y(u_k)\right), \quad 0 \le m \le M - 1 \tag{4.12}$$

for $k = 1, 2, \ldots$.

Proof. Noting that

$$a(y(u_k), w) = (u_k + f, w), \quad \forall w \in V, \tag{4.13}$$

we know that Step 2 is a standard Schwarz alternating algorithm for the Eq. (4.13) so that (4.12) holds. For details, see [27,3]. □

Lemma 4.2. *Let p_k^m be given by the algorithm SA–GP for $1 \le m \le M$ and $0 \le k < \infty$. There holds the following relationship*

$$p_k^{m+1} - p(u_k) - \mathcal{S}\big(y_{k+1} - y(u_k)\big) = \mathcal{E}_J\Big(p_k^m - p(u_k) - \mathcal{S}\big(y_{k+1} - y(u_k)\big)\Big), \quad 0 \le m \le M-1 \tag{4.14}$$

for $k = 1, 2, \ldots$.

Proof. From the algorithm

$$a\left(\tilde{p}_k^{m+\frac{j}{J}}, q\right) = (y_{k+1} - y_d, q), \quad \forall q \in V_j,\ 1 \le j \le J,$$

and the auxiliary problem

$$a(p(u_k), q) = (y(u_k) - y_d, q), \quad \forall q \in V,$$

we know

$$a\left(\tilde{p}_k^{m+\frac{j}{J}} - p(u_k), q\right) = (y_{k+1} - y(u_k), q), \quad \forall q \in V_j,\ 1 \le j \le J. \tag{4.15}$$

Since $p_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}} = \tilde{p}_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}}$ in Ω_j such that

$$\begin{aligned}
a\left(p_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}}, q\right) &= a\left(\tilde{p}_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}}, q\right) \\
&= a\left(\tilde{p}_k^{m+\frac{j}{J}} - p(u_k), q\right) - a\left(p_k^{m+\frac{j-1}{J}} - p(u_k), q\right) \\
&= (y_{k+1} - y(u_k), q) - a\left(\mathcal{P}_j\left(p_k^{m+\frac{j-1}{J}} - p(u_k)\right), q\right) \\
&= a(\mathcal{S}(y_{k+1} - y(u_k)), q) - a\left(\mathcal{P}_j\left(p_k^{m+\frac{j-1}{J}} - p(u_k)\right), q\right), \quad \forall q \in V_j,
\end{aligned}$$

and $p_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}} = 0$ on $\partial\Omega_j$. Therefore,

$$p_k^{m+\frac{j}{J}} - p_k^{m+\frac{j-1}{J}} = \mathcal{P}_j\left(\mathcal{S}(y_{k+1} - y(u_k)) - \left(p_k^{m+\frac{j-1}{J}} - p(u_k)\right)\right),$$

such that

$$p_k^{m+\frac{j}{J}} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k)) = (\mathcal{I} - \mathcal{P}_j)\left(p_k^{m+\frac{j-1}{J}} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))\right), \quad 1 \le j \le J. \tag{4.16}$$

By recursion, we have

$$\begin{aligned}
p_k^{m+1} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k)) &= (\mathcal{I} - \mathcal{P}_J)\left(p_k^{m+\frac{J-1}{J}} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))\right) \\
&= (\mathcal{I} - \mathcal{P}_J)(\mathcal{I} - \mathcal{P}_{J-1})\left(p_k^{m+\frac{J-2}{J}} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))\right) \\
&= \cdots \\
&= (\mathcal{I} - \mathcal{P}_J)(\mathcal{I} - \mathcal{P}_{J-1}) \cdots (\mathcal{I} - \mathcal{P}_1)\big(p_k^n - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))\big).
\end{aligned}$$

This is (4.2). The proof of Lemma 4.2 is complete. □

As a consequence of Lemmas 4.1 and 4.2, we have the following conclusions.

Lemma 4.3. *Let $\{(y_k, p_k)\}_{k=1}^{\infty}$ be given by the algorithm SA–GP. There holds the following relationship:*

$$\begin{aligned}&\text{(a) } y_{k+1} - y(u_k) = \mathcal{E}_J^M(y_k - y(u_k)),\\ &\text{(b) } p_{k+1} - p(u_k) - \mathcal{S}\big(y_{k+1} - y(u_k)\big) = \mathcal{E}_J^N\Big(p_k - p(u_k) - \mathcal{S}\big(y_{k+1} - y(u_k)\big)\Big).\end{aligned} \tag{4.17}$$

Lemma 4.4. *Let $\delta_{\gamma,M,N} = \gamma^M + \gamma^{M+N} + \gamma^N$. There holds the inequality:*

$$|p_{k+1} - p(u_k)|_1 \le \gamma^N |p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + \delta_{\gamma,M,N} C_\Omega^2 \Big\{|y_k - y(u_{k-1})|_1 + C_\Omega \|u_k - u_{k-1}\|_0\Big\}. \tag{4.18}$$

Proof. It follows from Lemma 4.3,

$$\begin{aligned}|p_{k+1} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))|_1 &\le \gamma^N |p_k - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))|_1\\ &= \gamma^N |p_k - p(u_{k-1}) - \mathcal{S}(y_{k+1} - y(u_{k-1}))|_1\\ &\le \gamma^N \Big\{|p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + |\mathcal{S}(y_{k+1} - y_k)|_1\Big\}\\ &\le \gamma^N \Big\{|p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + |\mathcal{S}(y_{k+1} - y(u_k))|_1\\ &\quad + |\mathcal{S}(y_k - y(u_{k-1}))|_1 + |\mathcal{S}(y(u_k) - y(u_{k-1}))|_1\Big\}\\ &\le \gamma^N \Big\{|p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + C_\Omega^2 |y_k - y(u_{k-1})|_1\\ &\quad + C_\Omega^2\big(\gamma^M |y_k - y(u_{k-1})|_1 + (1+\gamma^M)|y(u_k) - y(u_{k-1})|_1\big)\Big\},\end{aligned}$$

such that

$$\begin{aligned}|p_{k+1} - p(u_k)|_1 &\le |p_{k+1} - p(u_k) - \mathcal{S}(y_{k+1} - y(u_k))|_1 + |\mathcal{S}(y_{k+1} - y(u_k))|_1\\ &\le \gamma^N |p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + \big(\gamma^M + \gamma^N + \gamma^{M+N}\big)\\ &\quad \times \Big\{|\mathcal{S}(y_k - y(u_{k-1}))|_1 + |\mathcal{S}(y(u_k) - y(u_{k-1}))|_1\Big\}\\ &\le \gamma^N |p_k - p(u_{k-1}) - \mathcal{S}(y_k - y(u_{k-1}))|_1 + \big(\gamma^M + \gamma^N + \gamma^{M+N}\big) C_\Omega^2\\ &\quad \times \Big\{|y_k - y(u_{k-1})|_1 + C_\Omega \|u_k - u_{k-1}\|_0\Big\}.\end{aligned}$$

The proof of Lemma 4.4 is completed. □

In the following lemmas we consider $M = N$. Thus the following inequalities holds

$$\gamma^M < \frac{1}{2}\delta_{\gamma,M,N}, \qquad \gamma^N < \frac{1}{2}\delta_{\gamma,M,N}.$$

Lemma 4.5. *Let $(u^*, y^*, p^*) \in \mathcal{U}_{\text{ad}} \times V \times V$ be the solution of the model problem. There holds the estimation:*

$$\begin{aligned}\|u_{k+1} - u^*\|_0 + |p_{k+1} - p(u_k)|_1 + |y_{k+1} - y(u_k)|_1 &\le \alpha_1\Big(\|u_k - u^*\|_0 + |y_k - y(u_{k-1})|_1 + |p_k - p(u_{k-1})|_1\Big)\\ &\quad + \alpha_1^*(M,N)\|u_{k-1} - u^*\|_0\end{aligned} \tag{4.19}$$

provided

$$\Big(\alpha_3 + \alpha_4\alpha_5 + (\alpha_5 + \alpha_7)\delta_{\gamma,M,N}^{\frac{1}{2}}\Big)\delta_{\gamma,M,N}^{\frac{1}{2}} \le \alpha_1,$$

where

$$\alpha_1^*(M,N) = \Big(\alpha_3 + \alpha_4\alpha_6 + \Big(\alpha_6 + \frac{1}{2}C_\Omega(1 + \alpha_1 + \alpha_2)\Big)\delta_{\gamma,M,N}^{\frac{1}{2}}\Big)\delta_{\gamma,M,N}^{\frac{1}{2}}.$$

Proof. Noting that $\alpha u^* = \mathcal{Q}(\alpha u^*) = -\mathcal{Q}p^*$, we have

$$\begin{aligned}
\|u_{k+1} - u^*\|_0^2 &= \|\mathcal{Q}\tilde{u}_{k+1} - \mathcal{Q}(u^* - \rho(\alpha u^* + p^*))\|_0^2 \\
&\le \|u_k - \rho(\alpha u_k + p_{k+1}) - (u^* - \rho(\alpha u^* + p^*))\|_0^2 \\
&= \|(1-\alpha\rho)(u_k - u^*) - \rho(p_{k+1} - p^*)\|_0^2 \\
&= (1-\alpha\rho)^2\|u_k - u^*\|_0^2 + \rho^2\|p_{k+1} - p^*\|_0^2 - 2(1-\alpha\rho)\rho(u_k - u^*, p_{k+1} - p^*) \\
&= (1-\alpha\rho)^2\|u_k - u^*\|_0^2 + \rho^2\|p(u_k) - p^*\|_0^2 + \rho^2\|p_{k+1} - p(u_k)\|_0^2 + 2\rho^2(p(u_k) - p^*, p_{k+1} - p(u_k)) \\
&\quad - 2(1-\alpha\rho)\rho(u_k - u^*, p(u_k) - p^*) - 2(1-\alpha\rho)\rho(u_k - u^*, p_{k+1} - p(u_k)).
\end{aligned} \tag{4.20}$$

Then we will estimate the terms on the right-hand side of (4.20) one by one. Noting

$$\|p(u_k) - p^*\|_0 = \|\mathcal{S}(y(u_k) - y^*)\|_0 \le C_\Omega|\mathcal{S}(y(u_k) - y^*)|_1 \le C_\Omega^2\|y(u_k) - y^*\|_0,$$

and

$$(u_k - u^*, p(u_k) - p^*) = (u_k - u^*, \mathcal{S}(y(u_k) - y^*)) = \|y(u_k) - y^*\|_0^2,$$

we have

$$\begin{aligned}
\|u_{k+1} - u^*\|_0^2 \le{}& (1-\alpha\rho)^2\|u_k - u^*\|_0^2 - 2\rho\left(1 - \rho\frac{2\alpha + C_\Omega^4}{2}\right)\|y(u_k) - y^*\|_0^2 \\
&+ 2(\rho^2 C_\Omega^4 + (1-\alpha\rho)\rho)C_\Omega\|u_k - u^*\|_0|p_{k+1} - p(u_k)|_1 + \rho^2\|p_{k+1} - p(u_k)\|_0^2.
\end{aligned} \tag{4.21}$$

In the case of $\underline{\rho} \le \rho < \bar{\rho}$, it is clear that

$$\left(1 - \left(\alpha + C_\Omega^4\right)\rho\right)^2 < 1.$$

Noting that

$$-2\rho\left(1 - \rho\frac{2\alpha + C_\Omega^4}{2}\right) = 2\rho(\rho - \underline{\rho})\underline{\rho}^{-1} = \rho\left(\rho(2\alpha + C_\Omega^4) - 2\right) = \rho^2 C_\Omega^4 - 2\rho(1-\alpha\rho) \ge 0,$$

and

$$(1-\alpha\rho)^2 + \rho^2 C_\Omega^8 - 2\rho(1-\alpha\rho)C_\Omega^4 = \left(1 - (\alpha + C_\Omega^4)\rho\right)^2,$$

and $\|y(u_k) - y^*\|_0 \le C_\Omega^2\|u_k - u^*\|_0$, we have

$$\begin{aligned}
\|u_{k+1} - u^*\|_0^2 \le{}& \left(1 - (\alpha + C_\Omega^4)\rho\right)^2\|u_k - u^*\|_0^2 + \rho^2\|p_{k+1} - p(u_k)\|_0^2 + 2(\rho^2 C_\Omega^4 \\
&+ (1-\alpha\rho)\rho)\|u_k - u^*\|_0\|p_{k+1} - p(u_k)\|_0,
\end{aligned}$$

such that

$$\|u_{k+1} - u^*\|_0^2 \le \alpha_1^2\|u_k - u^*\|_0^2 + \rho^2\|p_{k+1} - p(u_k)\|_0^2 + 2\alpha_2^2\|u_k - u^*\|_0\|p_{k+1} - p(u_k)\|_0. \tag{4.22}$$

Similarly, we get

$$\begin{aligned}
\|u_k - u^*\| &\le \alpha_1\|u_{k-1} - u^*\|_0 + \rho\|p_k - p(u_{k-1})\|_0 + \sqrt{2}\alpha_2\|u_{k-1} - u^*\|_0^{\frac{1}{2}}\|p_k - p(u_{k-1})\|_0^{\frac{1}{2}} \\
&\le (\alpha_1 + \alpha_2)\|u_{k-1} - u^*\|_0 + (\rho + \alpha_2)\|p_k - p(u_{k-1})\|_0.
\end{aligned} \tag{4.23}$$

Thus we obtain

$$\begin{aligned}
|y_{k+1} - y(u_k)|_1 &\le \gamma^M|y_k - y(u_k)|_1 \\
&\le \gamma^M(|y_k - y(u_{k-1})|_1 + |y(u_k) - y(u_{k-1})|_1) \\
&\le \gamma^M\left(|y_k - y(u_{k-1})|_1 + C_\Omega(\|u_k - u^*\|_0 + \|u_{k-1} - u^*\|_0)\right) \\
&\le \gamma^M|y_k - y(u_{k-1})|_1 + \gamma^M C_\Omega(1 + C_\Omega(\alpha_1 + \alpha_2))\|u_{k-1} - u^*\|_0 \\
&\quad + \gamma^M C_\Omega^2(\rho + \alpha_2)|p_k - p(u_{k-1})|_1 \\
&\le \alpha_7\left(|p_k - p(u_{k-1})|_1 + |y_k - y(u_{k-1})|_1\right) + \gamma^M C_\Omega(1 + \alpha_1 + \alpha_2)\|u_{k-1} - u^*\|_0,
\end{aligned} \tag{4.24}$$

where

$$\alpha_7 = \max\{1, (\rho + \alpha_2)C_\Omega^2\},$$

and

$$\begin{aligned}\|u_{k+1}-u^*\|_0^2 &\le \alpha_1^2\|u_k-u^*\|_0^2+\rho^2\|p_{k+1}-p(u_k)\|_0^2+2\alpha_2\big((\alpha_1+\alpha_2)\|u_{k-1}-u^*\|_0\\ &\quad+(\rho+\alpha_2)\|p_k-p(u_{k-1})\|_0\big)\|p_{k+1}-p(u_k)\|_0\\ &\le \alpha_1^2\|u_k-u^*\|_0^2+\delta_{\gamma,M,N}\alpha_3^2\Big(\|u_{k-1}-u^*\|_0^2+|p_k-p(u_{k-1})|_1^2\Big)+\alpha_4^2\delta_{\gamma,M,N}^{-1}|p_{k+1}-p(u_k)|_1^2,\end{aligned}\tag{4.25}$$

where $\alpha_3^2=\max\{\alpha_2(\alpha_1+\alpha_2),\alpha_2C_\Omega^2(\rho+\alpha_2)\}$ and $\alpha_4^2=C_\Omega^2(3\gamma\rho^2+\alpha_2(\alpha_1+2\alpha_2+\rho))$ by $\delta_{\gamma,M,N}\le 3\gamma$. From Lemma 4.4, we see that

$$\begin{aligned}|p_{k+1}-p(u_k)|_1 &\le \gamma^N|p_k-p(u_{k-1})|_1+\gamma^NC_\Omega^2|y_k-y(u_{k-1})|_1+\delta_{\gamma,M,N}C_\Omega^2\big(|y_k-y(u_{k-1})|_1+C_\Omega\|u_k-u_{k-1}\|_0\big)\\ &= \gamma^N|p_k-p(u_{k-1})|_1+C_\Omega^2(\gamma^N+\delta_{\gamma,M,N})|y_k-y(u_{k-1})|_1+\delta_{\gamma,M,N}C_\Omega^3\|u_k-u_{k-1}\|_0\\ &\le \gamma^N|p_k-p(u_{k-1})|_1+C_\Omega^2(\gamma^N+\delta_{\gamma,M,N})|y_k-y(u_{k-1})|_1\\ &\quad+\delta_{\gamma,M,N}C_\Omega^3\big(\|u_k-u^*\|_0+\|u_{k-1}-u^*\|_0\big)\\ &\le \gamma^N|p_k-p(u_{k-1})|_1+C_\Omega^2(\gamma^N+\delta_{\gamma,M,N})|y_k-y(u_{k-1})|_1+\delta_{\gamma,M,N}C_\Omega^3\big((\alpha_1+\alpha_2)\|u_{k-1}-u^*\|_0\\ &\quad+(\rho+\alpha_2)\|p_k-p(u_{k-1})\|_0+\|u_{k-1}-u^*\|_0\big)\\ &\le \big(\gamma^N+\delta_{\gamma,M,N}C_\Omega^4(\rho+\alpha_2)\big)|p_k-p(u_{k-1})|_1+C_\Omega^2(\gamma^N+\delta_{\gamma,M,N})|y_k-y(u_{k-1})|_1\\ &\quad+\delta_{\gamma,M,N}C_\Omega^3(1+\alpha_1+\alpha_2)\|u^*-u_{k-1}\|_0\\ &\le \delta_{\gamma,M,N}\left(\frac{1}{2}+C_\Omega^4(\rho+\alpha_2)\right)|p_k-p(u_{k-1})|_1+\frac{3}{2}C_\Omega^2|y_k-y(u_{k-1})|_1\\ &\quad+\delta_{\gamma,M,N}C_\Omega^3(1+\alpha_1+\alpha_2)\|u^*-u_{k-1}\|_0\\ &\le \alpha_5\delta_{\gamma,M,N}\Big(|p_k-p(u_{k-1})|_1+|y_k-y(u_{k-1})|_1\Big)+\alpha_6\delta_{\gamma,M,N}\|u_{k-1}-u^*\|_0,\end{aligned}\tag{4.26}$$

where

$$\alpha_5=\max\left\{\frac{1}{2}+C_\Omega^4(\rho+\alpha_2),\frac{3}{2}C_\Omega^2\right\},\quad\text{and}\quad\alpha_6=C_\Omega^3(1+\alpha_1+\alpha_2).$$

So we have

$$\begin{aligned}\|u_{k+1}-u^*\|_0 &\le \alpha_1\|u_k-u^*\|_0+\alpha_3\delta_{\gamma,M,N}^{\frac{1}{2}}\big(\|u_{k-1}-u^*\|_0+|p_k-p(u_{k-1})|_1\big)+\alpha_4\delta_{\gamma,M,N}^{-\frac{1}{2}}|p_{k+1}-p(u_k)|_1\\ &\le \alpha_1\|u_k-u^*\|_0+\big(\alpha_3+\alpha_4\alpha_5\big)\delta_{\gamma,M,N}^{\frac{1}{2}}\Big(|p_k-p(u_{k-1})|_1+|y_k-y(u_{k-1})|_1\Big)\\ &\quad+\big(\alpha_3+\alpha_4\alpha_6\big)\delta_{\gamma,M,N}^{\frac{1}{2}}\|u_{k-1}-u^*\|_0,\end{aligned}\tag{4.27}$$

such that

$$\begin{aligned}&\|u_{k+1}-u^*\|_0+|p_{k+1}-p(u_k)|_1+|y_{k+1}-y(u_k)|_1\\ &\quad\le \alpha_1\|u_k-u^*\|_0+\left(\alpha_3+\alpha_4\alpha_5+\left(\alpha_5+\frac{1}{2}\alpha_7\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\cdot\Big(|p_k-p(u_{k-1})|_1+|y_k-y(u_{k-1})|_1\Big)\\ &\qquad+\left(\alpha_3+\alpha_4\alpha_6+\left(\frac{1}{2}C_\Omega(1+\alpha_1+\alpha_2)+\alpha_6\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\|u_{k-1}-u^*\|_0.\end{aligned}\tag{4.28}$$

If

$$\left(\alpha_3+\alpha_4\alpha_5+\left(\alpha_5+\frac{1}{2}\alpha_7\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\right)\delta_{\gamma,M,N}^{\frac{1}{2}}\le\alpha_1,$$

then we have

$$\begin{aligned}\|u_{k+1}-u^*\|_0+|p_{k+1}-p(u_k)|_1+|y_{k+1}-y(u_k)|_1 &\le \alpha_1\Big(\|u_k-u^*\|_0+|y_k-y(u_{k-1})|_1+|p_k-p(u_{k-1})|_1\Big)\\ &\quad+\alpha_1^*(M,N)\|u_{k-1}-u^*\|_0.\end{aligned}\tag{4.29}$$

In the case of $0 < \rho < \underline{\rho}$, we have

$$\begin{aligned}\|u_{k+1} - u^*\|_0^2 \le\ & (1-\alpha\rho)^2\|u_k - u^*\|_0^2 + 2\rho(\rho-\bar{\rho})(\bar{\rho})^{-1}\|y(u_k) - y^*\|_0^2 - 2(1-\alpha\rho)\rho(u_k - u^*, p_{k+1} - p(u_k)) \\ & + \rho^2\|p_{k+1} - p(u_k)\|_0^2 + 2\rho^2(p(u_k) - p^*, p_{k+1} - p(u_k)) \\ \le\ & (1-\alpha\rho)^2\|u_k - u^*\|_0^2 + \rho^2\|p_{k+1} - p(u_k)\|_0^2 - 2(1-\alpha\rho)\rho(u_k - u^*, p_{k+1} - p(u_k)) \\ & + 2\rho^2(p(u_k) - p^*, p_{k+1} - p(u_k)).\end{aligned} \tag{4.30}$$

The conclusion and its proof are similar, only α_1 substituted for $\alpha_1 = |1 - \alpha\rho|$. □

Lemma 4.6. *Assume that the condition in Lemma 4.5 holds. Let*

$$\varepsilon_k^* \triangleq \|u_k - u^*\|_0 + |y_k - y(u_{k-1})|_1 + |p_k - p(u_{k-1})|_1. \tag{4.31}$$

Then there holds the estimation

$$\varepsilon_{k+1}^* \le \alpha_1\varepsilon_k^* + \alpha_1^*(M, N)\varepsilon_{k-1}^*. \tag{4.32}$$

That is the direct consequence of Lemma 4.5. Moreover, based on (4.32), we will derive convergence rate.

Lemma 4.7. *If a sequence of positive numbers $\{a_k\}$ satisfies*

$$a_{k+1} \le \gamma_1 a_k + \gamma_2 a_{k-1}, \tag{4.33}$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ satisfying $\gamma_1 + \gamma_2 < 1$, then there exist two constants c_1 and c_2 such that

$$a_k \le c_1 x_1^k + c_2 x_2^k, \tag{4.34}$$

where $0 < x_1 < 1$ and $-1 < x_2 < 0$ are the roots of equation

$$x^2 - \gamma_1 x - \gamma_2 = 0. \tag{4.35}$$

Proof. Noting the Eq. (4.35) has two roots as follows

$$x_1 = \frac{\gamma_1 + \sqrt{\gamma_1^2 + 4\gamma_2}}{2}, \qquad x_2 = \frac{\gamma_1 - \sqrt{\gamma_1^2 + 4\gamma_2}}{2}.$$

These are two real roots, $x_1 > 0$ and $x_2 < 0$. Since $\gamma_1 + \gamma_2 < 1$, hence

$$(1 - x_1)(x_2 - 1) = x_1 + x_2 - x_1x_2 - 1 = \gamma_1 + \gamma_2 - 1 < 0.$$

This implies $x_1 < 1$ and $x_2 = \gamma_1 - x_1 > -x_1 > -1$. Let (c_1, c_2) be the unique solution to the non-singular linear algebraic equation:

$$\begin{cases} a_1 = c_1x_1 + c_2x_2, \\ a_2 = c_1x_1^2 + c_2x_2^2. \end{cases} \tag{4.36}$$

This means (4.34) is truth for $k = 1, 2$. Suppose that (4.34) is truth for $1 \le k \le n$. We consider $k = n + 1$. Noting that $a_{n+1} \le \gamma_1 a_n + \gamma_2 a_{n-1}$ and $\gamma_1, \gamma_2 > 0$, we have

$$\begin{aligned} a_{n+1} &\le \gamma_1(c_1x_1^n + c_2x_2^n) + \gamma_2(c_1x_1^{n-1} + c_2x_2^{n-1}) \\ &= c_1x_1^{n-1}(\gamma_1x_1 + \gamma_2) + c_2x_2^{n-1}(\gamma_1x_2 + \gamma_2) \\ &= c_1x_1^n + c_2x_2^n. \end{aligned}$$

Thus (4.34) is truth for $k = n + 1$. Based upon the principle of mathematical induction, we have proved that (4.34) is truth for all $1 \le k < \infty$. □

Applying Lemma 4.7 to Lemma 4.6, we get the following lemma.

Lemma 4.8. *Assume that the condition in Lemma 4.5 holds. There exist two constants c_1 and c_2 such that*

$$\|u_k - u^*\|_0 + |y_k - y(u_{k-1})|_1 + |p_k - p(u_{k-1})|_1 \le c_1\lambda_1^k + c_2\lambda_2^k, \tag{4.37}$$

where $0 < \lambda_1 < 1$ and $-1 < \lambda_2 < 0$ are the roots of Eq. (4.8).

Table 1
SA–GP vs. comparable algorithm.

SA–GPA				Comparable algorithm			
Total number of inner loops: 83 + 78				Total number of inner loops: 195 + 173			
k	M	N	$\|u_{h,k} - u_h\|_0$	k	M	N	$\|u_{h,k} - u_h\|_0$
1	14	14	2.124e+00	1	50	50	2.081e+00
2	14	14	5.186e−03	2	46	42	5.339e−03
3	14	14	1.263e−05	3	37	33	1.370e−05
4	14	14	3.076e−08	4	29	24	3.516e−08
5	14	14	7.585e−11	5	20	16	9.763e−11
6	11	7	2.542e−12	6	11	7	9.040e−12
7	2	1	2.387e−12	7	2	1	8.769e−12

Now we can prove Theorem 4.1.

Proof of Theorem 4.1. Noting that

$$|y_k - y^*|_1 \le |y_k - y(u_{k-1})| + |y^* - y(u_{k-1})|_1 \le |y_k - y(u_{k-1})|_1 + C_\Omega \|u^* - u_{k-1}\|_0,$$

and

$$|p_k - p^*|_1 \le |p_k - p(u_{k-1})| + |p^* - p(u_{k-1})|_1 = |y_k - y(u_{k-1})|_1 + |\mathcal{S}(y^* - y(u_{k-1}))|_1,$$

and applying Lemma 4.8, we derive (4.5). □

5. Numerical experiments

In this section, we perform some numerical tests to check theoretical results. We investigate the model problem on the domain $\Omega = [0, 1] \times [0, 1] \in \mathbb{R}^2$:

$$\begin{cases} \min\limits_{u \in \mathcal{U}_{\text{ad}}} & \mathcal{J}(u, y) = \dfrac{1}{2}\displaystyle\int_\Omega (y - y_d)^2 + \dfrac{1}{2}\displaystyle\int_\Omega u^2, \\ \text{s.t.} & -\Delta y = u + f \quad \text{in } \Omega,\ y = 0 \text{ on } \partial\Omega. \end{cases}$$

Set U^h and V^h to be the piecewise linear finite element spaces with the mesh size $h = .005$ and $\delta = 0.05$. We choose parameter $C_\Omega = \frac{1}{\sqrt{2}\pi}$ such that $\underline{\rho} = \frac{2}{2\alpha + C_\Omega^4} = 0.9987$ and $\bar{\rho} = \frac{2}{\alpha + C_\Omega^4} = 1.9949$. In our numerical examples, we take $\rho \approx 1$ by (4.9) so that $\alpha_1 = |1 - (\alpha + C_\Omega^4)\rho| = 0.0026$.

We introduce a comparable algorithm to show the efficiency of the proposed inner iteration number. The iteration in the inner loops is controlled by the tolerance which is the same as that of the outer loops.

5.1. Numerical experiments of two sub-domains

We consider two sub-domain decomposition $\Omega = \Omega_1 \bigcup \Omega_2$ where $\Omega_1 = \left[0, \frac{1}{2} + \delta\right] \times [0, 1]$ and $\Omega_2 = \left[\frac{1}{2} - \delta, 1\right] \times [0, 1]$. By numerical results, we get $\tilde{\gamma} \approx 0.5$ when $\delta = 0.05$. Further, we have $M \ge 13.6476$ by (3.9) and take $M = N = 14$. The tolerance of the outer loops $\epsilon = 1.0 \times 10^{-12}$. The initial value of the control and state variables are set to be $u_0 = 1.0 \times 10^3$, $p_0 = 0$, and $y_0 = 0$. We perform two numerical tests in the same meshes and the initial approximations.

5.1.1. Numerical example 1

In this example, $\mathcal{U}_{\text{ad}} = U$. The related exact solution is

$$\begin{cases} y = \dfrac{1}{4}\sin 2\pi x_1 \sin 2\pi x_2, \\ p = 2y, \\ u = -p, \\ y_d = y - 8\pi^2 p, \\ f = 8\pi^2 y + p. \end{cases}$$

The numerical results are put into Table 1, in which k is the counter of the outer loops and M, N are the counters of the inner loops.

5.1.2. Numerical experiment 2

In this example, we consider a constrained optimal control. Let

$$\mathcal{U}_{\text{ad}} = \left\{w \in U : w \ge 0 \text{ a.e. in } \Omega\right\}.$$

Table 2
SA–GP vs. comparable algorithm.

SA–GPA				Comparable algorithm			
Total number of inner loops: 67 + 62				Total number of inner loops: 170 + 149			
k	M	N	$\|u_{h,k} - u_h\|_0$	k	M	N	$\|u_{h,k} - u_h\|_0$
1	14	14	1.768e−01	1	50	50	1.768e−01
2	14	14	4.072e−05	2	46	42	1.742e−04
3	14	14	1.808e−08	3	33	28	2.311e−07
4	14	14	2.076e−11	4	23	19	2.998e−10
5	10	5	1.834e−12	5	14	9	6.302e−12
6	1	1	1.610e−12	6	4	1	6.384e−12

Table 3
SA–GP vs. comparable algorithm.

Domain number		SA–GPA				Comparable algorithm			
s_x	s_y	N_{out}	N^1_{in}	N^2_{in}	e	N_{out}	N^1_{in}	N^2_{in}	e
2	2	4	40	37	3.822e−9	4	78	60	3.793e−9
4	4	5	73	47	6.149e−9	4	114	91	6.211e−9

The corresponding exact solution is

$$\begin{cases} y = \dfrac{1}{4}\sin 2\pi x_1 \sin 2\pi x_2, \\ p = 2y, \\ u = \max\{0, -p\}, \\ y_d = y - 8\pi^2 p, \\ f = 8\pi^2 y - u. \end{cases}$$

The numerical results are put into Table 2.

5.2. Numerical experiments of more sub-domains

We keep the same overlapping degree $\delta = 0.05$. Thus the proposed iteration number in the inner loops is the same as the above. Set the initial values $u_0 = y_0 = p_0 = 0$, $\epsilon = 1.0\text{e} - 8$, and $\rho = 1$. And we supply the examples without explicit solution as follows:

$$\begin{cases} y_d = 10\text{e}^{x_1^2+x_2^2}\sin(4\pi xy) - 1, \\ f = x(1+y) + 1, \end{cases}$$

with $U_{\text{ad}} = \{u \in L^2(\Omega); u \geq 0 \text{ a.e. in } \Omega\}$. The domain is decomposed into $m = s_x \times s_y$ sub-domains $\Omega = \bigcup_{i,j=1}^{i=s_x, j=s_y} \Omega_{i,j}$ as follows: first define $H_x = \frac{1}{s_x}, H_y = \frac{1}{s_y}$. Then interval $[0, 1]$ is divided into sub-intervals as $[0, 1] = \bigcup_{i=1}^{s_x} A_x^i$, and $[0, 1] = \bigcup_{j=1}^{s_y} A_y^j$, with

$$A_x^i = [\max\{0, (i-1)H_x - \delta\}, \min\{1, iH_x + \delta\}],$$
$$A_y^j = \left[\max\{0, (j-1)H_y - \delta\}, \min\{1, jH_y + \delta\}\right].$$

Thus sub-domain $\Omega_{i,j} := A_x^i \times A_y^j$, for $1 \leq i \leq s_x$ and $1 \leq j \leq s_y$. The cases with $m = 4$ and $m = 16$ are tested. Let N_{out} be the total iteration number of the outer loops, N^1_{in} be the total iteration number of state variable y in the inner loops, and N^2_{in} be the total iteration number of co-state variable p. Let e be L^2-norm of control variable error between the solution by the gradient projection algorithm without DDM and the solutions of SA–GP algorithm, the comparable algorithm, respectively. The numerical results are put in Table 3.

See Figs. 1 and 2 for numerical solution to the problem.

From these numerical experiments, we see that the total iteration number in the inner loops is about one half of that of the comparable algorithm. In the example of more sub-domains, if we fix the overlapping degree δ, the total iteration number in the inner loops is also about one half of that of the comparable algorithm. From the examples of two sub-domains we see that the computation cost is saved in the first fourth iterations of the outer loops when the error in the outer loops is much larger than that in the inner loops. Thus by using the proposed iteration number in the inner loops, a lot of computational cost is saved. That verifies the theoretical results given in Section 4.

6. Conclusion

We have proposed and analyzed the SA–GP method to solve the control constrained optimal control problem. The gradient projection method is used to solve the variational inequality with respect to the control variable as the outer

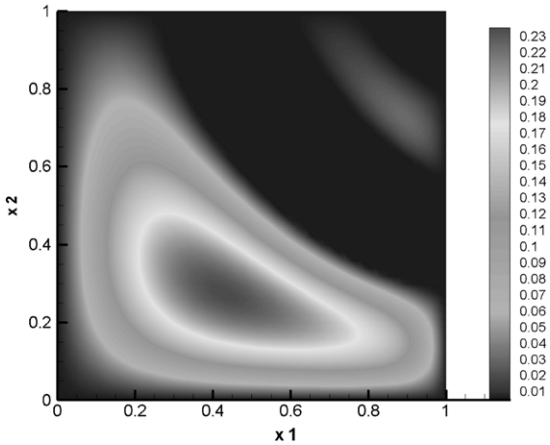

Fig. 1. u_h with $h = 0.02$.

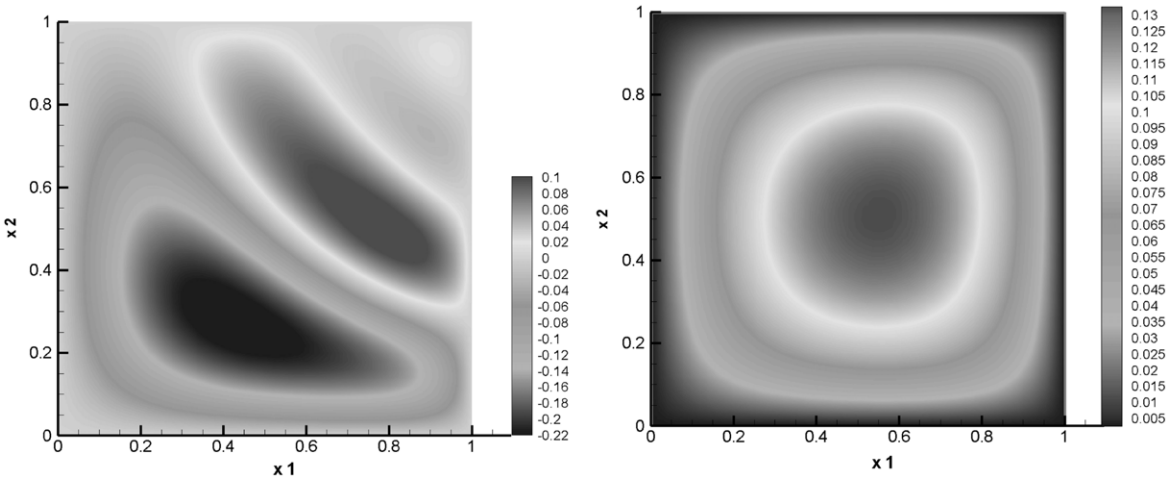

Fig. 2. p_h, y_h with $h = 0.02$.

loops, and the Schwarz alternating scheme to treat PDEs with respect to the state variables as the inner loops. Different from a classical algorithm, we propose and analyze an inner iteration number independent of the tolerance. By using this algorithm, a lot of computational cost is saved. In our numerical tests, almost one half of iteration cost is saved comparing with the algorithm which uses the same given tolerance in the inner loops as that in outer loops.

As we know, the gradient projection method in the outer loops is the first-order algorithm. It is the globally decent and convergent method with suitable steps. To derive much faster algorithm, the Newton-type methods of second-order convergence rate were discussed in [37,38]. When applying these second-order methods, how to use domain decomposition methods and how to control the iteration number in the inner loops are still a problem similar to that in this paper. There are several new difficult due to the control constraints and use of generalized derivative in [39]. That will be an interesting topic we intend to do in the future.

Acknowledgments

First author's research was partially supported by MOE IDM project NRF2007IDM-IDM002-010, Singapore, and Outstanding Doctor Fund with Grant No. 2010026 of East China Normal University, China. Second author's research was partially supported by the National Natural Science Foundation of China, Grant 11071080, Program of Shanghai Subject Chief Scientist, No. 09XD1401600, Fundamental Research Funds for the Central Universities of China and Shanghai Leading Academic Discipline Project B407.

References

[1] A. Quarteroni, A. Valli, Domain Decompositon Methods for Partial Differential Equations, Oxford Science Publications, 1999.
[2] B.F. Smith, P.E. Bjorstad, W.D. Gropp, Domain Decompositons: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 1996.
[3] A. Toselli, O. Widlund, Domain Decomposition Methods—Algorithm and Theory, Springer-Verlag, Berlin, Heidelberg, 2005.
[4] J.C. Xu, Iterative methods by space decomposition and subspace correction, SIAM Rev. 34 (4) (1992) 581–613.
[5] J.C. Xu, J. Zou, Some nonoverlapping domain decomposition methods, SIAM Rev. 40 (4) (1998) 857–914.
[6] A. Battermann, M. Heinkenschloss, Preconditioners for Karush–Kuhn–Tucker matrices arising in the optimal control of distributed systems, in: W. Desch, F. Kappel, K. Kunisch (Eds.), Control and Estimation of Distributed Parameter Systems, in: Internat. Ser. Numer. Math., vol. 126, Birkhäuser-Verlag, Basel, 1998, pp. 15–32.
[7] A. Battermannn, E.W. Sachs, Block preconditioners for KKT systems in PDE-governed optimal control problems, in: K.H. Hoffmann, R.H.W. Hoppe, V. Schulz (Eds.), Fast Solution of Discretized Optimization Problems, in: Internat. Ser. Numer. Math., vol. 138, Birkhäuser-Verlag, Basel, 2001, pp. 1–18.
[8] R. Bartlett, M. Heinkenschloss, D. Ridzal, B. van Bloemen Waanders, Domain decomposition methods for advection domainated linear-quadratic elliptic optimal control problems, Comput. Methods Appl. Mech. Engrg. 195 (44–47) (2006) 6428–6447.
[9] J.D. Benamou, A domain decompostion method with coupled transmission conditons for the optimal control of systems governed by elliptic parital differential equations, SIAM J. Numer. Anal. 33 (6) (1996) 2401–2416.
[10] J.D. Benamou, Domain decompositon methods with coupled transmission conditons for the optimal control of systems governed by elliptic partial differential equations, C. R. Acad. Sci. Paris 317 (1993).
[11] A. Bounaïm, A Lagrangian approach to a DDM for an optimal control problem, in: P.E. Bjorstad, M.S. Espedal, D.E. Keyes (Eds.), Ninth International Conference on Domain Decompositon Methods, 1998.
[12] A. Bounaïm, On the use of iterative Schwarz algorithms in the solution of an optimal control problem, in: N. Debit, M. Garbey, R. Hoppe, J. Périaux, D.E. Keyes, Y. Kuznetsov (Eds.), Thirteenth International Conference on Domain Decompositon Methods, 1998.
[13] G. Biros, O. Ghattas, Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: the Krylov–Schur solver, SIAM J. Sci. Comput. 27 (2005) 687–713.
[14] X.C. Cai, S. Liu, J. Zou, Parallel overlapping domain decomposition methods for coupled inverse elliptic problems, Commun. Appl. Math. Comput. Sci. 4 (2009) 1–26.
[15] M.D. Gunzburger, J. Lee, A domain decompositon method for optimization problems for partial differential equations, Comput. Math. Appl. 40 (2000) 177–192.
[16] M. Heinkenschloss, H. Nguyen, Balancing Neumann–Neumann methods for elliptic optimal control problems, in: R. Kornhuber, R.H.W. Hoppe, J. Periaux, O. Pironneau, O.B. Widlund, J. Xu (Eds.), Domain Decomposition Methods in Science and Engineering, in: Lecture Notes in Computational Science and Engineering, vol. 40, Springer, Heidelberg, 2004, pp. 589–596.
[17] M. Heinkenschloss, H. Nguyen, Neumann–Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems, SIAM J. Sci. Comput. 28 (3) (2006) 1001–1028.
[18] K. Kunisch, X.C. Tai, Nonoverlapping domain decomposition methods for inverse problems, in: Proceedings of the 9th International Conference on Domain Decomposition Methods, Domain Decomposition Press, Bergen, Norway, 1998, pp. 517–524.
[19] J.L. Lions, Bensoussan, R. Glowinski, Méthode de décompositin applquée au contrôle optimal de systèmes distribués, in: Fifth IFIP Conference on Optimization Techniques, in: Lecture Notes in Comput. Sci., Springer, New York, 1973, pp. 141–151.
[20] H.Q. Nguyen, Domain decomposition methods for linear-quadratic elliptic optimal control problems, Ph.D. Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2004. Available as CAAM TR04-16.
[21] E.E. Prudencio, R. Byrd, X.C. Cai, Parallel full space SQP Lagrange–Newton–Krylov–Schwarz algorithms for PDE-constrained optimization problems, SIAM J. Sci. Comput. 27 (4) (2006) 1305–1328.
[22] X.C. Tai, J. Frøyen, M.S. Espedal, T.F. Chan, Overlapping domain decomposition and multigrid methods for inverse problems, in: Domain Decomposition Methods 10, Boulder, CO, 1997, in: Contemp. Math., vol. 218, AMS, Providence, RI, 1998, pp. 523–529.
[23] J.D. Benamou, Domain decomposition, optimal control of systems governed by partial differential equations, and synthesis of feedback laws, J. Optim. Theory Appl. 102 (1) (1999) 15–36.
[24] M. Heinkenschloss, M. Herty, Distributed solution of optimal control problems governed by parabolic equations, in: A.J. Kurdila, P.M. Pardalos, M. Zabarankin (Eds.), Robust Optimization: Directed Design, in: Nonvex Optimization and its Applications, vol. 81, Springer, Heidelberg, 2006, pp. 71–90.
[25] M. Heinkenschloss, M. Herty, A spatial domain decomposition method for parabolic optimal control problems, J. Comput. Appl. Math. 201 (2007) 88–111.
[26] M. Heinkenschloss, Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problem, J. Comput. Appl. Math. 173 (2005) 169–198.
[27] P.L. Lions, in: T. Chan, R. Glowinski, J. Périaux, O.B. Widlund (Eds.), On Schwarz Alternating Method I, Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1989, pp. 401–430.
[28] P.L. Lions, On the Schwarz Alternating Method II, SIAM, Philadelphia, PA, 1989, pp. 47–70.
[29] P.L. Lions, On the Schwarz Alternating Method III, SIAM, Philadelphia, PA, 1990, pp. 202–223.
[30] J.L. Lions, Optimal Control of Systems Governed by Partial Differential Equations, Springer-Verlag, Berlin, 1971.
[31] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, Springer, NewYork, 1994.
[32] P.G. Ciarlet, The Finite Element Methods for Elliptic Problems, North-Holland Publishing Company, Amsterdam, 1978.
[33] W.B. Liu, N.N. Yan, Adaptive Finite Element Methods for Optimal Control Governed by PDEs, Science Press, Beijing, 2008.
[34] L.H. Wang, X.J. Xu, The Mathematical Foundation of Finite Element Methods, vol. 26, Science Press, Beijing, 2004 (in Chinese).
[35] S.N. He, L.S. Zhang, Optimal constant of poincaré inequality on rectangle regions, J. Civ. Aviat. Univ. China 26 (1) (2008) 59–64 (in Chinese).
[36] D. Gilbarg, N.S. Trudinger, Elliptic Partial Differential Equations of Second Order, Springer, 1998.
[37] M. Hintermüller, M. Ulbrich, A mesh-independence result for semismooth Newton methods, Math. Program. Ser. B 101 (2004) 151–184.
[38] M. Hintermüller, K. Ito, K. Kunisch, The primal–dual active set strategy as a semismooth Newton method, SIAM J. Optim. 13 (3) (2003) 865–888.
[39] F.H. Clarke, Optimization and Nonsmooth Analysis, Wiley-Interscience, New York, 1983.