DOMAIN DECOMPOSITION IN TIME FOR PDE-CONSTRAINED OPTIMIZATION*

ANDREW T. BARKER† AND MARTIN STOLL‡

Abstract. PDE-constrained optimization problems have a wide range of applications, but they lead to very large and ill-conditioned linear systems, especially if the problems are time dependent. In this paper we outline an approach for dealing with such problems by decomposing them in time and applying an additive Schwarz preconditioner in time, so that we can take advantage of parallel computers to deal with the very large linear systems. We then illustrate the performance of our method on a variety of problems.

Key words. PDE-constrained optimization, space-time methods, preconditioning, Schur complement, domain decomposition, parallel computing.

AMS subject classifications. 65F08, 65F10, 65F50, 92E20, 93C20

1. Introduction. Many challenging applications are modeled by partial differential equations (PDEs) and in the presence of measurements or expected data it is often desirable to tune the parameters of the equations to best reflect reality. This process is one of the core motivation in the field of PDE-constrained optimization. The goal is to find a state y and a control u to minimize

$$\mathcal{J}(y,u) = \frac{1}{2}\|y-\bar{y}\|^2_{\mathcal{L}_2(\Omega)} + \frac{\beta}{2}R(u) \tag{1.1}$$

given the expected state (or measurements) via $\bar{y}$ over a domain $\Omega \in \mathbb{R}^d$ $(d = 2, 3)$. The quantities of interest are then linked via a PDE-model that models the underlying physics and is written as

$$\mathcal{A}(y,u) = 0, \tag{1.2}$$

that is, the minimization in (1.1) is done subject to the constraint (1.2). Note that $R(u)$ is a regularization functional, which often depends on the underlying application. We here focus on the $\mathcal{L}_2$ norm of the control u. Here $\mathcal{A}$ denotes a partial differential operator equipped with appropriate boundary and initial conditions. Furthermore, β denotes a regularization parameter. Problems of this type have been of increasing interest over the last decade and we refer to [37, 17, 18, 6] for introductions to this field. Recently, computational aspects of statistical inverse problems have become a focus of many researchers as these problems are relevant when the uncertainties within a particular model are to be quantified. The problems are often of a similar nature to the problem given above (see [8, 19, 33]).

A typical solution technique for problems of the above kind is to discretize the objective function and the PDE to build a discrete Lagrangian. The first order conditions of the Lagrangian are then given by a large-scale saddle point or KKT problem [3, 12]. In case the function or the PDE are nonlinear one would additionally employ

*Portions of this work were performed under the auspices of the U.S. Department of Energy under Contract DE-AC52-07NA27344 (LLNL-JRNL-652253).

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, Mail Stop L-561, Livermore, CA 94551(barker29@llnl.gov),

‡Numerical Linear Algebra for Dynamical Systems, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de),

approaches from nonlinear optimization such as SQP or interior point methods [21]. In all cases at the heart of the algorithms lies the solution of a very large linear system. A technique that has recently been found to enable very effective numerical methods and in particular preconditioners is to employ a simultaneous discretization in both space and time [22, 4, 2].

Depending on the number of time-steps this can lead to prohibitively large vectors and one remedy is to consider parallel approaches that allow to distribute the work and storage among a possibly very large number of processors. As the storage requirements for the matrices corresponding to the spatial discretization of the PDE is essentially the same as for the steady case we here focus on a parallelization in time. For this we focus on the well-studied additive Schwarz preconditioner, decomposing the time-domain into overlapping pieces and using local, parallel solutions on these time subdomains to precondition the global linear system [36].

The paper is structured as follows. In Section 2 we introduce three different model problems, including the heat equation, the Stokes equations, and the convection-diffusion equations. Our focus here is on the discretization of the PDEs and the corresponding optimality systems. This is then followed by a description of Schur complement preconditioners in Section 3. We then discuss how this strategy can be adapted for a parallelization in time using a Schwarz preconditioner in Section 4. After discussing possible alternative we illustrate the scaling properties of our proposed method in Section 5.

2. PDE-constrained optimization model problems. We begin with the introduction of three model problems that illustrate many of the relevant structures that are encountered in PDE-constrained optimization problems. The goal of the optimization process is to bring the state y as close as possible to a desired or observed state $\bar{y}$ while using a control u, i.e.,

$$\min_{y,u} \ \frac{1}{2}\int_0^T \int_{\Omega_1} (y-\bar{y})^2 \, dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} u^2 dxdt, \tag{2.1}$$

with an observation domain $\Omega_1 \subseteq \Omega$ and a control domain $\Omega_2 \subseteq \Omega$. An obvious first choice for a time-dependent PDE connection between state and control is the heat equation

$$\begin{aligned} y_t - \triangle y &= u, \quad \text{in } \Omega, \\ y &= f, \quad \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

here equipped with a distributed control term and Dirichlet boundary condition. We can also consider the Neumann-boundary control problem defined by

$$\begin{aligned} y_t - \triangle y &= f, \quad \text{in } \Omega, \\ \frac{\partial y}{\partial \mathrm{n}} &= u, \quad \text{on } \partial\Omega. \end{aligned} \tag{2.3}$$

A detailed discussion on the well-posedness and existence of solutions can be found in [17, 18, 37]. For the solution process we form the Lagrangian to incorporate the constraints and then consider the first order optimality conditions or KKT conditions [18, 21, 37]. One can now do this by discretizing objective functions and constraints first and then optimize or first optimize and then discretize the optimality conditions. We here use the first discretize then optimize approach. Additionally, we are performing an all-at-once approach [22, 31] using a discrete problem within the space-time

cylinder $\Omega \times [0,T]$. Using the trapezoid rule in time and finite elements in space leads to the following discrete objective function

$$J(y,u) = \frac{\tau}{2}\left(y-\bar{y}\right)^T \mathcal{M}_1 \left(y-\bar{y}\right) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u. \tag{2.4}$$

Here, using $D_1 = \operatorname{diag}\left(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2}\right)$ we have $\mathcal{M}_1 = D_1 \otimes M_1, \mathcal{M}_2 = D_1 \otimes M_2$ being space-time matrices where M_1 is the mass matrix associated with the domain Ω_1 and M_2 is the corresponding mass matrix for Ω_2. The vectors $y = [y_1^T \ldots y_{n_t}^T]^T$ and $u = [u_1^T \ldots u_{n_t}^T]^T$ are space-time vectors that represent a collection of spatial vectors for all time steps.

The all-at-once discretization of the state equation using finite elements for the discretization in space and an implicit Euler scheme for the discretization in time is given by

$$\mathcal{K} y - \tau \mathcal{N} u = d \tag{2.5}$$

where

$$\mathcal{K} = \begin{bmatrix} L & & & \\ -M & L & & \\ & \ddots & \ddots & \\ & & -M & L \end{bmatrix}, \quad \mathcal{N} = \begin{bmatrix} N & & & \\ & N & & \\ & & \ddots & \\ & & & N \end{bmatrix}, \quad d = \begin{bmatrix} M_1 y_0 + f \\ f \\ \vdots \\ f \end{bmatrix}.$$

Here, M is the mass matrix for the domain Ω, L is defined as $L = M + \tau K$, where K is the stiffness matrix. The matrix N corresponds to the control term either via a distributed control (square matrix) or via the contributions of a boundary control problem (rectangular matrix), and the right-hand side d consists of a contribution from the initial condition y_0 and a vector f representing forcing terms and contributions of boundary conditions. The first order conditions using a Lagrangian formulation with Lagrange multiplier p leads to the following system

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K} & \tau\mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1\bar{y} \\ 0 \\ d \end{bmatrix}. \tag{2.6}$$

Systems of this form can be found in [31, 22, 32, 20]. These systems are of vast dimensionality, which prohibits the use of direct solvers [11, 9] and therefore it is crucial to find efficient preconditioners that are embedded into Krylov subspace methods in order to obtain an approximation to the solution.

Any Krylov method only needs the application of the system matrix to a vector and for this we do not need to construct the matrix $\mathcal{A}$ explicitly. We are able to perform this method in a matrix-free fashion. Nevertheless, we need to store the space-time vectors associated with the control, state and adjoint state. There are various schemes that can be used instead or are aimed at reducing the storage amount. We discuss this issue later in Section 4.1. Note that the simplest form of storage reduction is to work with the Schur-complement if it exists of the matrix $\mathcal{A}$ or to remove the control from the system matrix [28, 16]. As this does not reduce the main problem of efficiently approximating the Schur complement we proceed with the most general form of the unreduced system.

We now want to introduce two more model problems that result in a similar matrix structure but with a higher complexity regarding the derivation of efficient preconditioners. The first problem we consider is the optimal control of the Stokes equations

$$y_t - \nu \triangle y + \nabla p = u \text{ in } [0,T] \times \Omega \tag{2.7}$$
$$-\nabla \cdot y = 0 \text{ in } [0,T] \times \Omega \tag{2.8}$$
$$y(t,\cdot) = g(t) \text{ on } \partial\Omega, t \in [0,T] \tag{2.9}$$
$$y(0,\cdot) = y^0 \text{ in } \Omega, \tag{2.10}$$

and the objective function is again of misfit-type, i.e.,

$$J(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y-\bar{y})^2 \, dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} u^2 dxdt, \tag{2.11}$$

and proceeding by forming a discrete Lagrangian for a space time discretization we get

$$J(y,u) = \frac{\tau}{2}(y-\bar{y})^T \mathcal{M}_1 (y-\bar{y}) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u \tag{2.12}$$

(see [32]). Again, we have $\mathcal{M}_1 = D_1 \otimes M_1, \mathcal{M}_2 = D_1 \otimes M_2$ but now

$$D_1 = \text{diag}\left(\frac{1}{2}, 0, 1, 0, 1, 0, \ldots, 1, 0, \frac{1}{2}, 0\right).$$

Note that for the Stokes case the vectors y_i are split into a velocity v part with $d = 2, 3$ components and pressure part p, i.e.,

$$y_i = \begin{bmatrix} y_i^v \\ y_i^p \end{bmatrix}.$$

Similarly, for the discretized control u and the adjoint state p. The all-at-once discretization of the state equation using Q_2/Q_1 finite elements in space and an implicit Euler scheme in time is given by

$$\mathcal{K} y - \tau \mathcal{N} u = d \tag{2.13}$$

where we use the following

$$\mathcal{K} = \begin{bmatrix} \mathcal{L} & & & \\ -\mathcal{M} & \mathcal{L} & & \\ & \ddots & \ddots & \\ & & -\mathcal{M} & \mathcal{L} \end{bmatrix}, \quad \mathcal{N} = I_{N_T} \otimes \mathcal{N}_s, \quad d = \begin{bmatrix} Ly_0 + f \\ 0 \\ f \\ \vdots \\ f \\ 0 \end{bmatrix}.$$

In the Stokes case we have a 2×2 structure of the discretized PDE written as

$$\mathcal{L} = \begin{bmatrix} L & B^T \\ B & 0 \end{bmatrix},$$

which represents an instance of a time-dependent Stokes problem. Here B is the discrete divergence, M is the mass matrix for the domain Ω, the matrix L is defined as $L = \tau^{-1}M + K$, the matrix

$$\mathcal{N}_s = \begin{bmatrix} N \\ 0 \end{bmatrix}$$

corresponds to the distributed control term where $N = M$, and the matrix

$$\mathcal{M} = \begin{bmatrix} \tau^{-1}M & 0 \\ 0 & 0 \end{bmatrix}$$

is associated with the discretization in time via the implicit Euler scheme. The right-hand side d consists of a contribution from the initial condition y_0 and a vector f representing forcing terms and contributions of boundary conditions. Note that all matrices here correspond to the ones introduced for the heat equation but equipped with a block form corresponding to the components for the velocity y_v and pressure y_p. The first order conditions are then written as

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \mathcal{N}^T \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1\bar{y} \\ 0 \\ d \end{bmatrix}. \tag{2.14}$$

Before proceeding to our numerical scheme we introduce one more problem setup. The objective function

$$J(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y-\bar{y})^2 \, dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} u^2 dxdt. \tag{2.15}$$

is again the misfit function but the PDE constraint is now given by the convection diffusion equation

$$y_t - \varepsilon\triangle y + w \cdot \nabla y = u \text{ in } \Omega \tag{2.16}$$

$$y(:,x) = g \text{ on } \partial\Omega \tag{2.17}$$

$$y(0,:) = y_0. \tag{2.18}$$

The parameter ε is crucial to the convection-diffusion equation as a decrease in its value is adding more hyperbolicity to the PDE where the wind w is predefined. Such optimization problems have recently been discussed in [26, 15, 24]. We use here the symmetric interior penalty discontinuous Galerkin discretization in space, where the discretize-then-optimize and optimize-then-discretize approaches can be shown to commute [1, 40, 34]. Other possible approaches such as the streamline upwind Galerkin (SUPG) approach [7] or local projection stabilization [24] could also be used within our framework. Once again we employ a trapezoid rule in connection with finite elements and now the discretized objective function and state equation are given by

$$J(y,u) = \frac{\tau}{2}\left(y-\bar{y}\right)^T \mathcal{M}_1 \left(y-\bar{y}\right) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u,$$

which is the same as for the heat equation case. For the all-at-once discretization of the convection-diffusion equation we get the same structure as for the heat equation in (2.13)–(2.6), in particular

$$\mathcal{K}y - \tau\mathcal{N}u = d \tag{2.19}$$

with

$$\mathcal{K} = \begin{bmatrix} L_s & & & \\ -M_s & L_s & & \\ & \ddots & \ddots & \\ & & -M_s & L_s \end{bmatrix}, \quad \mathcal{N} = \begin{bmatrix} M_s & & & \\ & M_s & & \\ & & \ddots & \\ & & & M_s \end{bmatrix}, \quad d = \begin{bmatrix} M_1 y_0 + f \\ f \\ \vdots \\ f \end{bmatrix}.$$

Here M_s is the standard discontinuous Galerkin mass matrix,

$$L_s = M_s + \tau(\varepsilon K_s + C_s)$$

is the system matrix for the convection-diffusion system where K_s is the DG Laplacian and C_s represents the convection operator.

3. Schur complement preconditioning. Studying the structure of the linear systems introduced in the previous section we see that the $(1,1)$-block $\mathrm{blkdiag}(\tau\mathcal{M}_1, \tau\beta\mathcal{M}_2)$ of $\mathcal{A}$ is typically not overly complicated. This can change significantly when the linear system arises during the iteration of a nonlinear solver caused by a nonlinear objective function, PDE, or both. Nevertheless, the structure of the $(1,1)$-block is naturally easier than the associated Schur complement. Hence, we briefly overview previously established Schur-complement type approaches to preconditioning and solving the time-dependent PDE-constrained optimization problems outlined above. We explain why these methods are not directly applicable in a parallel computing setting, but they will form an important part of our overall algorithm so it is worth briefly reviewing them here. For a more thorough treatment see [22, 23, 25, 29, 32, 27] where also the approximation of the $(1,1)$-block is discussed. Our point of departure is the Schur complement

$$S = \tau^{-1}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T$$

and we hope to approximate it as best as possible while using cheap-to-apply methods. The simplest idea is to ignore one of the terms in S but this usually does not give the desired robustness with respect to the parameters τ and β. Thus, we use preconditioners based on the following decomposition

$$S \approx \tau^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)^T$$

where we see that the first term in S is obviously represented and we compute $\hat{\mathcal{M}}$ in such a way that

$$\tau^{-1}\hat{\mathcal{M}}\mathcal{M}_1^{-1}\hat{\mathcal{M}}^T = \frac{\tau}{\beta}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T.$$

The matrix $\hat{\mathcal{M}}$ can efficiently be chosen for the three examples introduced above. Not presenting the details, $\hat{\mathcal{M}}$ will often be a block-diagonal matrix scaled by terms involving the problem parameters such as τ and β. The solution of the system $\left(\mathcal{K} + \hat{\mathcal{M}}\right)$ then is similar to solving with the matrix $\mathcal{K}$, which is, of course, a block-triangular matrix. As the inversion is only needed within the preconditioner we do not need to solve this exactly but rather approximately. This means we approximate the diagonal blocks of the block-triangular matrix $\left(\mathcal{K} + \hat{\mathcal{M}}\right)$ by a multigrid process and then proceed by forward substitution.

4. A Schwarz preconditioner. Additive Schwarz preconditioning is a well-established domain decomposition strategy that has been used with great success for a wide variety of problems [36]. Here we focus on a simple additive Schwarz domain decomposition method in time for the discrete systems (2.6), (2.14). We begin by partitioning the time domain $\mathcal{T} = [0, T]$ into N_p non-overlapping time subdomains and then extending each subdomain to overlap its neighbors by an amount δ. For simplicity we will take δ to be an integer multiple of the time step size τ and we will denote the overlapping subdomains by $\mathcal{T}_k, k = 1, \ldots, N_p$.

Then on each time subdomain we formulate a discrete PDE-constrained optimization problem analogous to the original one. In particular, let $n_{t,k}$ be the number of time steps in $\mathcal{T}_k$, and define R_k with $n_{t,k}$ block rows and n_t block columns such that the block (i, j) of R_k is the (spatial) identity matrix if the global time step j corresponds to the ith time step in $\mathcal{T}_k$ and zero otherwise. As an example, if $n_t = 5$ and there are two subdomains with $n_{t,1} = n_{t,2} = 3$, then

$$R_1 = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{pmatrix}, R_2 = \begin{pmatrix} 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}. \tag{4.1}$$

Then we can define a local optimization operator

$$\mathcal{A}_k = \begin{pmatrix} R_k & & \\ & R_k & \\ & & R_k \end{pmatrix} \mathcal{A} \begin{pmatrix} R_k^T & & \\ & R_k^T & \\ & & R_k^T \end{pmatrix} \tag{4.2}$$

where we are using the same restriction and interpolation in time for the three components, that is the notation is using a distributed control but the same idea can handle the case of boundary control. Now we can define the one-level additive Schwarz preconditioner

$$B_{as}^{-1} = \sum_{k=1}^{N_p} R_k^T \mathcal{A}_k^{-1} R_k \tag{4.3}$$

where we understand the inverse of the local operator $\mathcal{A}_k$ in (4.3) to indicate a solution to a local optimization problem in time using the standard Schur complement approach outlined in the previous section.

Unfortunately, the preconditioner B_{as}^{-1} is in general indefinite (as it reflects the structure of the original indefinite matrix $\mathcal{A}$), and since MINRES requires a symmetric positive definite preconditioner we need a different linear solver. Since the underlying system and preconditioning are still symmetric, we choose a symmetric variant of QMR [13]. In addition, since we will want to solve the local subproblems inexactly, we employ the flexible QMR variant of Szyld and Vogel [35].

To summarize, the system (2.6) or (2.14) is solved with a flexible, symmetric QMR iteration. This QMR iteration is preconditioned with the one-level additive Schwarz preconditioner (4.3). Within the Schwarz preconditioner, inverting the local operators $\mathcal{A}_k$ is approximated independently on each processor by a MINRES method, and this MINRES is itself preconditioned using the Schur complement approach from Section 3.

The one-level Schwarz preconditioner is known to not scale to very large number of processors, a situation for which we need exchange of global information on coarser

meshes, that is, we need a two-level or multi-level Schwarz preconditioner. Such a scalable preconditioner is the subject of ongoing research. Nevertheless, we will see in Section 5 that for moderate processor counts the current one-level implementation scales quite well.

4.1. Alternative approaches. Parallel solvers and preconditioners are considered for PDE-constrained optimization problems in [5], but the problems considered are not time dependent, and the time dependence is a key focus of our work as this greatly increases the overall size of the system and gives the physics a different character.

In [10], a parallel in time method is introduced for similar problems, but here the authors use a reduced Hessian approach for the control variables only, while we want to solve for all variables at once. As a result, their parallel solvers in time involve forward and backward sweeps with the parareal algorithm, while we are interested in preserving both the forward and backward coupling in time within the preconditioner, not separating them into different sweeps.

In [39] and [38], parallel Schwarz methods are used in space for time–dependent PDE–constrained optimization problems. In these works the authors employ a so-called "suboptimal control" approach, where an optimal control problem is solved over a series of short time intervals to approximate the solution to the optimal control problem over the whole interval. The result is an algorithm which is parallel in space but sequential in time—in contrast our approach is sequential in space but parallel in time, and we are interested in finding the true optimal control for the entire interval.

The approach in the literature which is perhaps the most similar to ours in spirit is that of [14], which uses a kind of Gauss-Seidel-Schwarz domain decomposition in time. The use of large scale parallel computing for this approach is, however, not as straightforward as in our approach, and indeed the numerical examples presented in [14] are all rather small. In addition, this approach breaks the symmetry of the underlying KKT system, which we view as an undesirable property.

Recently, a technique based on low-rank presentations for the solution vectors was introduced [30]. The method allows to reduce the storage requirement for the solution vectors by constantly performing low-rank approximation to the solution. Currently, the method is limited to very specific structures of both the PDE and the objective function, whereas the techniques presented here are very general.

5. Numerical experiments. All of the numerical results in this section are performed on a Linux cluster with 90 nodes, each of which has 2 Intel Xeon X5650 CPUs, each of which has 6 cores. We run with 12 MPI processes per node, and do not distinguish between intranode and internode parallelism. Each node has 48 GB of memory, and in Infiniband network connects the nodes.

5.1. Heat equation. Here we report numerical results for the heat equation constrained optimization problem (2.2) with $\beta = 10^{-4}$. Our model problem has the desired state

$$\bar{y}(x) = 64t \sin(2\pi|x - (1/2, 1/2, 1/2)|^2).$$

This desired state can be seen in Figure 5.1 along with the computed value. Since our β is quite small, the computed solution comes very close to the desired state, except near the boundaries where $\bar{y}$ does not satisfy the boundary conditions. By the same token, a small β allows the control to be quite large, as shown in Figure 5.2.

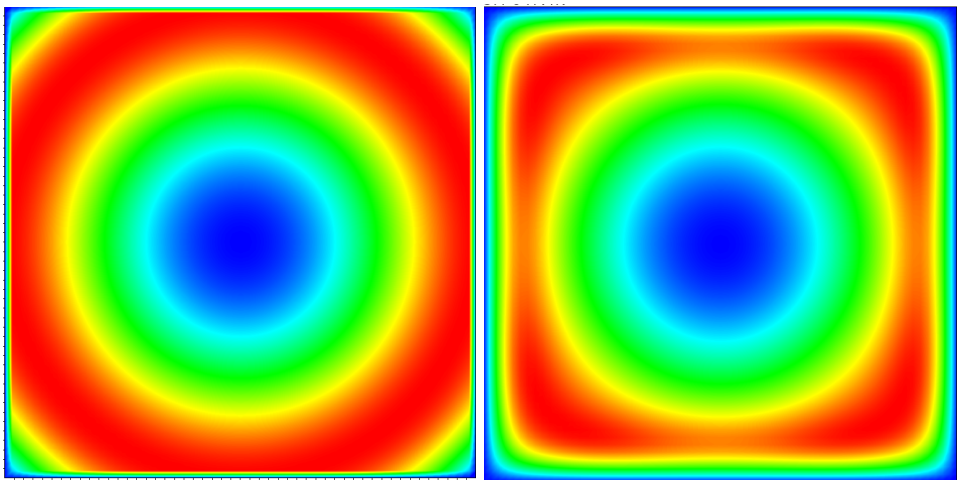

Fig. 5.1: Slice of a 3D solution for the heat equation constrained optimization problem (2.2) with $\beta = 10^{-6}$, desired state on the left, achieved solution on the right.

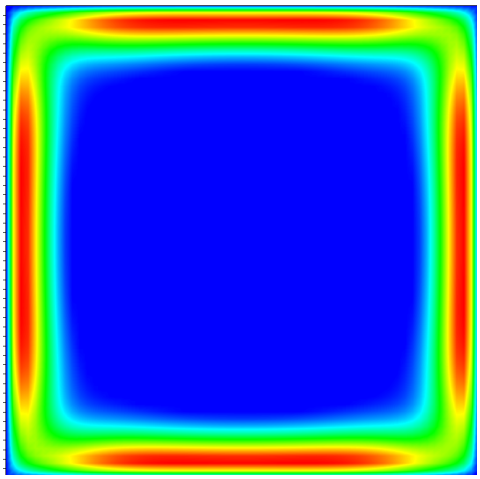

Fig. 5.2: Slice of the calculated control for a 3D solution for the heat equation constrained optimization problem with $\beta = 10^{-6}$.

The first parallel results we present are concerned not so much with parallel efficiency as with our ability to simply solve very large problems. To that end, we run the heat constrained optimization problem with 64 cores and scale up the problem size as far as possible, with results shown in Table 5.1. Here the number of spatial degrees of freedom is kept fixed at 275000, while we increase the number of time steps, which is our primary interest in this paper. We are able to solve problems with over 800 million unknowns, problem sizes that would be completely impossible without parallel computing and domain decomposition in time.

To illustrate the parallel efficiency of our algorithms presented in Section 4, we

Table 5.1: Scaling with respect to problem size, heat equation constrained optimization problems running on 64 cores with an overlap of 2.

N	N_T	N_x	iterations	time (sec)
1.05e8	128	274 625	10	1890.4
2.11e8	256		13	4385.3
4.22e8	512		17	5249.3
8.44e8	1024		16	9191.8

Table 5.2: Parallel scaling for the heat equation, with 35937 spatial degrees of freedom, 256 time steps, an overlap of 2 time steps, and an increasing number of cores.

cores	iterations	time (sec)	time/iteration
4	9	1848.8	205.4
8	10	890.6	89.1
16	10	609.3	60.9
32	10	379.7	38.0
64	14	367.8	26.3
128	15	339.5	22.6

present strong scaling results with a fixed problem size and increasing number of processing cores in Table 5.2. The iteration counts are very reasonable and increase somewhat as we scale the problem, as expected for a one-level Schwarz preconditioner. In terms of time to solution we do see some speedup from using parallel computing, although for this particular problem there are diminishing returns for using more than 32 parallel processes, largely do to increasing iteration counts. The use of a two-level or multi-level Schwarz preconditioner could allow for scaling to more processors on the parallel machine.

We also present weak scaling results in Table 5.3. Here the problem size is increased with the number of cores, so that we hope for constant run times. Even with only a one-level preconditioner, the iteration counts are quite small and grow only slowly. The times reported in this table are not quite constant but they grow slowly, so that for larger problems we can see that using larger numbers of cores is beneficial.

5.2. Stokes equation. For our numerical approach to the Stokes equation (2.7), our implementation is in two spatial dimensions. The problem is based on standard driven cavity flow, where the desired state corresponds to a driven cavity flow with steady flow on the lid. In contrast, the actual state is subject to an oscillating flow on the lid, and the distributed control is employed to drive the flow toward the steady–lid case. A representative picture of the velocity magnitude is in Figure 5.3.

Strong and weak scaling results for this problem are presented in Tables 5.4 and 5.5. Although we include timing results here, this implementation has not been optimized very carefully and the primary purpose of these results is to show that the outer iteration counts increase quite slowly as we scale to larger problems and more cores. With some additional effort we expect that the timings could be improved to be more in line with the timings for the heat equation problem above.

Table 5.3: Weak scaling for the heat equation, fixed at 274625 spatial degrees of freedom but with the number of time steps increasing as the number of cores is also increased. Overlap is set to 2.

N_T	cores	iterations	time (sec)
32	2	9	2994.0
64	4	8	3243.8
128	8	8	4272.1
256	16	9	4556.4
512	32	11	5265.3

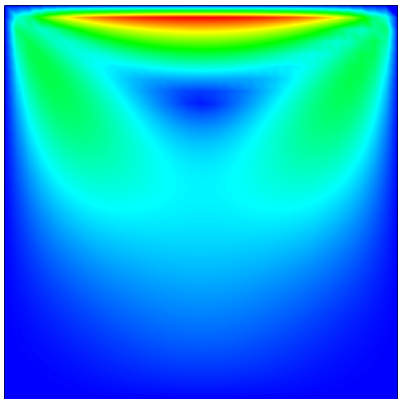

Fig. 5.3: Achieved state for the Stokes equation constrained optimization problem.

Table 5.4: Parallel scaling for the Stokes problem in two dimensions, with 37507 spatial degrees of freedom and $N_T = 256$ time steps.

cores	iterations	time (sec)
8	9	39400
16	9	20700
32	9	12000
64	11	8900

Table 5.5: Weak scaling for the Stokes problem in two dimensions, with a fixed number of 37507 spatial degrees of freedom and with number of time steps and cores increasing together.

N_T	cores	iterations	time (sec)
32	2	3	4320
64	4	4	6420
128	8	7	16700
256	16	9	20800
512	32	12	31200
1024	64	17	42200

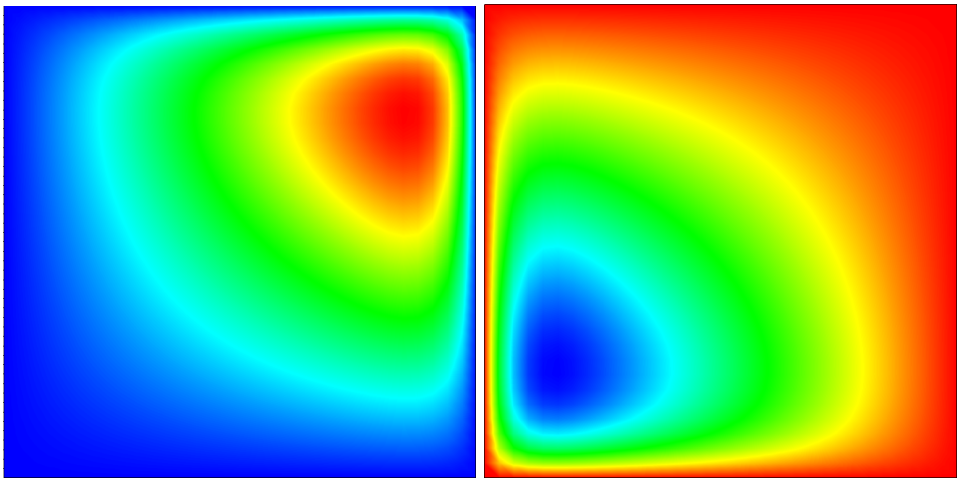

Fig. 5.4: Convection diffusion model problem, solution on the left, adjoint on the right, at time $t = \pi/2$.

5.3. Convection–diffusion equation. Here we present numerical results for the convection-diffusion model problem from (2.16). For our model problem, let

$$\eta(z,\alpha) = z - \frac{\exp(\alpha(z-1)/\epsilon) - \exp(-\alpha/\epsilon)}{1-\exp(-\alpha/\epsilon)}$$
$$\mu(z,\alpha) = 1 - z - \frac{\exp(-\alpha z/\epsilon) - \exp(-\alpha/\epsilon)}{1-\exp(-\alpha/\epsilon)}$$

and then choose the right-hand-side f and the desired state $\bar{y}$ so that the true solution y and the true adjoint p are given by

$$y = \sin(t)\eta(x,w_x)\eta(y,w_y)\eta(z,w_z)$$
$$p = -\sin(t)\eta(y,w_y)\eta(z,w_z) + w_y\eta(x,w_x)\eta(z,w_z) + w_z\eta(x,w_x)\eta(y,w_y)$$

where $w = (w_x, w_y, w_z)$ is the wind or advection direction. Pictures of typical state an adjoint variables are shown in Figure 5.4

Strong and weak scaling results in terms of iterations are shown in Tables 5.6 and 5.7. Since some of these results were done on a different computer than the others, we do not present timings, but qualitatively the parallel performance is similar to the other model problems in this section. For the convection-diffusion problem the non-optimality of the one-level preconditioner is visible, but we still see reasonable iteration counts for this problem.

This problem has quite a few parameters, including τ, h, β and ϵ, and we want our preconditioning strategy to be robust with respect to a wide range of these parameters. In Table 5.8 we consider the interplay of β and τ, seeing that in every case we get reasonable iteration numbers, and in Table 5.9 we consider the interplay of ϵ and β, noting that the case of relatively small ϵ can be a difficult case for convection-diffusion problems and is important in applications. We note that thanks to the discontinuous Galerkin discretization and our regularization-robust preconditioners we have good iteration counts for all these cases.

Table 5.6: Strong scaling for the convection-diffusion problem in three spatial dimensions with 32768 spatial degrees of freedom and 127 time steps, $\beta = 0.1, \epsilon = 0.1$, and overlap of 2 timesteps.

cores	iterations
4	8
8	7
16	9
32	10

Table 5.7: Weak scaling for the convection-diffusion problem in three spatial dimensions with 32768 spatial degrees of freedom, $\beta = 0.1, \epsilon = 0.1$, and overlap of 1 timestep

N_T	cores	iterations
64	2	10
127	4	7
255	8	9
511	16	10
1023	32	17

6. Conclusions. Our goal in this paper has been to address one of the drawbacks of the all-at-once approach to time dependent PDE-constrained optimization, namely the storage of the extremely large vectors that arise in the all-at-once systems. We have demonstrated that these large systems and vectors can be dealt with using a straightforward Schwarz preconditioner in the time domain, while still maintaining the good convergence properties of the approach, including the robustness with respect to the regularization parameter, timestep size, and other physical parameters that may arise in particular problems.

Although a complete theoretical treatment of the approach is out of reach, we have used known theory for simpler problems to motivate our approach and explain why it makes sense and can be expected to lead to well-conditioned systems. In addition, our numerical results have shown that this approach is effective for several different PDEs, that it can solve problems with hundreds of millions of unknowns, and that it achieves good parallel scaling on a moderate number of processors.

Future work will include developing a truly scalable two–level or multi–level preconditioner, extending the parallelism to both space and time, and considering more complicated and nonlinear problems such as the Navier–Stokes equations.

REFERENCES

[1] T. Akman, H. Yücel, and B. Karasözen, *A priori error analysis of the upwind symmetric interior penalty galerkin (sipg) method for the optimal control problems governed by unsteady convection diffusion equations*, Computational Optimization and Applications, (2013), pp. 1–27.

[2] R. Andreev and C. Tobler, *Multilevel preconditioning and low rank tensor iteration for space-time simultaneous discretizations of parabolic pdes*, Tech. Rep. 2012–16, Seminar for Applied Mathematics, ETH Zürich, 2012, 2012.

[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.

Table 5.8: Robustness with β. In three spatial dimensions with 32768 spatial degrees of freedom, running on 32 cores, $\epsilon = 0.1$.

	β			
N_T	10^{-2}	10^{-3}	10^{-4}	10^{-6}
127	11	7	4	3
255	12	8	7	3
511	14	10	6	3
1023	14	11	7	3

Table 5.9: Interplay of β and ϵ, with $N_T = 127$, in three spatial dimensions, with 32768 spatial degrees of freedom, running on 32 cores.

	β				
ϵ	0.1	0.01	10^{-3}	10^{-4}	10^{-6}
1.0	7	7	6	4	3
0.1	33	11	7	4	3
10^{-2}	18	14	7	5	3
10^{-4}	32	13	7	4	3

[4] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Advances in Computational Mathematics, 35 (2011), pp. 149–173.

[5] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. I. The Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[6] A. Borzì and V. Schulz, *Computational optimization of systems governed by partial differential equations*, vol. 8, SIAM, Philadelphia, 2012.

[7] A. N. Brooks and T. J. Hughes, *Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations*, Computer methods in applied mechanics and engineering, 32 (1982), pp. 199–259.

[8] D. Calvetti and E. Somersalo, *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, vol. 2, Springer, 2007.

[9] T. Davis, *Umfpack version 4.4 user guide*, tech. rep., Dept. of Computer and Information Science and Engineering Univ. of Florida, Gainesville, FL, 2005.

[10] X. Du, M. Sarkis, C. E. Schaerer, and D. B. Szyld, *Inexact and truncated parareal-in-time krylov subspace methods for parabolic optimal control problems*, Tech. Rep. 12-02-06, Department of Mathematics, Temple University, Feb. 2012.

[11] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Monographs on Numerical Analysis, The Clarendon Press Oxford University Press, New York, 1989.

[12] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[13] R. W. Freund and N. M. Nachtigal, *A New Krylov-Subspace Method for Symmetric Indefinite Linear Systems*, in Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, E. W. F. Ames, ed., IMACS, 1994, pp. 1253–1256.

[14] 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.

[15] M. Heinkenschloss and D. Leykekhman, *Local error estimates for supg solutions of advection-dominated elliptic linear-quadratic optimal control problems*, SIAM Journal on Numerical Analysis, 47 (2010), pp. 4607–4638.

[16] M. Hinze, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, Comput. Optim. Appl., 30 (2005), pp. 45–61.

[17] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.

[18] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[19] J. P. Kaipio and E. Somersalo, *Statistical and computational inverse problems*, vol. 160, Springer, 2005.

[20] T. P. Mathew, M. Sarkis, and C. E. Schaerer, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM J. Sci. Comput., 32 (2010), pp. 1180–1200.

[21] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[22] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.

[23] J. W. Pearson, M. Stoll, and A. J. Wathen, *Robust Iterative Solution of a Class of Time-Dependent Optimal Control Problems*, Submitted, (2012).

[24] J. W. Pearson and A. J. Wathen, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Submitted, (2011).

[25] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numerical Linear Algebra with Applications, 19 (2012), pp. 816–829.

[26] T. Rees, *Preconditioning Iterative Methods for PDE Constrained Optimization*, PhD thesis, University of Oxford, 2010.

[27] T. Rees, M. Stoll, and A. Wathen, *All-at-once preconditioners for PDE-constrained optimization*, Kybernetika, 46 (2010), pp. 341–360.

[28] V. Simoncini, *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems*, To appear in Computational Optimization and Applications, (2012).

[29] M. Stoll, *All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, Accepted IMA Journal of numerical analysis, (2013).

[30] M. Stoll and T. Breiten, *A low-rank in time approach to PDE-constrained optimization*, Submitted, (2013).

[31] M. Stoll and A. Wathen, *All-at-once solution of time-dependent PDE-constrained optimization problems*, tech. rep., University of Oxford, 2010.

[32] ———, *All-at-once solution of time-dependent Stokes control*, Journal of Computational Physics, 232 (2013), pp. 498–515.

[33] A. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numerica, 19 (2010), pp. 451–559.

[34] T. Sun, *Discontinuous galerkin finite element method with interior penalties for convection diffusion optimal control problem*, Int. J. Numer. Anal. Mod, 7 (2010), pp. 87–107.

[35] D. B. Szyld and J. A. Vogel, *FQMR: a flexible quasi-minimal residual method with inexact preconditioning*, SIAM J. Sci. Comput., 23 (2001), pp. 363–380.

[36] A. Toselli and O. Widlund, *Domain decomposition methods–algorithms and theory*, vol. 34, Springer Verlag, 2005.

[37] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Amer Mathematical Society, 2010.

[38] H. Yang and X.-C. Cai, *Parallel fully implicit two-grid Lagrange-Newton-Krylov-Schwarz methods for distributed control of unsteady incompressible flows*, Int. J. Numer. Meth. Fluids, 72 (2013), pp. 1–21.

[39] H. Yang, E. E. Prudencio, and X.-C. Cai, *Fully implicit Lagrange-Newton-Krylov-Schwarz algorithms for boundary control of unsteady incompressible flows*, Int. J. Numer. Meth. Engng., 91 (2012), pp. 644–665.

[40] H. Yücel, M. Heinkenschloss, and B. Karasözen, *Distributed optimal control of diffusion-convection-reaction equations using discontinuous galerkin methods*, in Numerical Mathematics and Advanced Applications 2011, Springer, 2013, pp. 389–397.