SAND2015-11034R

A TIME-PARALLEL METHOD FOR THE SOLUTION OF PDE-CONSTRAINED OPTIMIZATION PROBLEMS

M. HAJGHASSEM*, E. C. CYR†, AND D. RIDZAL‡

Abstract. We study a time-parallel approach to solving quadratic optimization problems with linear time-dependent partial differential equation (PDE) constraints. These problems arise in formulations of optimal control, optimal design and inverse problems that are governed by parabolic PDE models. They may also arise as subproblems in algorithms for the solution of optimization problems with nonlinear time-dependent PDE constraints, e.g., in sequential quadratic programming methods. We apply a piecewise linear finite element discretization in space to the PDE constraint, followed by the Crank-Nicolson discretization in time. The objective function is discretized using finite elements in space and the trapezoidal rule in time. At this point in the discretization, auxiliary state variables are introduced at each discrete time interval, with the goal to enable: (i) a decoupling in time; and (ii) a fixed-point iteration to recover the solution of the discrete optimality system. The fixed-point iterative schemes can be used either as preconditioners for Krylov subspace methods or as smoothers for multigrid (in time) schemes. We present promising numerical results for both use cases.

1. Introduction. Optimal control problems governed by time-dependent partial differential equations (PDEs) lead to large-scale optimization problems. The cost of numerically solving these problems is proportional to the size of the discretized time domain. In most numerical approaches the time discretization serializes the solution process, and introduces a bottleneck that cannot be overcome by additional parallelization in space. To speed up the solution process for linear-quadratic optimal control problems governed by parabolic PDEs, we study a time-domain decomposition approach with the potential to introduce time parallelism into the optimization algorithm. Parallelization in time approximates parts of the solution later in time simultaneously with the parts of the solution earlier in time.

In Gander's review article on parallel time integration [4], time-parallel methods are classified into four groups: methods based on multiple shooting; methods based on domain decomposition and waveform relaxation; space-time multigrid methods; and direct time-parallel methods. Our approach is based on domain decomposition. However, we note from the onset that our goal is the iterative solution of optimality systems that comprise discrete PDE systems, and not the solution of discrete PDE systems on their own as in [4] and the references cited therein. This introduces opportunities for time parallelization that have not been explored in the literature. Our work is closely related to the work of Heinkenschloss [5] and Comas [3], who consider time-domain decomposition (and multiple shooting) methods for optimal control problems to reduce storage demands imposed by the discretization of the state variables. In contrast, our work focuses on preconditioners that are appropriate for time parallelization. In principle, it is possible to balance storage reduction and time parallelization in a single scheme, which we plan to study in the future.

To streamline the presentation, we consider the Neumann boundary control of the one-dimensional linear heat equation. For the spatial discretization of the heat equation we use piecewise linear finite elements. For the time discretization we use

*University of Maryland Baltimore County, mona4@umbc.edu

†Sandia National Laboratories, Computational Mathematics, eccyr@sandia.gov

‡Sandia National Laboratories, Optimization and Uncertainty Quantification, dridzal@sandia.gov.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

the Crank-Nicolson method. The objective function is discretized using finite elements in space and the trapezoidal rule in time. At this point of the discretization process, we introduce auxiliary state variables $\bar{u}_i$ for each time interval, corresponding to the original discrete state variables u_i, and impose the time-continuity constraint equations $\bar{u}_i = u_i$. Additionally, the auxiliary variables $\bar{u}_i$ are incorporated into the objective function. The first-order necessary and sufficient optimality conditions for the discretized problem form a linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x}$ is a vector composed of the state variables, auxiliary state variables, control variables, the Lagrange multipliers associated with the state equation and the Lagrange multipliers associated with the continuity equations. With a suitable ordering of the necessary and sufficient optimality conditions, we arrive at a block tridiagonal system of optimality conditions, presented in Section 2. The block tridiagonal structure motivates fixed-point iterations based on matrix splittings, which are discussed in Section 3. For instance, the optimality system can be solved using the GMRES method and time-parallel fixed-point preconditioners, such as the Jacobi scheme or the red-black Gauss-Seidel scheme. Fixed-point iterations can also be used as smoothers in the development of new multigrid in time techniques. In Section 4 we present promising numerical results for both use cases. We demonstrate the potential effectiveness of the Jacobi scheme as a time-parallel preconditioner for linear systems arising in, e.g., inexact sequential quadratic programming (SQP) methods. We also demonstrate the smoothing properties of Jacobi and Gauss-Seidel iterations, with a future goal of incorporating them in a multigrid in time scheme.

2. Neumann control of the one-dimensional heat equation. We consider the optimization problem

$$\min \frac{1}{2}\int_0^T z^2(t)dt + \frac{\alpha}{2}\int_0^T \int_0^1 (u(t,x) - s(t,x))^2 dxdt \tag{2.1}$$

governed by the one-dimensional linear heat equation

$$\frac{\partial}{\partial t}u(t,x) - \frac{\partial^2}{\partial x^2}u(t,x) = f(t,x) \qquad t \in (0,T),\ x \in (0,1), \tag{2.2a}$$

$$\frac{\partial}{\partial x}u(t,0) = z(t), \qquad t \in (0,T), \tag{2.2b}$$

$$\frac{\partial}{\partial x}u(t,1) = r(t), \qquad t \in (0,T), \tag{2.2c}$$

$$u(0,x) = u_0(x), \qquad x \in (0,1). \tag{2.2d}$$

The functions u and z are called state and control, respectively. The above problem is posed in Hilbert space, as described in [5]. We assume that (2.2) admits a unique solution u for every control z and that the above optimal control problem has a solution, see [3, 5].

We discretize the state equation in space by applying piecewise linear finite elements with N_x spatial intervals on the $[0,1]$ interval. We arrive at the semi-discrete problem

$$\min \frac{1}{2}\int_0^T z^2(t)dt + \frac{\alpha}{2}\int_0^T \|u(t) - s(t)\|^2 dt \tag{2.3}$$

subject to

$$M\frac{\partial}{\partial t}u(t) + Su(t) = f(t) + Wz(t) \qquad t \in (0,T), \tag{2.4a}$$

$$u(0) = u_0, \tag{2.4b}$$

where M and S are the mass and stiffness matrices, respectively, of size $(N_x+1)\times(N_x+1)$, $f(t)$ is an (N_x+1)-vector that includes the Neumann boundary term $r(t)$, and W is an $(N_x+1)\times 1$ operator that applies the control as the Neumann condition at $x=0$. We let $\|\cdot\|$ denote a discrete norm over space, defined in this case as $\|u\|^2 = u^T M u$. To avoid vector notation we introduce a hierarchical notation for functions, based on the space or space-time discretization. We write $w(x,t)$ for a continuous function, $w(t)$ for its space discretization, and w_i for its space-time discretization, where i is a time index.

We apply the θ-method to discretize in time the semi-discrete constraint equation (2.4); in numerical experiments, we will use $\theta = 1/2$, i.e., the Crank-Nicolson method. We apply the trapezoidal rule to approximate the time integrals in the objective function (2.3). The time domain is divided into N intervals $[t_i, t_{i+1}]$, $i = 0, 1, ..., N-1$. At the time domain interfaces, auxiliary variables $\bar{u}_i$, $i = 1, ..., N-1$, are introduced to enforce time continuity of the state in the time-domain decomposition process. Additionally, these variables are incorporated in the objective function, and used simultaneously with the state variables u_i. The control variables $\bar{z}_i$ are scalar quantities, defined in each subinterval $[t_i, t_{i+1}]$. The fully discretized optimal control problem can be written as

$$\begin{aligned}\min J(u,\bar{u},\bar{z}) = \frac{\Delta t}{2}\sum_{i=0}^{N-1}(\bar{z}_i)^2 \\ + \frac{\alpha\Delta t}{4}\sum_{i=0}^{N-1}(\|\bar{u}_i - s_i\|^2 + \|u_{i+1} - s_{i+1}\|^2)\end{aligned} \tag{2.5}$$

subject to

$$\begin{aligned} M\frac{u_{i+1}-\bar{u}_i}{\Delta t} =& -S(\theta u_{i+1} + (1-\theta)\bar{u}_i) + \theta(f_{i+1} + W\bar{z}_i) + (1-\theta)(f_i + W\bar{z}_i) \\ \bar{u}_i =& u_i \qquad i = 0, 1, \ldots, N-1, \end{aligned} \tag{2.6}$$

where Δt is the time step. By rearranging the constraint equation we obtain:

$$\begin{aligned}(M + \Delta t\theta S)u_{i+1} &= (M - \Delta t(1-\theta)S)\bar{u}_i + \Delta t(W\bar{z}_i + \theta f_{i+1} + (1-\theta)f_i) \\ \bar{u}_i &= u_i \qquad i = 0, 1, \ldots, N-1.\end{aligned} \tag{2.7}$$

Note that the minimizer of the time-domain decomposition formulation is also a minimizer of the discretized continuous time formulation. To see this let u_i^*, $\bar{u}_i^*$ and $\bar{z}_i^*$ satisfy Eq. (2.5) subject to Eq. (2.7). Then eliminating the auxiliary variable $\bar{u}_i^*$ using the continuity condition, the state constraint implies

$$(M + \Delta t\theta S)u_{i+1} = (M - \Delta t(1-\theta)S)u_i + \Delta t\bar{z}_i + \Delta t(\theta f_{i+1} + (1-\theta)f_i). \tag{2.8}$$

This is an approximation of Eq. (2.4) using the θ-method. Similarly, using the continuity condition to eliminate $\bar{u}_i^*$ from Eq. (2.5) gives the trapezoidal-rule approximation of the objective function from Eq. (2.3).

The motivation for introducing the continuity condition is twofold. First, including it in the discretization is more general than enforcing it by construction. Given an implementation of the discontinuous case it is easy to construct the continuous case. Second, explicit treatment of the continuity condition as a state equation exposes the condition to the optimization algorithm. This enables a lifted Newton method with respect to the time-continuous problem [1], which may (in future studies) provide additional algorithmic advantages.

The Lagrangian function for the above discretized problem is given by

$$\begin{aligned} L(u,\bar{u},\bar{z},\lambda,\bar{\lambda}) = \frac{\Delta t}{2}\sum_{i=0}^{N-1}(\bar{z}_i)^2 \\ + \frac{\alpha\Delta t}{4}\sum_{i=0}^{N-1}\left(\|\bar{u}_i - s_i\|^2 + \|u_{i+1} - s_{i+1}\|^2\right) \\ + \sum_{i=0}^{N-1}(\lambda_{i+1}, K\bar{u}_i + \Delta t W\bar{z}_i + \Delta t(\theta f_{i+1} + (1-\theta)f_i) - Cu_{i+1}) \\ + \sum_{i=0}^{N-1}\left(\bar{\lambda}_{i+1}, u_i - \bar{u}_i\right), \end{aligned} \tag{2.9}$$

where $C = M + \Delta t\theta S$ and $K = M - \Delta t(1-\theta)S$. The optimality conditions can be obtained by setting the partial derivatives of L with respect to $\bar{u}_i$, $\bar{z}_i$, λ_i, $\bar{\lambda}_i$, and u_i to zero. The partial derivatives of the Lagrangian read:

$$\begin{aligned} \frac{\partial L}{\partial \bar{u}_i} &= \frac{\alpha\Delta t}{2}M(\bar{u}_i - s_i) + K^T\lambda_{i+1} - \bar{\lambda}_{i+1}, && \text{for } i = 0\ldots N-1; \\ \frac{\partial L}{\partial \bar{z}_i} &= \Delta t(\bar{z}_i + W^T\lambda_{i+1}), && \text{for } i = 0\ldots N-1; \\ \frac{\partial L}{\partial \lambda_i} &= K\bar{u}_{i-1} + \Delta t W\bar{z}_{i-1} + \Delta t\theta f_i + (1-\theta)f_{i-1} - Cu_i, && \text{for } i = 1\ldots N; \\ \frac{\partial L}{\partial \bar{\lambda}_i} &= u_{i-1} - \bar{u}_{i-1}, && \text{for } i = 2\ldots N; \\ \frac{\partial L}{\partial u_i} &= \frac{\alpha\Delta t}{2}M(u_i - s_i) - C^T\lambda_i + \bar{\lambda}_{i+1}, && \text{for } i = 1\ldots N. \end{aligned} \tag{2.10}$$

Notice for brevity we have not been precise with respect to the fringe cases (e.g., $i = 0$, $i = N$).

The Lagrangian defined in Eq. (2.9) uses an identity matrix in space to enforce continuity in time (e.g., enforcing the constraint $\bar{u}_i = u_i$). This implies the adjoint equations correspond to

$$C^T\lambda_i - \bar{\lambda}_{i+1} = \frac{\alpha\Delta t}{2}M(u_i - s_i) \tag{2.11}$$

$$K^T\lambda_{i+1} - \bar{\lambda}_{i+1} = \frac{\alpha\Delta t}{2}M(\bar{u}_i - s_i). \tag{2.12}$$

The equations represent adjoint time continuity conditions. In the first equation the adjoint variables λ_i and $\bar{\lambda}_{i+1}$ have different scaling. This can be optionally rectified by changing the state continuity condition to $C(\bar{u}_i - u_i) = 0$. In the numerical results

section this condition is used for the smoothing study, while the original condition $\bar{u}_i = u_i$ is used for the GMRES study.

The optimality conditions lead to a linear system, $\mathbf{Ax} = \mathbf{b}$. The equations and variables in the linear system can be ordered in a variety of ways. We take adjoint variables to be λ_i and $\bar{\lambda}_i$ instead of λ_{i+1} and $\bar{\lambda}_{i+1}$. We consider the following ordering for the vector $\mathbf{x}$,

$$\mathbf{x}^T = \begin{pmatrix} . & . & . & \bar{u}_i & \bar{z}_i & \lambda_{i+1} & \bar{\lambda}_{i+1} & u_{i+1} & . & . & . \end{pmatrix}. \tag{2.13}$$

This presents the ordering in the interior of the time domain focusing on the time interval $[t_i, t_{i+1}]$. Again for simplicity we ignore the ordering for $i = 0$ and $i = N$. The matrix $\mathbf{A}$ corresponding to the vector $\mathbf{x}$ is of size $N \times N$ blocks, and reads:

$$\mathbf{A} = \begin{pmatrix} D & U & & & \\ L & D & U & & \\ & \ddots & \ddots & \ddots & \\ & & L & D & U \\ & & & L & D \end{pmatrix}, \tag{2.14}$$

where

$$D = \begin{pmatrix} \frac{\alpha_1 \Delta t}{2} M & 0 & K^T & -I & 0 \\ 0 & \Delta t I & \Delta t W^T & 0 & 0 \\ K & \Delta t W & 0 & 0 & -C \\ -I & 0 & 0 & 0 & 0 \\ 0 & 0 & -C^T & 0 & \frac{\alpha_1 \Delta t}{2} M \end{pmatrix} \tag{2.15}$$

and

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{pmatrix}. \tag{2.16}$$

The right-hand side vector $\mathbf{b}$ is:

$$\mathbf{b}^T = \begin{pmatrix} 0 & 0 & -F_1 & 0 & \frac{\alpha_1 \Delta t}{2} s_1 & . & . & . & 0 & 0 & -F_N & 0 & \frac{\alpha_1 \Delta t}{2} s_N \end{pmatrix}. \tag{2.17}$$

3. Time-domain decomposition preconditioners. As seen in Section 2, the system of optimality conditions for the time-dependent optimal control problem leads to a block tridiagonal linear system $\mathbf{Ax} = \mathbf{b}$. This system can be solved using the GMRES method of Saad and Schultz [8], combined with various preconditioners. In this section we discuss the construction of time-domain decomposition preconditioners based on matrix splitting methods, such as Jacobi or Gauss-Seidel.

We will use the 'right-preconditioned' form of $\mathbf{Ax} = \mathbf{b}$,

$$\mathbf{AP}^{-1}\mathbf{y} = \mathbf{b}, \qquad \mathbf{x} = \mathbf{P}^{-1}\mathbf{y}, \tag{3.1}$$

where $\mathbf{P}$ denotes the preconditioning matrix. We use the diag$(\cdot)$ operator to denote the $N \times N$ block diagonal matrix composed of its arguments. Similarly, we introduce

operators subdiag$(\cdot)$ and supdiag$(\cdot)$, which denote $N \times N$ block subdiagonal and superdiagonal matrices composed of the operator arguments. We define the matrices:

$$\mathbf{D} = \text{diag}(D, \ldots, D); \quad \mathbf{L} = \text{subdiag}(L, \ldots, L); \quad \mathbf{U} = \text{supdiag}(U, \ldots, U);$$

and write the matrix $\mathbf{A}$ as their sum, $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$. Moreover, we can define preconditioners by zeroing some of the entries of the matrices $\mathbf{L}$ and $\mathbf{U}$, as follows. First, for a positive integer j, we introduce the matrices:

$$\begin{aligned}\mathbf{L}(j) &= \text{subdiag}(L, \ldots, L, \underset{j}{0}, L, \ldots, L, \underset{2j}{0}, L, \ldots, L, \underset{3j}{0}, \ldots); \quad \text{and} \\ \mathbf{U}(j) &= \text{supdiag}(U, \ldots, U, \underset{j}{0}, U, \ldots, U, \underset{2j}{0}, U, \ldots, U, \underset{3j}{0}, \ldots).\end{aligned}$$

Second, recall that N denotes the total number of time steps, and let N_t denote the number of time subdomains. In $\mathbf{L}(j)$ and $\mathbf{U}(j)$ we zero the entries of $\mathbf{L}$ and $\mathbf{U}$ in locations that are the integer multiples of $j = \lfloor N/N_t \rfloor$. The locations j, $2j$, etc., indicate the boundaries of the time subdomains. In other words, they indicate the breaking of time continuity in the state variables. This leads to the definition of three preconditioners:

$$\begin{aligned}&\mathbf{P}_{\text{JAC}} = \mathbf{L}(j) + \mathbf{D} + \mathbf{U}(j): && \text{Jacobi;} \\ &\mathbf{P}_{\text{FGS}} = \mathbf{L} + \mathbf{D} + \mathbf{U}(j): && \text{forward Gauss-Seidel;} \quad \text{and} \\ &\mathbf{P}_{\text{BGS}} = \mathbf{L}(j) + \mathbf{D} + \mathbf{U}: && \text{backward Gauss-Seidel.}\end{aligned}$$

For example, for the number of time steps $N = 10$ and the number of time sudomains $N_t = 2$, we have that $j = 5$, and the Jacobi preconditioner assumes the following form:

$$\mathbf{P}_{\text{JAC}} = \begin{pmatrix} D & U & & & & & & & & \\ L & D & U & & & & & & & \\ & L & D & U & & & & & & \\ & & L & D & U & & & & & \\ & & & L & D & & & & & \\ & & & & & D & U & & & \\ & & & & & L & D & U & & \\ & & & & & & L & D & U & \\ & & & & & & & L & D & U \\ & & & & & & & & L & D \end{pmatrix}. \tag{3.2}$$

Similarly, the forward Gauss-Seidel preconditioner is given by

$$\mathbf{P}_{\text{FGS}} = \begin{pmatrix} D & U & & & & & & & & \\ L & D & U & & & & & & & \\ & L & D & U & & & & & & \\ & & L & D & U & & & & & \\ & & & L & D & & & & & \\ & & & & \boldsymbol{L} & D & U & & & \\ & & & & & L & D & U & & \\ & & & & & & L & D & U & \\ & & & & & & & L & D & U \\ & & & & & & & & L & D \end{pmatrix}, \tag{3.3}$$

where we have emphasized the "tying" block (in bold) on the subdiagonal, and the backward Gauss-Seidel preconditioner is given by

$$\mathbf{P}_{\text{BGS}} = \begin{pmatrix} D & U & & & & & & & & \\ L & D & U & & & & & & & \\ & L & D & U & & & & & & \\ & & L & D & U & & & & & \\ & & & L & D & \boldsymbol{U} & & & & \\ & & & & & D & U & & & \\ & & & & & L & D & U & & \\ & & & & & & L & D & U & \\ & & & & & & & L & D & U \\ & & & & & & & & L & D \end{pmatrix}, \tag{3.4}$$

where we have emphasized the tying block (in bold) on the superdiagonal. The Jacobi preconditioner parallelizes trivially, as each of the blocks corresponding to the time subdomains can be inverted independently, while the Gauss-Seidel preconditioners do not. Nonetheless, it is possible to parallelize the application of Gauss-Seidel preconditioners through a so-called "red-black" reordering, which we will investigate in a future publication.

4. Numerical results. This section examines the performance of the three preconditioners for the optimal control problem given in Section 2. We present four experiments. The first experiment is designed to compare the effectiveness of our preconditioners with those investigated in [5]. The second experiment studies the potential for their efficient use in inexact SQP methods, such as [6]. The third experiment examines preconditioner performance for a fixed size of time subdomains. The fourth experiment examines the smoothing properties of our preconditioners, with the outlook for use in new multigrid in time methods for PDE-constrained optimization.

REMARK 4.1. *For the given data* $z(t) = r(t) = 2\pi(1-e^{-t})$, $u_0 = 0$, *and* $f(x,t) = (4\pi^2(1-e^{-t})+e^{-t})\sin(2\pi x)$, *the solution of the constraint equation* (2.2) *is* $u(t,x) = \sin(2\pi x)(1-e^{-t})$. *We verify our finite element implementation of the constraint equation by using this manufactured solution, as follows. If we consider a sufficiently fine time discretization, we observe second order convergence of the numerical PDE solution in space, and if we consider a sufficiently fine spatial discretization, we get second order convergence in time for the Crank-Nicolson method* ($\theta = 1/2$). *The implementation of the optimal control problem is verified using manufactured solutions as well, where we have studied convergence of the optimal state* u_* *in space and time.*

In all cases below, $f(t,x) = (4\pi^2(1-e^{-t})+e^{-t})\sin(2\pi x)$, $r(t) = 2\pi(1-e^{-t})$, $u_0(x) = 0$, $\alpha_2 = 0$, $s_1 = 1$, and the initial guess for the iterative solution of the optimality system is zero. We use the Crank-Nicolson scheme for the time discretization.

Experiment 1. Here we compare the performance of our preconditioners with similar preconditioners investigated in [5]. We consider the penalty parameter $\alpha_1 = 10^3$. We use $N_x = 10$ spatial intervals and $N = 30$ time intervals. The GMRES method is applied with the 2-norm residual tolerance of 10^{-7}. The maximum number of iterations for GMRES is set to 100.

The iteration numbers in Figure 4.1 for the forward and backward Gauss-Seidel preconditioners are very close to those published in [5, Fig. 4]. This implies that our Gauss-Seidel preconditioners could be used for storage reduction, similar to [5]. Unfortunately, their application, as implemented here, does not parallelize in time. In

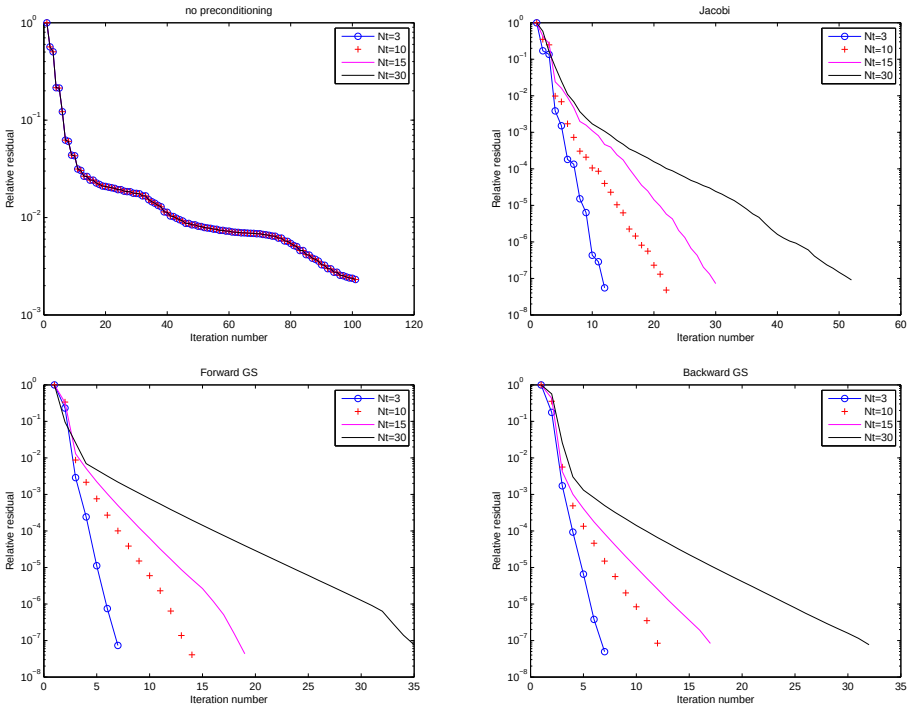


FIG. 4.1. *Convergence history of GMRES with Gauss-Seidel and Jacobi preconditioners for* $N_x = 10$, $N = 30$ *and different numbers of time subdomains,* N_t. *The iteration numbers for the forward and backward Gauss-Seidel preconditioners are very close to those published in [5, Fig. 4].*

contrast, the application of the Jacobi preconditioner, which was not studied in [5], parallelizes in time.

Experiment 2. Here we demonstrate the potential effectiveness of the Jacobi scheme as a time-parallel preconditioner for linear systems arising in, e.g., inexact sequential quadratic programming (SQP) methods. The key to efficiency in these algorithms is the use of very coarse, i.e., inexact, iterative solves for the optimality systems. It is reported in [7, 6] that residual tolerances as large as 0.5 can be handled robustly and efficiently by inexact SQP schemes, and that tolerance levels of 10^{-2} or 10^{-3} are optimal for many applications. We let $N_x = 100$ and $N = 100$, set the maximum number of GMRES iterations to 200, and keep all other parameters the same as in Experiment 1. While we are particularly interested in the performance of the Jacobi scheme, we report forward Gauss-Seidel results as well, see Figure 4.2.

Table 4.1 shows the GMRES iterations at which the relative residual drops below 10^{-3} and 10^{-2}, respectively, for both preconditioners. For the Jacobi preconditioner, the potential speedup due to time parallelization can be defined as the ratio of the number of time subdomains and the iteration numbers reported in Table 4.1. In the conservative case where the relative residual is 10^{-3}, the potential parallel speedup for 50 time subdomains is $50/18 = 2.8$, while the speedup for the relative residual of 10^{-2} is $50/7 = 7.1$. These results encourage a closer study of the time-domain decomposition Jacobi preconditioner in the context of inexact SQP methods.

Experiment 3. Here we take $N_x = 10$, set the maximum number of GMRES iterations to 100, set the relative residual tolerance to 10^{-7}, and fix the number of

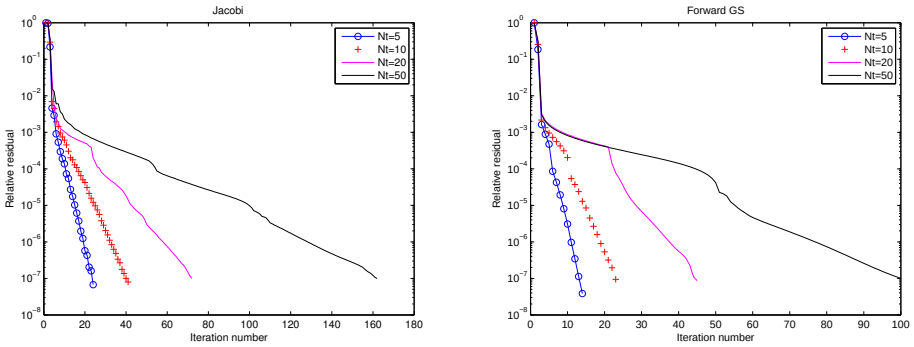


FIG. 4.2. *Convergence history of GMRES with the Jacobi and forward Gauss-Seidel preconditioners for $N_x = 100$, $N = 100$ and different numbers of time subdomains, N_t. The results are analyzed in Table 4.1.*

N_t	5	10	20	50	5	10	20	50
Forward GS	4	5	8	9	3	3	3	3
Jacobi	6	8	10	18	4	4	5	7

TABLE 4.1

The GMRES iteration number at which the relative residual drops below 10^{-3} (left part of the table) or 10^{-2} (right part of the table), for $N_x = 100$, $N = 100$ and different numbers of time subdomains, N_t. For the relative residual of 10^{-3}, the maximum parallel speedup is $50/18 = 2.8$, while the speedup for the relative residual of 10^{-2} is $50/7 = 7.1$.

time steps per time subdomain to five, $N/N_t = 5$. We vary the total number of time steps from 25 to 200. The performance of the Jacobi and forward Gauss-Seidel preconditioners is illustrated in Figure 4.3. The backward Gauss-Seidel preconditioner performs similarly to the forward Gauss-Seidel preconditioner, so we omit those results. Figure 4.3 demonstrates good iteration scaling for both preconditioners as the total number of time steps increases. Going from 25 to 200 total time steps, the number of iterations for the Jacobi preconditioner increases from 16 to only 21, while the iteration increase for the forward Gauss-Seidel scheme is only one, from 10 to 11.

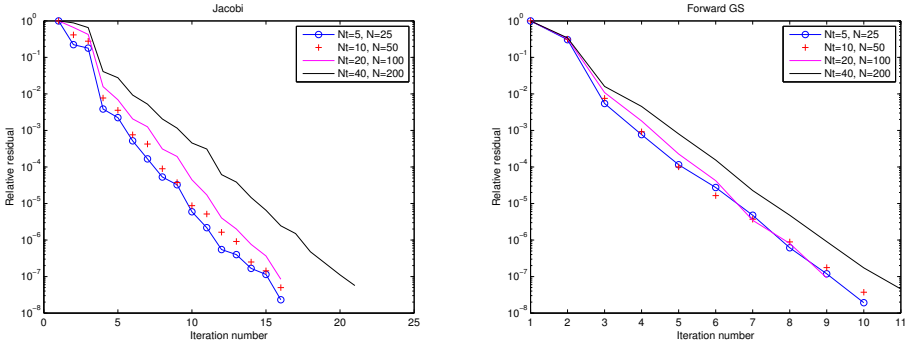


FIG. 4.3. *Convergence history of GMRES with the Jacobi and forward Gauss-Seidel preconditioners for $N_x = 10$, $N/N_t = 5$, and different total numbers of time steps, N. Both preconditioners show good iteration scaling as the total number of time steps increases.*

Experiment 4. Here we present an empirical study of the the smoothing properties of our preconditioners, based on error reduction, see e.g., [2]. The goal is to assess their potential as smoothers for multigrid-in-time methods. By writing $\mathbf{A} = \mathbf{A} + \mathbf{P} - \mathbf{P}$, we obtain a new form of $\mathbf{Ax} = \mathbf{b}$,

$$\mathbf{Px} = (\mathbf{P} - \mathbf{A})\mathbf{x} + \mathbf{b}, \tag{4.1}$$

where $\mathbf{P}$ could be any of the above preconditioners. This form suggests a fixed-point iteration,

$$\mathbf{Px}_{k+1} = (\mathbf{P} - \mathbf{A})\mathbf{x}_k + \mathbf{b}. \tag{4.2}$$

We subtract equation (4.2) from (4.1) to find the error equation

$$\mathbf{P}(\mathbf{x} - \mathbf{x}_{k+1}) = (\mathbf{P} - \mathbf{A})(\mathbf{x} - \mathbf{x}_k). \tag{4.3}$$

We introduce error vectors $\mathbf{e}_k = \mathbf{x} - \mathbf{x}_k$ and $\mathbf{e}_{k+1} = \mathbf{x} - \mathbf{x}_{k+1}$, multiply both sides of the equation by the inverse of $\mathbf{P}$, and arrive at the error update equation

$$\mathbf{e}_{k+1} = (\mathbf{I} - \mathbf{P}^{-1}\mathbf{A})\mathbf{e}_k. \tag{4.4}$$

This fixed-point iteration converges if the spectral radius of the operator $\mathbf{I} - \mathbf{P}^{-1}\mathbf{A}$ is less than one. We do not expect this to hold in the optimal control context, however it is informative to examine the error based on equation (4.4).

For all studies below we choose $N = 100$ and $N_x = 100$, and set the number of time subdomains to $N_t = 10$. Additionally, instead of enforcing $\bar{u}_i = u_i$ as the continuity condition, we will use $C\bar{u}_i = Cu_i$ to improve the scaling of the Lagrange multipliers. The starting condition $\mathbf{e}_0$ is a vector with each component randomly drawn from a uniform distribution over the open interval $(-1, 1)$. The goal is to seed the initial error with a range of spectral content in both space and time. Then after applying the relaxation method the resulting value empirically represents the smooth error. In the context of multilevel methods, this smooth error would be handled with a coarse grid correction. Multilevel schemes will be studied in a future publication.

We study the state and adjoint components of the error. Figure 4.4 has four plots of the state error (left column) and four plots of the adjoint error (right column). The first row shows the random initial condition. The second row gives a view of the error with respect to time after five applications of the Jacobi smoothing operator. The third row shows a colormap of the error in time and space after five iterations. The final row shows a similar colormap after 100 iterations. Figures 4.5 and 4.6 contain similar sets of plots for forward and backward Gauss-Seidel, respectively. There are two interesting features. First is the scaling of the error, and its reduction with increasing iteration counts. All methods reduce the error by at least an order of magnitude after 100 iterations. The second feature is the spectral content of the error. For all the methods, after five iterations some of the noise of the initial condition remains in the spatial direction. Also, the structure of the discontinuous-in-time solution becomes more apparent, as the preconditioners do not enforce continuity explicitly. After 100 iterations the noise in the spatial direction is completely eliminated. On the other hand, the error at the time discontinuities remains relatively large. This large error represents a "smooth mode" that can be removed using a coarse grid correction in multigrid methods.

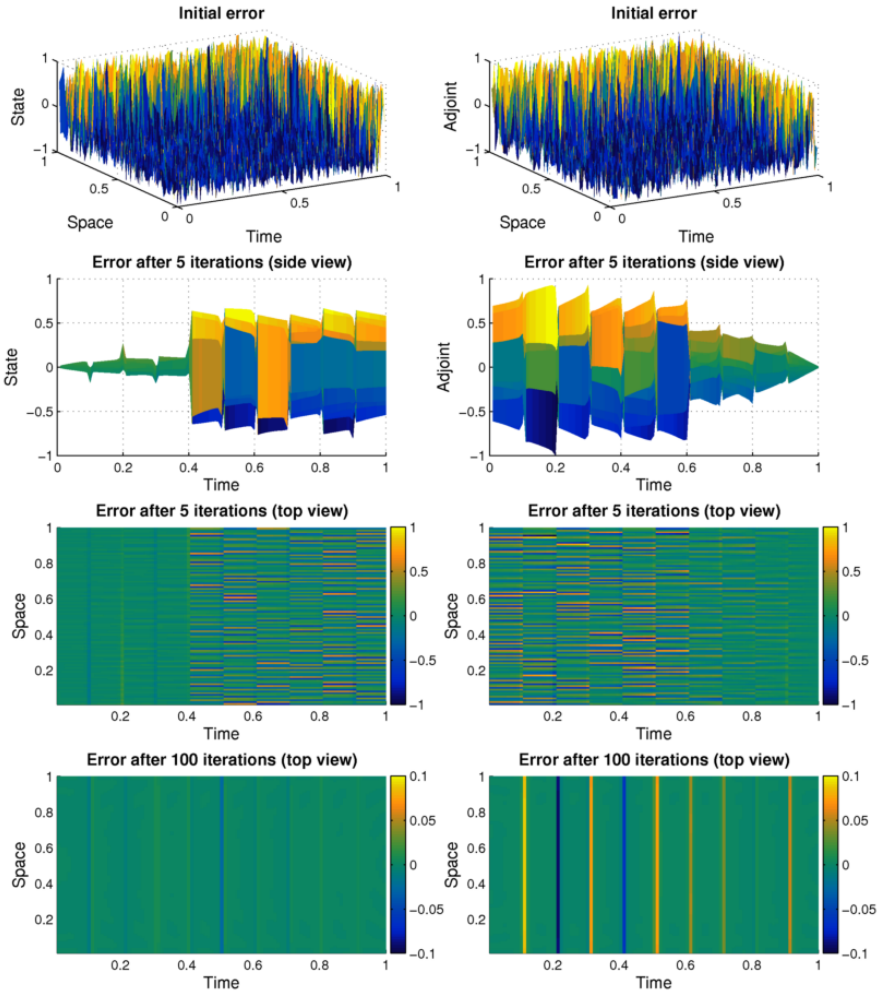


FIG. 4.4. *The state component of the error (left) and the adjoint component of the error (right) for the Jacobi preconditioner at different iterations. Note the difference in scales between the top six images and the bottom two images.*

5. Conclusions and future work. We studied a time-parallel approach to solving quadratic optimization problems with linear time-dependent PDE constraints. Our approach is based on time-domain decomposition, block tridiagonalization and the subsequent use of fixed-point iterations with matrix splittings. We demonstrated the potential effectiveness of the Jacobi scheme as a time-parallel preconditioner for linear systems arising in inexact SQP methods. We also showed that the Jacobi and Gauss-Seidel schemes may be good smoothers for multigrid in time methods, which we will study next. We will also integrate the proposed schemes into inexact matrix-free SQP methods for nonlinear optimization, and tackle optimization problems with nonlinear time-dependent PDE constraints.

REFERENCES

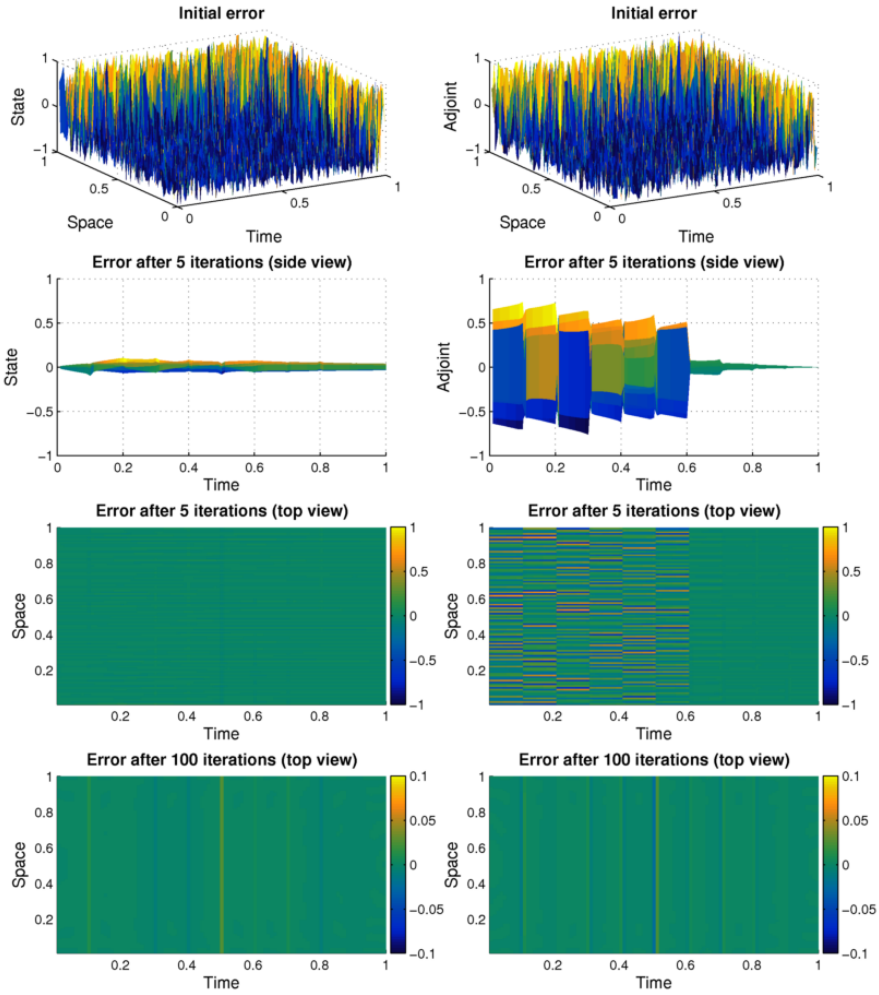


FIG. 4.5. *The state component of the error (left) and the adjoint component of the error (right) for the forward Gauss-Seidel preconditioner at different iterations. Note the difference in scales between the top six images and the bottom two images.*

[1] J. ALBERSMEYER AND M. DIEHL, *The lifted Newton method and its application in optimization*, SIAM J. Optim., 20 (2010), pp. 1655–1684.

[2] W. L. BRIGGS, S. F. MCCORMICK, ET AL., *A multigrid tutorial*, SIAM, 2000.

[3] A. COMAS, *Time-domain decomposition preconditioners for the solution of discretized parabolic optimal control problems*, ProQuest LLC, Ann Arbor, MI, 2006. Thesis (Ph.D.)–Rice University.

[4] M. J. GANDER, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition, Springer, 2015. In press.

[5] M. HEINKENSCHLOSS, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, J. Comput. Appl. Math., 173 (2005), pp. 169–198.

[6] M. HEINKENSCHLOSS AND D. RIDZAL, *A matrix-free trust-region SQP method for equality constrained optimization*, SIAM J. Optim., 24 (2014), pp. 1507–1541.

[7] D. RIDZAL, M. AGUILÓ, AND M. HEINKENSCHLOSS, *Numerical study of a matrix-free trust-region SQP method for equality constrained optimization*, Tech. Rep. SAND2011-9346, Sandia National Laboratories, 2011.

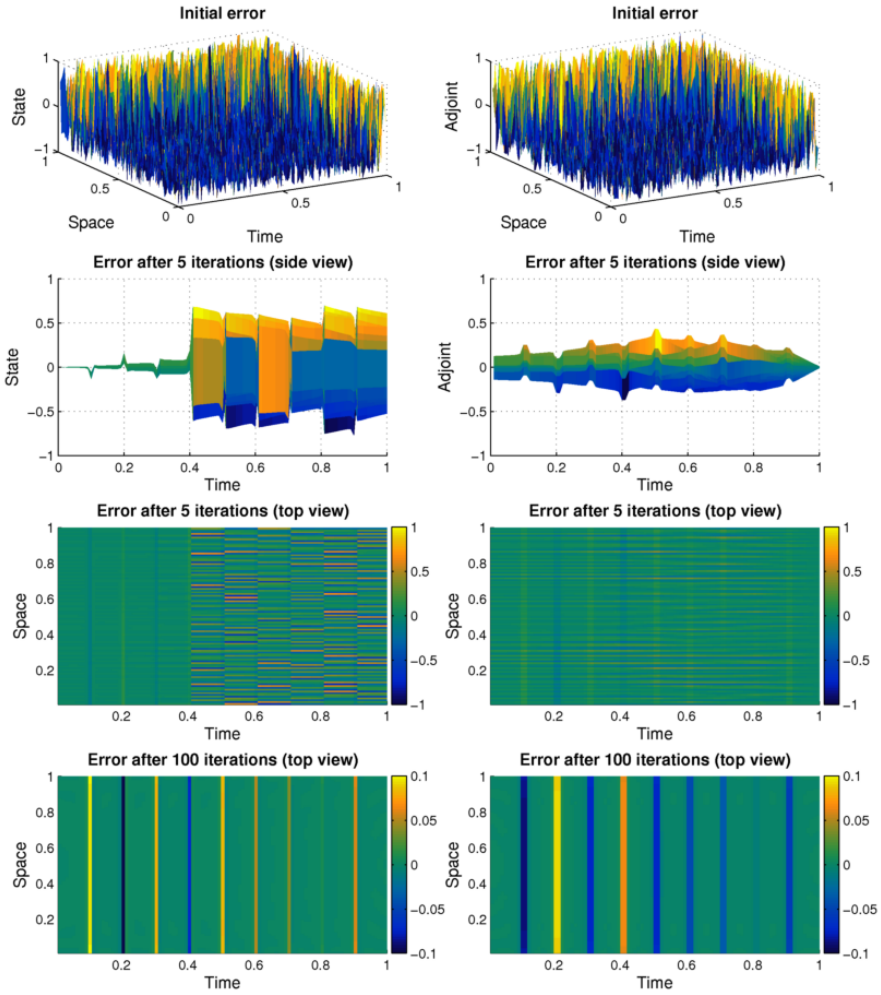


FIG. 4.6. *The state component of the error (left) and the adjoint component of the error (right) for the backward Gauss-Seidel preconditioner at different iterations. Note the difference in scales between the top six images and the bottom two images.*

[8] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.