First Order Numerical Algorithms for Some Optimal Control Problems with PDE Constraints

Yongcun Song

Chair for Dynamics, Control and Numerics

Alexander von Humboldt Professorship

Department of Data Science

Friedrich-Alexander-Universität Erlangen-Nürnberg

October 1, 2021

Outline

Outline

Optimal Control

- Optimal control aims at controlling a given system over a period of time such that a certain goal is achieved.

Optimal Control

- Optimal control aims at controlling a given system over a period of time such that a certain goal is achieved.
- The system can be mathematically described by ordinary differential equations (ODEs), partial differential equations (PDEs), integral equations, difference equations, etc.

Optimal Control

- Optimal control aims at controlling a given system over a period of time such that a certain goal is achieved.
- The system can be mathematically described by ordinary differential equations (ODEs), partial differential equations (PDEs), integral equations, difference equations, etc.
- We focus on some PDE-related optimal control problems.

Conceptual Model

- An optimal control problem with PDE constraints can be abstractly represented as

$$\min_{u\in U, y\in Y} J(u,y), \quad \text{s.t.} \quad e(u,y)=0, u\in U_{ad}, y\in Y_{ad},$$

  - U and Y are Banach spaces, $U_{ad}\subset U$ and $Y_{ad}\subset Y$ are closed convex sets;
  - $J: U\times Y\to\mathbb{R}$ is the objective functional;
  - $e(u,y)=0$ represents a PDE or a system of coupled PDEs;
  - the state variable $y\in Y$ describes the state (e.g., temperature distribution) of the considered system modeled by $e(u,y)=0$;
  - the control variable $u\in U$ is a parameter (e.g., source term) that shall be adapted in an optimal way;
  - the control constraint $u\in U_{ad}$ and the state constraint $y\in Y_{ad}$ describe some physical restrictions and realistic requirements.

State-of-the-Art

- There are many works for theoretical analysis of the existence and regularity of the optimal solution, numerical discretization schemes, and applications perspectives (e.g., J. L. Lions 1971, J. L. Lions and R. Glowinski 1994, F. Tröltzsch 2010, J. L. Lions, R. Glowinski and J. He 2008, M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich 2009, etc.).
- We aim at algorithmic design for these problems.

Challenges

Challenges

- From PDE perspectives
  - Models are generally complex, e.g., the coupling of PDEs with other constraints.
  - Solving the involved PDEs is already not easy.
  - Results for PDEs (numerical scheme, error estimation, etc.) cannot be extended directly.

Challenges

- From PDE perspectives
  - Models are generally complex, e.g., the coupling of PDEs with other constraints.
  - Solving the involved PDEs is already not easy.
  - Results for PDEs (numerical scheme, error estimation, etc.) cannot be extended directly.
- From optimization perspectives
  - High dimensionality after discretization of PDEs, especially for time-dependent cases.
  - Extremely ill-conditioned linear systems.
  - Theoretical obstacles (nonsmoothness, nonconvexity, infinite-dimensional spaces, etc.).

Challenges

- From PDE perspectives
  - Models are generally complex, e.g., the coupling of PDEs with other constraints.
  - Solving the involved PDEs is already not easy.
  - Results for PDEs (numerical scheme, error estimation, etc.) cannot be extended directly.
- From optimization perspectives
  - High dimensionality after discretization of PDEs, especially for time-dependent cases.
  - Extremely ill-conditioned linear systems.
  - Theoretical obstacles (nonsmoothness, nonconvexity, infinite-dimensional spaces, etc.).
- Simply combining off-the-shelf PDE solvers and optimization algorithms does not work.

Challenges

- From PDE perspectives
  - Models are generally complex, e.g., the coupling of PDEs with other constraints.
  - Solving the involved PDEs is already not easy.
  - Results for PDEs (numerical scheme, error estimation, etc.) cannot be extended directly.
- From optimization perspectives
  - High dimensionality after discretization of PDEs, especially for time-dependent cases.
  - Extremely ill-conditioned linear systems.
  - Theoretical obstacles (nonsmoothness, nonconvexity, infinite-dimensional spaces, etc.).
- Simply combining off-the-shelf PDE solvers and optimization algorithms does not work. Specific structures and properties of the model should be considered deliberately.

Outline

Model

- We focus on some PDE-constrained optimal control problems with control constraints that can be unified as:

$$\min_{u\in U, y\in Y} \quad \frac{1}{2}\|y-y_d\|_Y^2 + \frac{\alpha}{2}\|u\|_U^2 + \theta(u)$$
$$\text{s.t.} \quad y = Su,$$

  - U and Y are proper function spaces, $u \in U$ and $y \in Y$ are called the control variable and state variable, respectively;
  - the constant $\alpha > 0$ is a regularization parameter;
  - the function y_d is a given target;
  - $S : U \to Y$ is a solution operator associated with some linear PDEs (e.g., elliptic equations and parabolic equations);
  - the nonsmooth convex function $\theta : U \to \mathbb{R}$ represents some additional box or sparsity constraints on the control variable u.

Quick Review for some Existing Methods

- Semismooth Newton methods, e.g., K. Ito and K. Kunisch 2008, M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich 2009.
- Interior point methods, e.g., J. W. Pearson and J. Gondzio 2017.

Our Recipe

- Decoupling PDE and control constraints by some operator splitting strategies, and solving the decoupled subproblems inexactly, while guaranteeing the overall convergence rigorously.

Our Recipe

- Decoupling PDE and control constraints by some operator splitting strategies, and solving the decoupled subproblems inexactly, while guaranteeing the overall convergence rigorously.
- Targets: easy to implement, appropriate accuracy for subproblems, mesh-independent, rigorously guaranteed convergence.

Outcomes

- Inexact Uzawa algorithmic framework [Y. Song, X. Yuan, and H. Yue 2019].
- Inexact alternating direction method of multipliers (ADMM) [R. Glowinski, Y. Song, X. Yuan, and H. Yue 2021].
- Some primal-dual hybrid gradient (PDHG) type methods [U. Biccari, Y. Song, X. Yuan, and E. Zuaua 2021].

An Elliptic Optimal Control Problem with Control Constraints

- We consider the following elliptic optimal control problem with control constraint:

$$\min_{y\in H_0^1(\Omega),u\in L^2(\Omega)} J(y,u)=\frac{1}{2}\|y-y_d\|^2_{L^2(\Omega)}+\frac{\alpha}{2}\|u\|^2_{L^2(\Omega)}+\theta(u),$$

where y and u satisfy the elliptic equation

$$\begin{cases} \mathcal{K}y=u & \text{in} \quad \Omega, \\ \quad y=0 & \text{on} \quad \Gamma. \end{cases}$$

  - The domain $\Omega\subset\mathbb{R}^d (d\geq 1)$ is convex and bounded and its boundary $\Gamma=\partial\Omega$ is Lipschitz continuous.
  - The given target $y_d\in L^2(\Omega)$.
  - $\mathcal{K}$ is a linear second-order elliptic operator.
  - $\theta(u)$ is the indicator function of the admissible set U_{ad}:

$$U_{ad}=\{u\in L^\infty(\Omega)|a\leq u(x)\leq b, \text{ a.e. in } \Omega\}\subset L^2(\Omega),$$

where $-\infty<a<b<+\infty$ are two given constants.

Optimality Condition

- We discretize the optimal control problem by the standard piecewise linear finite element method.
- The optimality condition for the discretized optimal control problem reads as:

$$\begin{pmatrix} \alpha M + \partial\theta & 0 & M^\top \\ 0 & M & -K^\top \\ M & -K & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{y} \\ \boldsymbol{p} \end{pmatrix} \ni \begin{pmatrix} 0 \\ M\boldsymbol{y}_d \\ 0 \end{pmatrix},$$

  where K and M are discrete matrices associated with $\mathcal{K}$ and $(\cdot,\cdot)_{L^2(\Omega)}$, respectively; and $\partial\theta(u)$ is the sub-differential of θ.
- Both K and M are symmetric and positive definite.

A Nonlinear Saddle Point Problem

- We first introduce

$$A = \begin{pmatrix} \alpha M & 0 \\ 0 & M \end{pmatrix}, B = \begin{pmatrix} M & -K \end{pmatrix}, w = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{y} \end{pmatrix},$$

$$v = \boldsymbol{p}, \quad \Theta(w) = \theta(\boldsymbol{u}),\ f = \begin{pmatrix} 0 \\ M\boldsymbol{y}_d \end{pmatrix},\ g = 0.$$

- The discretized optimality condition can be reformulated as:

$$\begin{pmatrix} A + \partial\Theta & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} \ni \begin{pmatrix} f \\ g \end{pmatrix}.$$

An Inexact Uzawa Algorithmic Framework

- To solve the above nonlinear saddle point problem, we propose the following inexact Uzawa algorithmic framework:

$$\begin{cases} w^{k+1} = (Q_A + \partial\Theta)^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g). \end{cases}$$

An Inexact Uzawa Algorithmic Framework

- To solve the above nonlinear saddle point problem, we propose the following inexact Uzawa algorithmic framework:

$$\begin{cases} w^{k+1} = (Q_A + \partial\Theta)^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g). \end{cases}$$

- Q_A and Q_B are two symmetric positive definite preconditioners of A and $BA^{-1}B^\top$ satisfying $Q_A \succeq A$ and $Q_B \succeq BA^{-1}B^\top$, respectively.

An Inexact Uzawa Algorithmic Framework

- To solve the above nonlinear saddle point problem, we propose the following inexact Uzawa algorithmic framework:

$$\begin{cases} w^{k+1} = (Q_A + \partial\Theta)^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g). \end{cases}$$

- Q_A and Q_B are two symmetric positive definite preconditioners of A and $BA^{-1}B^\top$ satisfying $Q_A \succeq A$ and $Q_B \succeq BA^{-1}B^\top$, respectively.
- When $Q_A = A$ and $Q_B = \frac{1}{\omega}I$, it reduces to the exact version.

An Inexact Uzawa Algorithmic Framework

- To solve the above nonlinear saddle point problem, we propose the following inexact Uzawa algorithmic framework:

$$\begin{cases} w^{k+1} = (Q_A + \partial\Theta)^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g). \end{cases}$$

- Q_A and Q_B are two symmetric positive definite preconditioners of A and $BA^{-1}B^\top$ satisfying $Q_A \succeq A$ and $Q_B \succeq BA^{-1}B^\top$, respectively.
- When $Q_A = A$ and $Q_B = \frac{1}{\omega}I$, it reduces to the exact version.
- The convergence and linear convergence rate can be rigorously proved under mild conditions.

An Inexact Uzawa Algorithmic Framework

- To solve the above nonlinear saddle point problem, we propose the following inexact Uzawa algorithmic framework:

$$\begin{cases} w^{k+1} = (Q_A + \partial\Theta)^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g). \end{cases}$$

- Q_A and Q_B are two symmetric positive definite preconditioners of A and $BA^{-1}B^\top$ satisfying $Q_A \succeq A$ and $Q_B \succeq BA^{-1}B^\top$, respectively.
- When $Q_A = A$ and $Q_B = \frac{1}{\omega}I$, it reduces to the exact version.
- The convergence and linear convergence rate can be rigorously proved under mild conditions.
- Above algorithmic framework is abstract and it becomes practical only when the preconditioners Q_A and Q_B are chosen appropriately.

Choices of Q_A and Q_B

- To choose appropriate Q_A and Q_B, we start from considering the application of the exact Uzawa method, i.e. $Q_A = A$ and $Q_B = \frac{1}{\omega} I(\omega > 0)$:

$$\begin{cases} 0 \in \partial\theta(\boldsymbol{u}^{k+1}) + \alpha M \boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

Choices of Q_A and Q_B

- To choose appropriate Q_A and Q_B, we start from considering the application of the exact Uzawa method, i.e. $Q_A = A$ and $Q_B = \frac{1}{\omega} I (\omega > 0)$:

$$\begin{cases} 0 \in \partial\theta(\boldsymbol{u}^{k+1}) + \alpha M \boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

- Numerical issues:

Choices of Q_A and Q_B

- To choose appropriate Q_A and Q_B, we start from considering the application of the exact Uzawa method, i.e. $Q_A = A$ and $Q_B = \frac{1}{\omega} I (\omega > 0)$:

$$\begin{cases} 0 \in \partial\theta(\boldsymbol{u}^{k+1}) + \alpha M\boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

- Numerical issues:
  - since M is not diagonal, u^{k+1} has no closed-form solution;

Choices of Q_A and Q_B

- To choose appropriate Q_A and Q_B, we start from considering the application of the exact Uzawa method, i.e. $Q_A = A$ and $Q_B = \frac{1}{\omega} I (\omega > 0)$:

$$\begin{cases} 0 \in \partial\theta(\boldsymbol{u}^{k+1}) + \alpha M \boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

- Numerical issues:
  - since M is not diagonal, u^{k+1} has no closed-form solution;
  - computing y^{k+1} requires the solution of a linear system which is large dimensional especially for the fine discretization case;

Choices of Q_A and Q_B

- To choose appropriate Q_A and Q_B, we start from considering the application of the exact Uzawa method, i.e. $Q_A = A$ and $Q_B = \frac{1}{\omega} I (\omega > 0)$:

$$\begin{cases} 0 \in \partial\theta(\boldsymbol{u}^{k+1}) + \alpha M \boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

- Numerical issues:
  - since M is not diagonal, u^{k+1} has no closed-form solution;
  - computing y^{k+1} requires the solution of a linear system which is large dimensional especially for the fine discretization case;
  - to guarantee the convergence, it is required that $Q_B \succeq S := BA^{-1}B^\top = \frac{1}{\alpha} M + KM^{-1}K^\top$ which implies that $\omega \leq \frac{1}{\rho(S)}$ (order $O(h^2)$) — leading to slow convergence.

Choice of Q_A

- First, consider the lumped mass matrix

$$W := diag(\int_{\Omega_h} \phi_i(x)dx)_{i=1}^n,$$

which satisfies $W \succeq M$ for any $h > 0$ (see e.g., [A.J. Wathen, 1987]).

- Adding the proximal term $\frac{\alpha}{2}\|\boldsymbol{u} - \boldsymbol{u^k}\|_{W-M}$ to the objective function, the u-subproblem is transformed to

$$\min_{\boldsymbol{u}\in\mathbb{R}^n} \theta(\boldsymbol{u}) + \frac{\alpha}{2}\|\boldsymbol{u} + W^{-1}(\frac{1}{\alpha}M^\top \boldsymbol{p}^k - (W - M)\boldsymbol{u}^k)\|_W^2.$$

- Since W is diagonal, we obtain

$$\boldsymbol{u}^{k+1} = P_{\boldsymbol{U_{ad}}}(W^{-1}((W - M)\boldsymbol{u}^k - \frac{1}{\alpha}M^\top \boldsymbol{p}^k)).$$

Choice of Q_A – Cont'd

- For the second difficulty, we consider some iterative schemes that are tailored for the system of linear equations.
- We choose $D = 2diag(M)$ and compute $\boldsymbol{y}^{k+1}$ via

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d),$$

  which is essentially the application of the damped Jacobi iteration method to the $\boldsymbol{y}$-subproblem.
- As a result, we can update $\boldsymbol{y}^{k+1}$ element-wisely, which is easy and cheap to implement.

Choice of Q_A – Cont'd

- Preconditioner Q_A:

$$Q_A := \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix} \succeq A,$$

where

  - $W := diag(\int_{\Omega_h} \phi_i(x)dx)_{i=1}^n$ is the lump mass matrix satisfying $W \succeq M$ [A. J. Wathen, 1987];
  - $D = 2diag(M) \succ M$ since M is diagonally dominant.

Choice of Q_A – Cont'd

- Preconditioner Q_A:

$$Q_A := \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix} \succeq A,$$

  where

  - $W := diag(\int_{\Omega_h} \phi_i(x)dx)_{i=1}^n$ is the lump mass matrix satisfying $W \succeq M$ [A. J. Wathen, 1987];
  - $D = 2diag(M) \succ M$ since M is diagonally dominant.

- Accordingly, we obtain

$$\begin{cases} \boldsymbol{u}^{k+1} = P_{\boldsymbol{U}_{ad}}(W^{-1}((W - M)\boldsymbol{u}^k - \frac{1}{\alpha}M^\top \boldsymbol{p}^k)), \\ \boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d). \end{cases}$$

Choice of Q_B

- Note that the Schur complement $S = \frac{1}{\alpha}M + KM^{-1}K^\top$ can be written as

$$S = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top - \frac{2}{\sqrt{\alpha}}K.$$

- According to [J. W. Pearson and A. J. Wathen, 2012], S can be well approximated by

$$P_B := (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top \succ S.$$

Theorem

Suppose that we approximate S by P_B. Then, we can bound the eigenvalues of $P_B^{-1}S$ as follows:

$$\lambda(P_B^{-1}S) \in [\frac{1}{2},\ 1],$$

which is independent of α and h.

Choice of Q_B – Cont'd

- For $P_B = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top$, the matrix $K + \frac{1}{\sqrt{\alpha}}M$ is still ill-conditioned especially for fine mesh sizes, due to the stiffness matrix K whose condition number is of $O(h^{-2})$.

Choice of Q_B – Cont'd

- For $P_B = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^{\top}$, the matrix $K + \frac{1}{\sqrt{\alpha}}M$ is still ill-conditioned especially for fine mesh sizes, due to the stiffness matrix K whose condition number is of $O(h^{-2})$.
- The matrix K is discretized from the linear second-order elliptic operator $\mathcal{K}$, a spectrally equivalent approximation of K can be obtained by performing one or more multi-grid sweeps.

Choice of Q_B – Cont'd

- For $P_B = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top$, the matrix $K + \frac{1}{\sqrt{\alpha}}M$ is still ill-conditioned especially for fine mesh sizes, due to the stiffness matrix K whose condition number is of $O(h^{-2})$.
- The matrix K is discretized from the linear second-order elliptic operator $\mathcal{K}$, a spectrally equivalent approximation of K can be obtained by performing one or more multi-grid sweeps.
- Via the implementation of AMG V-circles, the matrix $(K + \frac{1}{\sqrt{\alpha}}M)$ is implicitly approximated by a matrix, denoted by G.

Choice of Q_B – Cont'd

- For $P_B = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top$, the matrix $K + \frac{1}{\sqrt{\alpha}}M$ is still ill-conditioned especially for fine mesh sizes, due to the stiffness matrix K whose condition number is of $O(h^{-2})$.
- The matrix K is discretized from the linear second-order elliptic operator $\mathcal{K}$, a spectrally equivalent approximation of K can be obtained by performing one or more multi-grid sweeps.
- Via the implementation of AMG V-circles, the matrix $(K + \frac{1}{\sqrt{\alpha}}M)$ is implicitly approximated by a matrix, denoted by G.
- Therefore we can choose Q_B as $Q_B := \tau G M^{-1} G^\top (\tau > 0)$.

Specifying the Inexact Uzawa Algorithmic Framework

- With the choice of $Q_A = \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix}$ and $Q_B = \tau G M^{-1} G^\top \succeq P_B \succ S$, we specify the inexact Uzawa algorithmic framework as

$$\begin{cases} \boldsymbol{u}^{k+1} = P_{\boldsymbol{U}_{ad}}(W^{-1}((W - M)\boldsymbol{u}^k - \frac{1}{\alpha} M^\top \boldsymbol{p}^k)), \\ \boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d), \\ \boldsymbol{p}^{k+1} = \boldsymbol{p}^k + Q_B^{-1}(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

Specifying the Inexact Uzawa Algorithmic Framework

- With the choice of $Q_A = \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix}$ and $Q_B = \tau G M^{-1} G^\top \succeq P_B \succ S$, we specify the inexact Uzawa algorithmic framework as

$$\begin{cases} \boldsymbol{u}^{k+1} = P_{\boldsymbol{U_{ad}}}(W^{-1}((W-M)\boldsymbol{u}^k - \frac{1}{\alpha} M^\top \boldsymbol{p}^k)), \\ \boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d), \\ \boldsymbol{p}^{k+1} = \boldsymbol{p}^k + Q_B^{-1}(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). \end{cases}$$

- No optimization subproblem or system of linear equations over fine mesh is required to solve.

Specifying the Inexact Uzawa Algorithmic Framework

- With the choice of $Q_A = \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix}$ and $Q_B = \tau G M^{-1} G^\top \succeq P_B \succ S$, we specify the inexact Uzawa algorithmic framework as

$$
\begin{cases}
\boldsymbol{u}^{k+1} = P_{\boldsymbol{U}_{ad}}(W^{-1}((W - M)\boldsymbol{u}^k - \frac{1}{\alpha} M^\top \boldsymbol{p}^k)), \\
\boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d), \\
\boldsymbol{p}^{k+1} = \boldsymbol{p}^k + Q_B^{-1}(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}).
\end{cases}
$$

- No optimization subproblem or system of linear equations over fine mesh is required to solve.
- At each iteration, only compute the projection onto a simple admissible set, a few AMG V-cycles, and some matrix-vector multiplications.

Numerical Experiments

Setting-ups:

- Stopping criterion:

$$\max\{p_s, d_s\} \leq tol,$$

  where the primal residual p_s and dual residual d_s are defined as:

$$p_s = (\|u^k - u^{k-1}\|^2 + \|y^k - y^{k-1}\|^2)^{\frac{1}{2}}, \quad \text{and} \quad d_s = \|p^k - p^{k-1}\|.$$

- Initial values: $u = 0, y = 0$ and $p = 0$.
- The AMG V-circle is implemented based on the iFEM package with a Jacobian smoother.

Numerical Example

$$\min_{y\in H_0^1(\Omega),u\in U_{ad}} \quad J(y,u)=\frac{1}{2}\|y-y_d\|^2+\frac{\alpha}{2}\|u\|^2$$

$$\text{s.t.} \qquad \begin{cases} -\Delta y = u & \text{in} \quad \Omega, \\ y = 0 & \text{on} \quad \Gamma. \end{cases}$$

- The domain $\Omega=(0,1)\times(0,1)$, the constraint is $0.3\leq u(x)\leq 1$ and $\alpha=10^{-4}$;
- The target functional $y_d=4\pi^2\alpha\sin(\pi x)\sin(\pi y)+y_r$, where y_r is the solution of

$$\begin{cases} -\Delta y_r = r & \text{in} \quad \Omega, \\ y_r = 0 & \text{on} \quad \Gamma, \end{cases}$$

  with $r=\min\left(1,\max\left(0.3, 2\sin(\pi x)\sin(\pi y)\right)\right)$;
- The exact solution is $u^*=r$.

Numerical Example – Cont'd

Table: Numerical results with $tol = 10^{-9}$.

h	$Iter$	$CPU(s)$	$e_u(h)$	$Order(u)$	$e_y(h)$	$Order(y)$
$1/2^6$	82	0.511972	5.3773×10^{-5}	1.9937	9.6936×10^{-7}	2.0076
$1/2^7$	82	1.242614	1.3438×10^{-5}	2.0005	2.4253×10^{-7}	1.9987
$1/2^8$	82	3.433469	3.3609×10^{-6}	1.9994	6.6031×10^{-8}	2.0000
$1/2^9$	82	16.899456	8.4238×10^{-7}	1.9963	1.5157×10^{-8}	2.0000
$1/2^{10}$	82	74.143923	2.1271×10^{-7}	1.9856	3.7853×10^{-9}	2.0015

- Converging to a rather high-accuracy solution very fast.
- The convergence is independent of the mesh size h.
- The convergence order of $e_u(h)$ is approximately $O(h^2)$ and $e_y(h)$ is $O(h^2)$ which validate the theoretical results $e_u(h) = o(h)$ (see [E. Casas 2007]) and $e_y(h) = O(h^2)$ (see [R.S. Falk 1973]).

Numerical Example – Cont'd

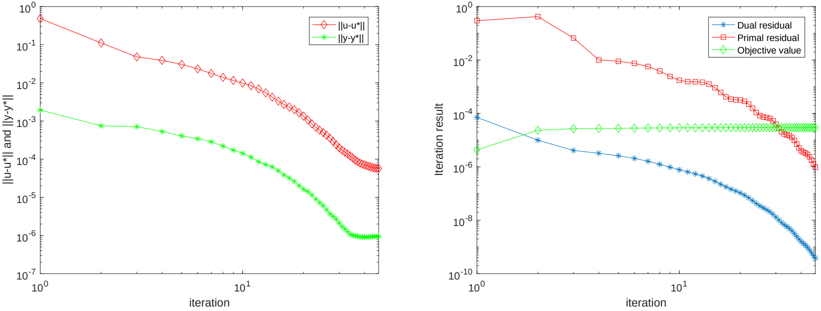

Figure: Iteration error $\|u - u^*\|$ and $\|y - y^*\|$ (left), primal residual p_s, dual residual d_s and objective function value (right).

- Linear convergence rate is seen.

Numerical Example – Cont'd

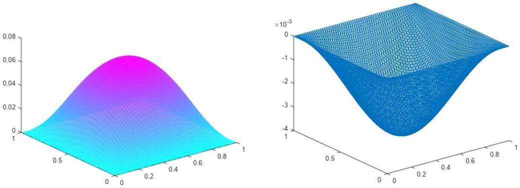

Figure: Numerical solution y and error $y - y_d$ with $h = 1/64$.

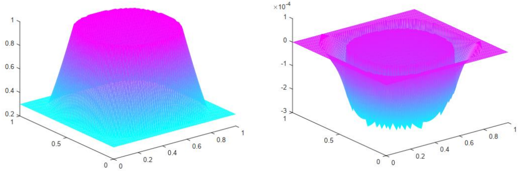

Figure: Numerical solution u and error $u - u^*$ with $h = 1/64$.

Numerical Example – Cont'd

- Comparison with the semismooth Newton method in [M. Porcelli, V. Simoncini and M. Tani 2015].

Numerical Example – Cont'd

- Comparison with the semismooth Newton method in [M. Porcelli, V. Simoncini and M. Tani 2015].

Table: Numerical results of semismooth Newton methods (SSN)

Algorithm	h	No. of Newton	Total GMRES	CPU (s)	$\|u - u^*\|_{L^2(\Omega)}$
SSN	2^{-6}	6	55	0.73565	1.4671×10^{-5}
	2^{-7}	6	49	1.7944	3.6631×10^{-6}
	2^{-8}	6	47	6.7322	9.1543×10^{-7}
	2^{-9}	6	44	31.4861	2.2885×10^{-7}
	2^{-10}	6	42	224.9426	5.7213×10^{-8}

Table: Numerical results of the inexact Uzawa method

Algorithm	h	Total iteration No.	CPU (s)	$\|u - u^*\|_{L^2(\Omega)}$
InUzawa	2^{-6}	71	0.4025	1.4669×10^{-5}
	2^{-7}	71	0.8067	3.6603×10^{-6}
	2^{-8}	71	2.6450	9.1274×10^{-7}
	2^{-9}	70	13.1539	2.2626×10^{-7}
	2^{-10}	70	55.051712	5.6967×10^{-8}

Outline

Motivation

Motivation

- In a typical optimal control model, either boundary or internal locally distributed controls are usually used; these controls are called additive controls because they arise in model equations as additive terms.

Motivation

- In a typical optimal control model, either boundary or internal locally distributed controls are usually used; these controls are called additive controls because they arise in model equations as additive terms.
- Additive controls describe the effect of external added sources or forces and they do NOT change the principal intrinsic properties of the system.

Motivation

- In a typical optimal control model, either boundary or internal locally distributed controls are usually used; these controls are called additive controls because they arise in model equations as additive terms.
- Additive controls describe the effect of external added sources or forces and they do NOT change the principal intrinsic properties of the system.
- Additive controls are not suitable to deal with processes whose principal intrinsic properties should be changed by some control actions.

Motivation

- In a typical optimal control model, either boundary or internal locally distributed controls are usually used; these controls are called additive controls because they arise in model equations as additive terms.
- Additive controls describe the effect of external added sources or forces and they do NOT change the principal intrinsic properties of the system.
- Additive controls are not suitable to deal with processes whose principal intrinsic properties should be changed by some control actions.
- Bilinear controls are necessary!

Bilinear Controls

Bilinear Controls

- Bilinear controls, also known as multiplicative controls, enter the model as coefficients of the corresponding PDEs.

Bilinear Controls

- Bilinear controls, also known as multiplicative controls, enter the model as coefficients of the corresponding PDEs.
- Bilinear controls can change some main physical characteristics of the system under investigation.

Bilinear Controls

- Bilinear controls, also known as multiplicative controls, enter the model as coefficients of the corresponding PDEs.
- Bilinear controls can change some main physical characteristics of the system under investigation.
- In the literature, bilinear controls have become an increasingly popular topic and bilinear optimal control problems constrained by various PDEs have been widely studied:
  - elliptic equations [A. Kröner and B. Vexler 2009];
  - convection-diffusion equations [A. Borzì, E. J. Park and M. Vallejos Lass 2016];
  - parabolic equations [A. Y. Khapalov 2003];
  - the Schrödinger equation [K. Ito and K. Kunisch 2007];
  - the Fokker–Planck equation [A. Fleig and R. Guglielmi 2017].

Model

We consider the following bilinear optimal control problem:

$$\begin{cases} \boldsymbol{u} \in \mathcal{U}, \\ J(\boldsymbol{u}) \leq J(\boldsymbol{v}), \forall \boldsymbol{v} \in \mathcal{U}, \end{cases} \tag{BCP}$$

with the objective functional J defined by

$$J(\boldsymbol{v}) = \frac{1}{2}\iint_Q |\boldsymbol{v}|^2 dxdt + \frac{\alpha_1}{2}\iint_Q |y - y_d|^2 dxdt + \frac{\alpha_2}{2}\int_\Omega |y(T) - y_T|^2 dx,$$

and $y = y(t; \boldsymbol{v})$ the solution of the following advection-reaction-diffusion equation

$$\frac{\partial y}{\partial t} - \nu \nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \text{ in } Q, \quad y = g \text{ on } \Sigma, \quad y(0) = \phi.$$

- Ω is a bounded domain of $\mathbb{R}^d$ with $d \geq 1$ and Γ is its boundary;
- $Q = \Omega \times (0, T)$ and $\Sigma = \Gamma \times (0, T)$ with $0 < T < +\infty$;
- $y_d \in L^2(Q)$, $y_T \in L^2(\Omega)$, $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\alpha_1 + \alpha_2 > 0$;
- $\nu > 0$ and a_0 are constants; $f \in L^2(Q)$, $g \in L^2(0, T; H^{1/2}(\Gamma))$, $\phi \in L^2(\Omega)$;
- the set $\mathcal{U}$ of the admissible controls is defined by

$$\mathcal{U} := \{\boldsymbol{v} | \boldsymbol{v} \in [L^2(Q)]^d, \nabla \cdot \boldsymbol{v} = 0\}.$$

Model–Cont'd

Model–Cont'd

- The (BCP) problem was proposed in 1990s by Late Professor J. L. Lions (1928-2001).

Model–Cont'd

- The (BCP) problem was proposed in 1990s by Late Professor J. L. Lions (1928-2001).
- This model captures important applications in various areas, such as
  - bioremediation [N. Handagama and S. Lenhart 1998];
  - environmental remediation process [S. Lenhart 1995];
  - mixing enhancement of different fluids [W. Liu 2008].

Model–Cont'd

- The (BCP) problem was proposed in 1990s by Late Professor J. L. Lions (1928-2001).
- This model captures important applications in various areas, such as
  - bioremediation [N. Handagama and S. Lenhart 1998];
  - environmental remediation process [S. Lenhart 1995];
  - mixing enhancement of different fluids [W. Liu 2008].
- To the best of our knowledge, no work has been done yet to develop efficient numerical methods for solving such kind of bilinear optimal control problems.

Some Theoretical Results

Some Theoretical Results

- There exists at least one optimal control $\boldsymbol{u} \in \mathcal{U}$ for (BCP).

Some Theoretical Results

- There exists at least one optimal control $\boldsymbol{u} \in \mathcal{U}$ for (BCP).
- Let $DJ(\boldsymbol{v})$ be the first-order differential of J at $\boldsymbol{v}$ and $\boldsymbol{u}$ an optimal control of (BCP). The first-order optimality condition of (BCP) reads

$$DJ(\boldsymbol{u}) = 0.$$

Some Theoretical Results

- There exists at least one optimal control $\boldsymbol{u} \in \mathcal{U}$ for (BCP).
- Let $DJ(\boldsymbol{v})$ be the first-order differential of J at $\boldsymbol{v}$ and $\boldsymbol{u}$ an optimal control of (BCP). The first-order optimality condition of (BCP) reads

$$DJ(\boldsymbol{u}) = 0.$$

- By some sophisticated manipulations, we have that

$$\begin{cases} DJ(\boldsymbol{v}) \in \mathcal{U}, \\ \iint_Q DJ(\boldsymbol{v}) \cdot \boldsymbol{z} dxdt = \iint_Q (\boldsymbol{v} - p\nabla y) \cdot \boldsymbol{z} dxdt, \forall \boldsymbol{z} \in \mathcal{U}, \end{cases}$$

where y and p are obtained from $\boldsymbol{v}$ via the following two PDEs:

$$\frac{\partial y}{\partial t} - \nu \nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \text{ in } Q, \quad y = g \text{ on } \Sigma, \quad y(0) = \phi,$$

and

$$-\frac{\partial p}{\partial t} - \nu \nabla^2 p - \boldsymbol{u} \cdot \nabla p + a_0 p = \alpha_1 (y - y_d) \text{ in } Q,\ p = 0 \text{ on } \Sigma,\ p(T) = \alpha_2 (y(T) - y_T).$$

A Generic Conjugate Gradient Method

(a) Given $\boldsymbol{u}^0 \in \mathcal{U}$.

(b) Compute $\boldsymbol{g}^0 = DJ(\boldsymbol{u}^0)$. If $DJ(\boldsymbol{u}^0) = 0$, take $\boldsymbol{u} = \boldsymbol{u}^0$; otherwise set $\boldsymbol{w}^0 = \boldsymbol{g}^0$.

For $k \geq 0$, $\boldsymbol{u}^k$, $\boldsymbol{g}^k$ and $\boldsymbol{w}^k$ being known, with the last two different from 0, one computes $\boldsymbol{u}^{k+1}$, $\boldsymbol{g}^{k+1}$ and if necessary $\boldsymbol{w}^{k+1}$ as follows:

(c) Compute the stepsize ρ_k by solving the following optimization problem

$$\begin{cases} \rho_k \in \mathbb{R}, \\ J(\boldsymbol{u}^k - \rho_k \boldsymbol{w}^k) \leq J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k), \forall \rho \in \mathbb{R}. \end{cases}$$

(d) Update $\boldsymbol{u}^{k+1}$ and $\boldsymbol{g}^{k+1}$, respectively, by

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \rho_k \boldsymbol{w}^k, \text{ and } \boldsymbol{g}^{k+1} = DJ(\boldsymbol{u}^{k+1}).$$

If $DJ(\boldsymbol{u}^{k+1}) = 0$, take $\boldsymbol{u} = \boldsymbol{u}^{k+1}$; otherwise,

(e) Compute

$$\beta_k = \frac{\iint_Q |\boldsymbol{g}^{k+1}|^2 dxdt}{\iint_Q |\boldsymbol{g}^k|^2 dxdt},$$

and then update

$$\boldsymbol{w}^{k+1} = \boldsymbol{g}^{k+1} + \beta_k \boldsymbol{w}^k.$$

Do $k + 1 \to k$ and return to (**c**).

Difficulties to Implement the CG Method

Difficulties to Implement the CG Method

- Computation of the gradient $DJ(\boldsymbol{v})$. Recall that

$$\begin{cases} DJ(\boldsymbol{v}) \in \mathcal{U}, \\ \iint_Q DJ(\boldsymbol{v}) \cdot \boldsymbol{z} dxdt = \iint_Q (\boldsymbol{v} - p\nabla y) \cdot \boldsymbol{z} dxdt, \forall \boldsymbol{z} \in \mathcal{U}. \end{cases}$$

where y and p are obtained from $\boldsymbol{v}$ by solving

$$\frac{\partial y}{\partial t} - \nu\nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \text{ in } Q, \quad y = g \text{ on } \Sigma, \quad y(0) = \phi,$$

and

$$-\frac{\partial p}{\partial t} - \nu\nabla^2 p - \boldsymbol{u} \cdot \nabla p + a_0 p = \alpha_1(y - y_d) \text{ in } Q,\ p = 0 \text{ on } \Sigma,\ p(T) = \alpha_2(y(T) - y_T).$$

Difficulties to Implement the CG Method

- Computation of the gradient $DJ(\boldsymbol{v})$. Recall that

$$\begin{cases} DJ(\boldsymbol{v}) \in \mathcal{U}, \\ \iint_Q DJ(\boldsymbol{v}) \cdot \boldsymbol{z} dxdt = \iint_Q (\boldsymbol{v} - p\nabla y) \cdot \boldsymbol{z} dxdt, \forall \boldsymbol{z} \in \mathcal{U}. \end{cases}$$

where y and p are obtained from $\boldsymbol{v}$ by solving

$$\frac{\partial y}{\partial t} - \nu\nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \text{ in } Q, \quad y = g \text{ on } \Sigma, \quad y(0) = \phi,$$

and

$$-\frac{\partial p}{\partial t} - \nu\nabla^2 p - \boldsymbol{u} \cdot \nabla p + a_0 p = \alpha_1(y - y_d) \text{ in } Q,\ p = 0 \text{ on } \Sigma,\ p(T) = \alpha_2(y(T) - y_T).$$

- Computation of the stepsize ρ_k.

$$\begin{cases} \rho_k \in \mathbb{R}, \\ J(\boldsymbol{u}^k - \rho_k \boldsymbol{w}^k) \leq J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k), \forall \rho \in \mathbb{R}. \end{cases}$$

Gradient Computation

- First of all, it follows from [J. L. Lions 1971] that the computation of $DJ(v)$ can be reformulated as

$$\begin{cases} DJ(\boldsymbol{v})(t) \in \mathbb{S}, \text{ for a.e. } t \in (0,T), \\ \int_\Omega DJ(\boldsymbol{v})(t) \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}(t) - p(t)\nabla y(t)) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}, \end{cases}$$

where

$$\mathbb{S} = \{\boldsymbol{z} | \boldsymbol{z} \in [L^2(\Omega)]^d, \nabla \cdot \boldsymbol{z} = 0\}.$$

Gradient Computation

- First of all, it follows from [J. L. Lions 1971] that the computation of $DJ(v)$ can be reformulated as

$$\begin{cases} DJ(\boldsymbol{v})(t) \in \mathbb{S}, \text{ for a.e. } t \in (0, T), \\ \int_\Omega DJ(\boldsymbol{v})(t) \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}(t) - p(t)\nabla y(t)) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}, \end{cases}$$

where

$$\mathbb{S} = \{\boldsymbol{z} | \boldsymbol{z} \in [L^2(\Omega)]^d, \nabla \cdot \boldsymbol{z} = 0\}.$$

- The above problem is a particular case of

$$\begin{cases} \boldsymbol{g} \in \mathbb{S}, \\ \int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}, \end{cases}$$

with $\boldsymbol{f}$ given in $[L^2(\Omega)]^d$.

Gradient Computation–Cont'd

- Introducing a Lagrange multiplier $\lambda \in H_0^1(\Omega)$ associated with the constraint $\nabla \cdot \boldsymbol{z} = 0$ and then we have the following saddle point problem

$$\begin{cases} (\boldsymbol{g}, \lambda) \in [L^2(\Omega)]^d \times H_0^1(\Omega), \\ \int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx + \int_\Omega \lambda \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \\ \int_\Omega \nabla \cdot \boldsymbol{g} q dx = 0, \forall q \in H_0^1(\Omega). \end{cases}$$

Gradient Computation–Cont'd

- Introducing a Lagrange multiplier $\lambda \in H_0^1(\Omega)$ associated with the constraint $\nabla \cdot \boldsymbol{z} = 0$ and then we have the following saddle point problem

$$\begin{cases} (\boldsymbol{g}, \lambda) \in [L^2(\Omega)]^d \times H_0^1(\Omega), \\ \displaystyle\int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx + \int_\Omega \lambda \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \\ \displaystyle\int_\Omega \nabla \cdot \boldsymbol{g} q dx = 0, \forall q \in H_0^1(\Omega). \end{cases}$$

- This is actually a Stokes type problem.

Gradient Computation–Cont'd

- Introducing a Lagrange multiplier $\lambda \in H_0^1(\Omega)$ associated with the constraint $\nabla \cdot \boldsymbol{z} = 0$ and then we have the following saddle point problem

$$\begin{cases} (\boldsymbol{g}, \lambda) \in [L^2(\Omega)]^d \times H_0^1(\Omega), \\ \int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx + \int_\Omega \lambda \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \\ \int_\Omega \nabla \cdot \boldsymbol{g} q dx = 0, \forall q \in H_0^1(\Omega). \end{cases}$$

- This is actually a Stokes type problem.
- We advocate employing a preconditioned CG method inspired from [R. Glowinski 2003] — leading to a nested CG method for solving (BCP).

Stepsize Computation

- For a given $\boldsymbol{w}^k \in \mathcal{U}$, we replace the state $y = S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ in $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ by

$$S(\boldsymbol{u}^k) - \rho S'(\boldsymbol{u}^k)\boldsymbol{w}^k,$$

which is indeed the linearization of the mapping $\rho \mapsto S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ at $\rho = 0$.

Stepsize Computation

- For a given $\boldsymbol{w}^k \in \mathcal{U}$, we replace the state $y = S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ in $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ by
$$S(\boldsymbol{u}^k) - \rho S'(\boldsymbol{u}^k)\boldsymbol{w}^k,$$
which is indeed the linearization of the mapping $\rho \mapsto S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ at $\rho = 0$.
- We thus obtain the following quadratic approximation of $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$:
$$\begin{aligned} Q_k(\rho) :=& \frac{1}{2} \iint_Q |\boldsymbol{u}^k - \rho \boldsymbol{w}^k|^2 dxdt \\ &+ \frac{\alpha_1}{2} \iint_Q |y^k - \rho z^k - y_d|^2 dxdt + \frac{\alpha_2}{2} \int_\Omega |y^k(T) - \rho z^k(T) - y_T|^2 dx, \end{aligned}$$
where $y^k = S(\boldsymbol{u}^k)$ is the solution of the state equation associated with $\boldsymbol{u}^k$, and $z^k := S'(\boldsymbol{u}^k)\boldsymbol{w}^k$ satisfies the following linear parabolic problem
$$\frac{\partial z^k}{\partial t} - \nu \nabla^2 z^k + \boldsymbol{w}^k \cdot \nabla y^k + \boldsymbol{u}^k \cdot \nabla z^k + a_0 z^k = 0 \text{ in } Q, \quad z^k = 0 \text{ on } \Sigma, \quad z^k(0) = 0.$$

Stepsize Computation

- For a given $\boldsymbol{w}^k \in \mathcal{U}$, we replace the state $y = S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ in $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ by

$$S(\boldsymbol{u}^k) - \rho S'(\boldsymbol{u}^k)\boldsymbol{w}^k,$$

which is indeed the linearization of the mapping $\rho \mapsto S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ at $\rho = 0$.

- We thus obtain the following quadratic approximation of $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$:

$$\begin{aligned} Q_k(\rho) :=& \frac{1}{2} \iint_Q |\boldsymbol{u}^k - \rho \boldsymbol{w}^k|^2 dxdt \\ &+ \frac{\alpha_1}{2} \iint_Q |y^k - \rho z^k - y_d|^2 dxdt + \frac{\alpha_2}{2} \int_\Omega |y^k(T) - \rho z^k(T) - y_T|^2 dx, \end{aligned}$$

where $y^k = S(\boldsymbol{u}^k)$ is the solution of the state equation associated with $\boldsymbol{u}^k$, and $z^k := S'(\boldsymbol{u}^k)\boldsymbol{w}^k$ satisfies the following linear parabolic problem

$$\frac{\partial z^k}{\partial t} - \nu \nabla^2 z^k + \boldsymbol{w}^k \cdot \nabla y^k + \boldsymbol{u}^k \cdot \nabla z^k + a_0 z^k = 0 \text{ in } Q, \quad z^k = 0 \text{ on } \Sigma, \quad z^k(0) = 0.$$

- It is easy to show that the equation $Q_k'(\rho) = 0$ admits a unique solution

$$\hat{\rho}_k = \frac{\iint_Q \boldsymbol{g}^k \cdot \boldsymbol{w}^k dxdt}{\iint_Q |\boldsymbol{w}^k|^2 dxdt + \alpha_1 \iint_Q |z^k|^2 dxdt + \alpha_2 \int_\Omega |z^k(T)|^2 dx}.$$

Numerical Discretization

Numerical Discretization

- Time discretization
  - We employ a semi-implicit finite difference method, namely, we use explicit advection and reaction terms and treat the diffusion term implicitly.
  - Only a simple linear elliptic equation is required to be solved at each time step.

Numerical Discretization

- Time discretization
  - We employ a semi-implicit finite difference method, namely, we use explicit advection and reaction terms and treat the diffusion term implicitly.
  - Only a simple linear elliptic equation is required to be solved at each time step.
- Space discretization
  - Direct applications of standard finite element methods do not work.
  - We employ the Bercovier-Pironneau finite element pair (a.k.a. P1-P1 iso P2 finite element) to approximate the control $\boldsymbol{v}$ and the Lagrange multiplier λ.

Numerical Discretization

- Time discretization
  - We employ a semi-implicit finite difference method, namely, we use explicit advection and reaction terms and treat the diffusion term implicitly.
  - Only a simple linear elliptic equation is required to be solved at each time step.
- Space discretization
  - Direct applications of standard finite element methods do not work.
  - We employ the Bercovier-Pironneau finite element pair (a.k.a. P1-P1 iso P2 finite element) to approximate the control $\boldsymbol{v}$ and the Lagrange multiplier λ.
- With the above discretization schemes, we can obtain the fully discrete version of (BCP) and derive the discrete analogue of our proposed nested CG method.

Numerical Experiments

Setting-ups:

- $\Omega = (0,1)^2$ and $T = 1$.
- $\alpha_2 = 0$, $\nu = 1$ and $a_0 = 1$.
- $$\begin{aligned} y &= e^t(-3\sin(2\pi x_1)\sin(\pi x_2) + 1.5\sin(\pi x_1)\sin(2\pi x_2)), \\ p &= (T-t)\sin\pi x_1 \sin\pi x_2, \text{ and } \boldsymbol{u} = P_{\mathcal{U}}(p\nabla y), \\ f &= \frac{\partial y}{\partial t} - \nabla^2 y + \boldsymbol{u}\cdot\nabla y + y, \\ \phi &= -3\sin(2\pi x_1)\sin(\pi x_2) + 1.5\sin(\pi x_1)\sin(2\pi x_2), \\ y_d &= y - \frac{1}{\alpha_1}\left(-\frac{\partial p}{\partial t} - \nabla^2 p - \boldsymbol{u}\cdot\nabla p + p\right), \quad g = 0. \end{aligned}$$
- $\boldsymbol{u}$ is an optimal control.
- Note $\boldsymbol{u} = P_{\mathcal{U}}(p\nabla y)$ has no analytical solution. We solve $\boldsymbol{u} = P_{\mathcal{U}}(p\nabla y)$ by the preconditioned CG algorithm with $h = \frac{1}{2^9}$ and $\Delta t = \frac{1}{2^{10}}$, and use the resulting control $\boldsymbol{u}$ as a reference solution.

Numerical Experiments–Cont'd

- The stopping criteria of the outer CG algorithm and the inner preconditioned CG algorithm are respectively set as

$$\frac{\Delta t\sum_{n=1}^{N}\int_\Omega |\boldsymbol{g}_n^{k+1}|^2 dx}{\Delta t\sum_{n=1}^{N}\int_\Omega |\boldsymbol{g}_n^{0}|^2 dx} \leq 5\times 10^{-8}, \text{ and } \frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\max\{1,\int_\Omega |\nabla r^{0}|^2 dx\}} \leq 10^{-8}.$$

- Initial values: $\boldsymbol{u}^0=(0,0)^\top$ and $\lambda^0=0$.
- We denote by $\boldsymbol{u}_h^{\Delta t}$ and $y_h^{\Delta t}$ the computed control and state, respectively.

Numerical Experiments–Cont'd

Table: Results of the nested CG algorithm with different h and Δt.

Mesh sizes	$Iter_{CG}$	Max_{PCG}	$\\|\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}\\|_{L^2(Q)}$	$\\|y_h^{\Delta t} - y\\|_{L^2(Q)}$	$\frac{\\|y_h^{\Delta t} - y_d\\|_{L^2(Q)}}{\\|y_d\\|_{L^2(Q)}}$
$h = \frac{1}{2^6}, \Delta t = \frac{1}{2^7}$	443	9	3.7450×10^{-3}	9.7930×10^{-5}	1.0906×10^{-6}
$h = \frac{1}{2^7}, \Delta t = \frac{1}{2^8}$	410	9	1.8990×10^{-3}	1.7423×10^{-5}	3.3863×10^{-7}
$h = \frac{1}{2^8}, \Delta t = \frac{1}{2^9}$	405	8	1.1223×10^{-3}	4.4003×10^{-6}	1.0378×10^{-7}

- The outer CG algorithm converges fast and is robust with respect to different mesh sizes.
- The inner preconditioned CG algorithm converges within 10 iterations for all cases and thus is efficient for computing the gradient.
- The target function y_d has been reached within a good accuracy. Similar comments hold for the approximation of the optimal control $\boldsymbol{u}$ and of the state y.

Numerical Experiments–Cont'd

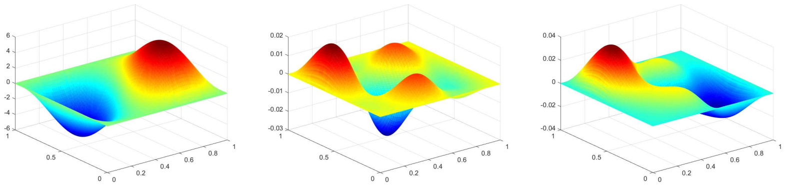

Figure: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ (from left to right) at $t = 0.5$.

Numerical Experiments–Cont'd

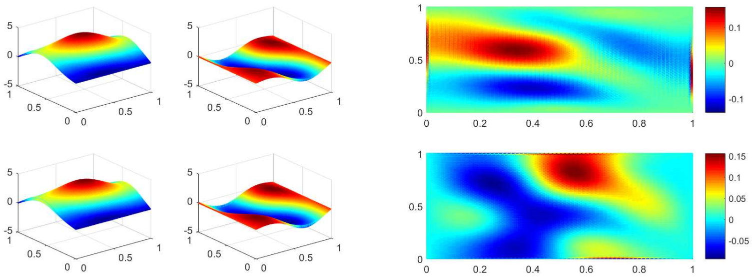

Figure: Computed control $\boldsymbol{u}_h^{\Delta t}$ and exact control $\boldsymbol{u}$ (left, from top to bottom) and the error $\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}$ (right) with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ at $t = 0.5$.

Outline

Conclusions

Conclusions

- It is generally nontrivial to find efficient numerical solvers for optimal control problems with PDE constraints.

Conclusions

- It is generally nontrivial to find efficient numerical solvers for optimal control problems with PDE constraints.
- Specific structures and properties of the model should be considered deliberately.

Conclusions

- It is generally nontrivial to find efficient numerical solvers for optimal control problems with PDE constraints.
- Specific structures and properties of the model should be considered deliberately.
- Operator splitting strategies are promising for various optimal control problems.

Conclusions

- It is generally nontrivial to find efficient numerical solvers for optimal control problems with PDE constraints.
- Specific structures and properties of the model should be considered deliberately.
- Operator splitting strategies are promising for various optimal control problems.
- In particular, numerical linear algebra techniques such as preconditioning, matrix factorization, multigrid methods and Krylov subspace methods are very important.