



MAX-PLANCK-GESELLSCHAFT

Martin Stoll

Tobias Breiten

**A low-rank in time approach to  
PDE-constrained optimization**



**Max Planck Institute Magdeburg  
Preprints**

**MPIMD/13-08**

**June 26, 2013**

**Impressum:**

**Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg**

**Publisher:**

Max Planck Institute for Dynamics of Complex  
Technical Systems

**Address:**

Max Planck Institute for Dynamics of  
Complex Technical Systems  
Sandtorstr. 1  
39106 Magdeburg

[www.mpi-magdeburg.mpg.de/preprints](http://www.mpi-magdeburg.mpg.de/preprints)

# A LOW-RANK IN TIME APPROACH TO PDE-CONSTRAINED OPTIMIZATION

MARTIN STOLL\* AND TOBIAS BREITEN†

**Abstract.** The solution of time-dependent PDE-constrained optimization problems is a challenging task in numerical analysis and applied mathematics. All-at-once discretizations and corresponding solvers provide efficient methods to robustly solve the arising discretized equations. One of the drawbacks of this approach is the high storage demand for the vectors representing the discrete space-time cylinder. We here introduce a low-rank in time technique that exploits the low-rank nature of the solution. The theoretical foundations for this approach originate in the numerical treatment of matrix equations and can be carried over to PDE-constrained optimization. We illustrate how three different problems can be rewritten and used within a low-rank Krylov subspace solver with appropriate preconditioning.

**Key words.** PDE-constrained optimization, low-rank methods, space-time methods, preconditioning, Schur complement, matrix equations.

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

**1. Introduction.** Many complex phenomena in the natural, engineering and life sciences are modeled using partial differential equations (PDEs). To obtain optimal configurations of these equations one typically formulates this as a PDE-constrained optimization problem of the following form

$$\min \mathcal{J}(y, u)$$

subject to

$$\mathcal{L}(y, u) = 0.$$

With  $\mathcal{J}(y, u)$  the functional of interest and  $\mathcal{L}(y, u)$  representing the differential operator. Problems of this type have been carefully analyzed in the past (see [46, 81] and the references therein).

Recently with the advancement of algorithms and technology, research has focused on the efficient numerical solution of these problems. In this paper we focus on the efficient solution of the discretized first order conditions in a space-time framework. The KKT conditions when considered in an all-at-once approach, i.e., simultaneous discretization in space and time, are typically of vast dimensionality. Matrix-free approaches have recently been developed to guarantee the (nearly) optimal convergence of iterative Krylov subspace solvers. The focus both for steady [65, 70] and transient problems [58, 78] has been on the development of efficient preconditioning strategies for the linear system that typically are of structured form (see [16, 26] for introductions to the numerical solution of saddle point systems).

One of the obstacles using a space-time discretization is the storage requirement for the large vectors needed to represent the solution at all times. Approaches such as checkpointing [37] or multiple shooting [42] are possible alternatives. We here want

---

\*Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de),

†Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (breiten@mpi-magdeburg.mpg.de),

to introduce an alternative to these schemes that can for certain problems provide an efficient representation with a minimal amount of storage. We are basing our methodology on recent developments within the solution of large and sparse matrix equations, see e.g., [4, 13, 24, 27, 35, 47, 48, 51, 67, 71, 74, 82] and references therein. One classical representative in this category is the Lyapunov equation

$$AX + XA^T = -\tilde{C}\tilde{C}^T$$

where we are interested in approximating the matrix-valued unknown  $X$ . Solving this system is equivalent to solving the linear system

$$(I \otimes A + A \otimes I)x = \tilde{c}$$

where  $x$  and  $\tilde{c}$  are related to  $X$  and  $\tilde{C}\tilde{C}^T$ , respectively. For details on the relevance of this equation within control theory, see [3, 41, 49]. In [15, 53, 61, 62, 67] the authors have introduced low-rank iterative schemes that approximate intermediate iterates  $X_k$  in a low-rank fashion that is maintained until convergence. We can exploit these technologies for problems coming from PDE-constrained optimization. It is not expected that these techniques outperforms optimal solvers with a few time-steps. The more crucial component is that they enable computations with many time-steps that would otherwise not be possible.

The paper is structured as follows. In Section 2 we introduce the heat equation as our model problem and discuss its discretization. Section 3 illustrates how this problem can be reformulated using Kronecker technology and how we need to adapt a standard Krylov-subspace solver to be able to solve this problem efficiently. As we need a preconditioner for fast convergence we next discuss possible preconditioners in Section 3. We provide some theoretical results in Section 4. Section 5 is devoted to illustrating that our methodology can be carried over to other state equations such as Stokes equations and the convection-diffusion equation. Finally, in Section 6 we illustrate the competitiveness of our approach.

**2. A PDE-constrained optimization model problem.** We start the derivation of the low-rank in time method by considering an often used model problem in PDE-constrained optimization (see [44, 46, 81]) that nevertheless reflects the crucial structure exhibited by many problems of similar type. Our goal is the minimization of a misfit functional that aims at bringing the state  $y$  as close as possible to a desired or observed state  $y_{obs}$  while using a control  $u$ , i.e.,

$$\min_{y,u} \frac{1}{2} \|y - y_{obs}\|_{L_2(\Omega_1)}^2 + \frac{\beta}{2} \|u\|_{L_2(\Omega_2)}^2, \quad (2.1)$$

$$(2.2)$$

subject to a partial differential equation that connects both state and control, referred to as the state equation. We start by considering the heat equation with a distributed control term

$$\begin{aligned} y_t - \nabla^2 y &= u, & \text{in } \Omega, \\ y &= f, & \text{on } \partial\Omega, \end{aligned} \quad (2.3)$$

or equipped with Neumann-boundary control

$$\begin{aligned} y_t - \nabla^2 y &= f, & \text{in } \Omega, \\ \frac{\partial y}{\partial \mathbf{n}} &= u, & \text{on } \partial\Omega. \end{aligned} \quad (2.4)$$

For a more detailed discussion on the well-posedness, existence of solutions, etc. we refer the interested reader to [44, 46, 81]. Classically these problems are solved using a Lagrangian to incorporate the constraints and then consider the first order optimality conditions or KKT conditions [46, 55, 81]. This can be done either by forming a discrete Lagrangian and then perform the optimization procedure or by first considering an infinite-dimensional Lagrangian for whose first order conditions we employ a suitable discretization. We here perform the first approach although much of what we state in this paper is valid for both cases. Our goal is to build a discrete Lagrangian using an all-at-once approach [58, 78] using a discrete problem within the space-time cylinder  $\Omega \times [0, T]$ . Using the trapezoidal rule in time and finite elements in space leads to the following discrete objective function

$$J(y, u) = \frac{\tau}{2} (y - y_{obs})^T \mathcal{M}_1 (y - y_{obs}) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u \quad (2.5)$$

with  $\mathcal{M}_1 = \text{blkdiag}(\frac{1}{2}M_1, M_1, \dots, M_1, \frac{1}{2}M_1)$ ,  $\mathcal{M}_2 = \text{blkdiag}(\frac{1}{2}M_2, M_2, \dots, M_2, \frac{1}{2}M_2)$  being space-time matrices where  $M_1$  is the mass matrix associated with the domain  $\Omega_1$  and  $M_2$  is the corresponding mass matrix for  $\Omega_2$ . The vectors  $y = [y_1^T \dots y_{n_t}^T]^T$  and  $u = [u_1^T \dots u_{n_t}^T]^T$  are of vast dimensionality and represent a collection of spatial vectors for all time steps collected into one single vector.

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

$$\mathcal{K}y - \tau\mathcal{N}u = d \quad (2.6)$$

where

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

Here,  $M$  is the mass matrix for the domain  $\Omega$ , the matrix  $L$  is defined as  $L = M + \tau K$ , the matrix  $N$  represents 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

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K} & \tau\mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1 y_{obs} \\ 0 \\ d \end{bmatrix}. \quad (2.7)$$

Systems of this form have previously been studied in [78, 58, 79, 54]. As these systems are of vast dimensionality it is crucial to find appropriate preconditioners together with Krylov subspace solvers to efficiently obtain an approximation to the solution. The vast dimensionality of system matrices does not allow the use of direct solvers [25, 22] but we can employ Krylov subspace solvers in a matrix free way by never forming the matrix  $\mathcal{A}$  and only implicitly performing the matrix-vector product. The

main bottleneck of this approach is the storage requirement for the space-time vectors which can be reduced by working on the Schur-complement if it exists of the matrix  $\mathcal{A}$  or to remove the control from the system matrix [73, 42]. Other approaches that can be employed are checkpointing schemes [37] or multiple shooting approaches [39]. We in the following want to present an alternative that is using the underlying tensor structure of the first order conditions.

**3. A Kronecker view.** We noticed earlier that the linear system in (2.7) is of vast dimensionality and that we only need very few matrices to efficiently perform the matrix vector multiplication with  $\mathcal{A}$  and we can approach this in a matrix-free form by never forming  $\mathcal{A}$ . Nevertheless, the vectors  $y$ ,  $u$ , and  $p$  themselves are enormous and every storage reduction would help to improve the performance of an optimization scheme. The goal now is to employ the structure of the linear system to reduce the storage requirement for the iterative method. Our approach is based on recent developments for matrix equations [10, 51, 35]. Using the definition of the Kronecker product

$$W \otimes V = \begin{bmatrix} w_{11}V & \dots & w_{1m}V \\ \vdots & \ddots & \vdots \\ w_{n1}V & \dots & w_{nm}V \end{bmatrix}$$

we note that (2.7) can also be written as

$$\underbrace{\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -(I_{n_t} \otimes L + C^T \otimes M) \\ 0 & D_2 \otimes \beta \tau M_2 & D_3 \otimes \tau N^T \\ -(I_{n_t} \otimes L + C \otimes M) & D_3 \otimes \tau N & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} D_1 \otimes \tau M_1 y_{obs} \\ 0 \\ d \end{bmatrix}, \quad (3.1)$$

where  $D_1 = D_2 = \text{diag}(\frac{1}{2}, 1, \dots, 1, \frac{1}{2})$  and  $D_3 = I_{n_t}$ . Other choices for these matrices are of course possible. Additionally, the matrix  $C \in \mathbb{R}^{n_t, n_t}$  is given by

$$C = \begin{bmatrix} 0 & & & & & \\ -1 & 0 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}$$

and represents the implicit Euler scheme. It is of course possible to use a different discretization in time. So far we have simply reformulated the previously given system. But our goal was to derive a scheme that allows for a reduction in storage requirement for the vectors  $y$ ,  $u$ , and  $p$ . For this we remind the reader of the definition of the vec operator via

$$\text{vec}(W) = \begin{bmatrix} w_{11} \\ \vdots \\ w_{n1} \\ \vdots \\ w_{nm} \end{bmatrix}$$

as well as the relation

$$(W^T \otimes V) \text{vec}(Y) = \text{vec}(VYW).$$

Now employing this and using the notation

$$Y = [ y_1, y_2, \dots, y_{n_t} ], \quad U = [ u_1, u_2, \dots, u_{n_t} ], \quad P = [ p_1, p_2, \dots, p_{n_t} ]$$

we get that

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -(I_{n_t} \otimes L + C^T \otimes M) \\ 0 & D_2 \otimes \beta \tau M_2 & D_3 \otimes \tau N^T \\ -(I_{n_t} \otimes L + C \otimes M) & D_3 \otimes \tau N & 0 \end{bmatrix} \begin{bmatrix} \text{vec}(Y) \\ \text{vec}(U) \\ \text{vec}(P) \end{bmatrix} \quad (3.2)$$

$$= \text{vec} \left( \begin{bmatrix} \tau M_1 Y D_1^T - L P I_{n_t}^T - M P C \\ \tau \beta M_2 U D_2^T + \tau N^T P D_3^T \\ -L Y I_{n_t}^T - M Y C^T + \tau N U D_3^T \end{bmatrix} \right).$$

So far nothing is gained from rewriting the problem in this form. As was previously done in [10] we assume for now that if  $Y$ ,  $U$ , and  $P$  can be represented by a low-rank approximation any iterative Krylov subspace solver can be implemented using a low-rank version of (3.2). We denote the low-rank representations by

$$Y = W_Y V_Y^T \text{ with } W_Y \in \mathbb{R}^{n_1, k_1}, V_Y \in \mathbb{R}^{n_t, k_1} \quad (3.3)$$

$$U = W_U V_U^T \text{ with } W_U \in \mathbb{R}^{n_2, k_2}, V_U \in \mathbb{R}^{n_t, k_2} \quad (3.4)$$

$$P = W_P V_P^T \text{ with } W_P \in \mathbb{R}^{n_1, k_3}, V_P \in \mathbb{R}^{n_t, k_3} \quad (3.5)$$

with  $k_{1,2,3}$  being small in comparison to  $n_t$  and rewrite (3.2) accordingly to get

$$\begin{bmatrix} \tau M_1 W_Y V_Y^T D_1^T - L W_P V_P^T I_{n_t}^T - M W_P V_P^T C \\ \tau \beta M_2 W_U V_U^T D_2^T + \tau N^T W_P V_P^T D_3^T \\ -L W_Y V_Y^T I_{n_t}^T - M W_Y V_Y^T C^T + \tau N W_U V_U^T D_3^T \end{bmatrix}, \quad (3.6)$$

where we skipped the  $\text{vec}$  operator and formulated the problem as a matrix equation with matrix-valued unknowns. Note that we can write the block-rows of (3.6) as

$$\begin{aligned} \text{(first block-row)} \quad & [ \tau M_1 W_Y \quad -L W_P \quad -M W_P ] \begin{bmatrix} V_Y^T D_1^T \\ V_P^T I_{n_t}^T \\ V_P^T C \end{bmatrix}, \\ \text{(second block-row)} \quad & [ \tau \beta M_2 W_U \quad \tau N^T W_P ] \begin{bmatrix} V_U^T D_2^T \\ V_P^T D_3^T \end{bmatrix}, \quad (3.7) \\ \text{(third block-row)} \quad & [ -L W_Y \quad -M W_Y \quad \tau N W_U ] \begin{bmatrix} V_Y^T I_{n_t}^T \\ V_Y^T C^T \\ V_U^T D_3^T \end{bmatrix}. \end{aligned}$$

We obtain a significant storage reduction if we can base our approximation of the solution using the low-rank factors (3.7). It is easily seen that due to the low-rank nature of the factors we have to perform fewer multiplications with the submatrices by also maintaining smaller storage requirements. As the usage of a direct solver is out of the question we here rely on a preconditioned Krylov subspace solver, namely MINRES introduced in [56] as the underlying matrix is symmetric and indefinite. Before explaining all the intricacies of the method we state the resulting algorithm and carefully explain the necessary details afterwards. Algorithm 1 shows a low rank

implementation of the classical preconditioned MINRES method as presented in [56]. Note that due to a small truncation tolerance being employed within our algorithm we treat all quantities as being almost as exact representations of their full-rank equivalents. In the future this should be investigated and flexible solvers should be employed to account for the changes in the application of the preconditioner.

It is hard to hide the fact that the low-rank version presented here seems much messier than its vector-based relative. This is due to the fact that we want to maintain the structure of the saddle point system, which is reflected in low-rank representations associated with the state (all matrices with indices 11 and 12), the control (all matrices with indices 21 and 22), and the Lagrange multiplier (all matrices with indices 31 and 32). Please keep in mind that

$$\text{vec} \left( \begin{bmatrix} Z_{11} Z_{12}^T \\ Z_{21} Z_{22}^T \\ Z_{31} Z_{32}^T \end{bmatrix} \right) = z$$

corresponds to the associated vector  $z$  from a vector-based version of MINRES.

```

Zero-Initialization of  $V_{11}^{(0)}, \dots, W_{11}^{(0)}, \dots$ , and  $W_{11}^{(1)}, \dots$ 
Choose  $U_{11}^{(0)}, U_{12}^{(0)}, U_{21}^{(0)}, U_{22}^{(0)}, U_{31}^{(0)}, U_{32}^{(0)}$ 
Set  $V_{11}, V_{12}, \dots$  to normalized residual
while residual norm > tolerance do
   $Z_{11}^{(j)} = Z_{11}^{(j)} / \gamma_j, Z_{21}^{(j)} = Z_{21}^{(j)} / \gamma_j, Z_{31}^{(j)} = Z_{31}^{(j)} / \gamma_j,$ 
   $[F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}] = \text{Amult}(Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$ 
   $\delta_j = \text{traceproduct}(F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}, Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$ 
   $V_{11}^{(j+1)} = \left\{ \begin{matrix} F_{11} & -\frac{\delta_j}{\gamma_j} V_{11}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{11}^{(j-1)} \end{matrix} \right\}, V_{12}^{(j+1)} = \left\{ \begin{matrix} F_{12} & V_{12}^{(j)} & V_{12}^{(j-1)} \end{matrix} \right\}$ 
   $V_{21}^{(j+1)} = \left\{ \begin{matrix} F_{21} & -\frac{\delta_j}{\gamma_j} V_{21}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{21}^{(j-1)} \end{matrix} \right\}, V_{22}^{(j+1)} = \left\{ \begin{matrix} F_{22} & V_{22}^{(j)} & V_{22}^{(j-1)} \end{matrix} \right\}$ 
   $V_{31}^{(j+1)} = \left\{ \begin{matrix} F_{31} & -\frac{\delta_j}{\gamma_j} V_{31}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{31}^{(j-1)} \end{matrix} \right\}, V_{32}^{(j+1)} = \left\{ \begin{matrix} F_{32} & V_{32}^{(j)} & V_{32}^{(j-1)} \end{matrix} \right\}$ 
   $\left\{ Z_{11}^{(j+1)}, Z_{12}^{(j+1)}, Z_{21}^{(j+1)}, Z_{22}^{(j+1)}, Z_{31}^{(j+1)}, Z_{32}^{(j+1)} \right\} =$ 
   $\text{Aprec}(V_{11}^{(j+1)}, V_{12}^{(j+1)}, V_{21}^{(j+1)}, V_{22}^{(j+1)}, V_{31}^{(j+1)}, V_{32}^{(j+1)})$ 
   $\gamma_{j+1} = \sqrt{\text{traceproduct}(Z_{11}^{(j+1)}, \dots, V_{11}^{(j+1)}, \dots)}$ 
   $\alpha_0 = c_j \delta_j - c_{j-1} s_j \gamma_j$ 
   $\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$ 
   $\alpha_2 = s_j \delta_j + c_{j-1} c_j \gamma_j$ 
   $\alpha_3 = s_{j-1} \gamma_j$ 
   $c_{j+1} = \frac{\alpha_0}{\alpha_1}$ 
   $s_{j+1} = \frac{\gamma_{j+1}}{\alpha_1}$ 
   $W_{11}^{(j+1)} = \left\{ \begin{matrix} Z_{11}^{(j)} & -\alpha_3 W_{11}^{(j-1)} & -\alpha_2 W_{11}^{(j)} \end{matrix} \right\}, W_{12}^{(j+1)} = \left\{ \begin{matrix} Z_{12}^{(j)} & W_{12}^{(j-1)} & W_{12}^{(j)} \end{matrix} \right\}$ 
   $W_{21}^{(j+1)} = \left\{ \begin{matrix} Z_{21}^{(j)} & -\alpha_3 W_{21}^{(j-1)} & -\alpha_2 W_{21}^{(j)} \end{matrix} \right\}, W_{22}^{(j+1)} = \left\{ \begin{matrix} Z_{22}^{(j)} & W_{22}^{(j-1)} & W_{22}^{(j)} \end{matrix} \right\}$ 
   $W_{31}^{(j+1)} = \left\{ \begin{matrix} Z_{31}^{(j)} & -\alpha_3 W_{31}^{(j-1)} & -\alpha_2 W_{31}^{(j)} \end{matrix} \right\}, W_{32}^{(j+1)} = \left\{ \begin{matrix} Z_{32}^{(j)} & W_{32}^{(j-1)} & W_{32}^{(j)} \end{matrix} \right\}$ 
if Convergence criterion fulfilled then
  Compute approximate solution
  stop
end if
end while

```

Algorithm 1: Low-rank MINRES

For Algorithm 1 to be accessible to the reader, we need to dissect its different parts. Starting with the inner-products of the classical MINRES method we see that  $(z^{(j)}, v^{(j)})$  with

$$\text{vec} \left( \begin{bmatrix} Z_{11}^{(j)} (Z_{12}^{(j)})^T \\ Z_{21}^{(j)} (Z_{22}^{(j)})^T \\ Z_{31}^{(j)} (Z_{32}^{(j)})^T \end{bmatrix} \right) = z$$

and similarly for  $v^{(j)}$  using that

$$\text{trace}(A^T B) = \text{vec}(A)^T \text{vec}(B)$$

$\delta$  (for convenience ignoring the index  $j$ ) can be computed via

$$\begin{aligned} \delta_j &= \text{trace} \left( (Z_{11} Z_{12}^T)^T (V_{11} V_{12}^T) \right) \\ &+ \text{trace} \left( (Z_{21} Z_{22}^T)^T (V_{21} V_{22}^T) \right) \\ &+ \text{trace} \left( (Z_{31} Z_{32}^T)^T (V_{31} V_{32}^T) \right). \end{aligned} \quad (3.8)$$

Note that so far we have rewritten the vector-problem in matrix form but the interested reader might have noted that the matrices formed as part of (3.8) are of the full dimensionality  $n \times n_t$  in the case of a distributed control problem. Due to the properties of the trace operator we are in luck as

$$\text{trace} \left( (Z_{11} Z_{12}^T)^T (V_{11} V_{12}^T) \right) = \text{trace} (Z_{11}^T V_{11} V_{12}^T Z_{12})$$

allows us to compute the trace of small matrices rather than of the ones from the full temporal/spatial discretization. We denote the reformulation of the trace in Algorithm 1 by the term traceproduct.

Input:  $W_{11}, W_{12}, W_{21}, W_{22}, W_{31}, W_{32}$   
Output:  $Z_{11}, Z_{12}, Z_{21}, Z_{22}, Z_{31}, Z_{32}$   
 $Z_{11} = \begin{bmatrix} \tau M_1 W_{11} & -LW_{31} & -MW_{31} \end{bmatrix}$   
 $Z_{21} = \begin{bmatrix} \tau \beta M_2 W_{21} & \tau N W_{31} \end{bmatrix}$   
 $Z_{31} = \begin{bmatrix} -LW_{11} & -MW_{31} & \tau N W_{21} \end{bmatrix}$   
 $Z_{12} = \begin{bmatrix} D_1 W_{12} & I_{n_t} W_{32} & C^T W_{32} \end{bmatrix}$   
 $Z_{22} = \begin{bmatrix} D_2 W_{22} & D_3 W_{32} \end{bmatrix}$   
 $Z_{32} = \begin{bmatrix} I_{n_t} W_{12} & C W_{12} & D_3 W_{22} \end{bmatrix}$

Algorithm 2: Matrix multiplication

We have now defined the matrix vector multiplication in Algorithm 2 and the efficient computation of the inner products within the low-rank MINRES algorithm. We have not yet defined the brackets  $\{\}$ . The brackets  $U := \{U_1 \ V_1 \ W_1\}$  and  $\{U_2 \ V_2 \ W_2\}$  can be understood as a concatenation and truncation, i.e., short for the truncation of the related  $\text{trunc}(U_1 U_2^T + V_1 V_2^T + W_1 W_2^T)$ . We now briefly discuss the possibilities to perform the truncation of two matrices  $V$  and  $U$  that represent the low-rank representation of  $Z = VU^T$ . As discussed in [51] we can perform skinny QR factorizations of both matrices, i.e.,  $V = Q_v R_v$  and  $U = Q_u R_u$ . We then note that

$Z = Q_v R_v R_u^T Q_u^T$ . A singular value decomposition [31] of the matrix  $R_v R_u^T = B \Sigma C^T$  and using a low-rank approximation via  $B(:, 1 : k) \Sigma(1 : k, 1 : k) C(:, 1 : k)^T$  leads to the overall low-rank approximation  $V_{new} = Q_v B(:, 1 : k)$  and  $U_{new} = Q_u C(:, 1 : k) \Sigma(1 : k, 1 : k)$ . We have implemented this approach in MATLAB<sup>®</sup> but noted that the computation of the skinny QR factorization was rather slow. Alternatively, we exploited the MATLAB<sup>®</sup> function `svds` function to directly compute a truncated singular value decomposition of  $VU^T$  by passing a function handle that allowed the implicit application of the  $Z = VU^T$  without ever forming this matrix. This approach proved advantageous in terms of the time needed for the truncation. Note that alternative ways to compute the truncated SVD are of course possible [45, 6, 76].

Before discussing the possible preconditioners we state that the vector update formulas given in Algorithm 1 are straightforward versions of vector versions of MINRES. The necessity to precondition has not changed in comparison to the classical algorithm and we devote the next section to a discussion of possible preconditioning strategies.

**Preconditioning for low-rank MINRES** . The study of preconditioners for the optimal control subject to parabolic PDEs has recently seen developments that were aimed at providing robust performance with respect to the many system parameters such as mesh-size or regularization parameters (see [70, 58, 57, 50]). We start our derivation of suitable preconditioners based on an approach presented by Pearson *et al.* [58, 60] where we start with a block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & & \\ & A_1 & \\ & & \hat{S} \end{bmatrix} \quad (3.9)$$

where  $A_0 \approx \tau \mathcal{M}_1$  and  $A_1 \approx \tau \beta \mathcal{M}_2$  are approximations to the upper left block of  $\mathcal{A}$  and  $\hat{S}$  is an approximation to the Schur-complement

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

One approximation that has proven to be very effective [58, 60] is of the form

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

where in the case of a distributed control problem the matrix  $\hat{\mathcal{M}}$  is given by

$$\hat{\mathcal{M}} = \frac{\tau}{\sqrt{\beta}} \text{blkdiag}(M, \dots, M).$$

This approach will be the basis for the derivation of efficient preconditioners for the low-rank version of MINRES. For this we need the preconditioner  $\mathcal{P}$  to maintain the low-rank structure as described in (3.7). Due to the nature of the upper left block of  $\mathcal{A}$  given by

$$\begin{bmatrix} D_1 \otimes \tau \mathcal{M}_1 & 0 \\ 0 & D_2 \otimes \beta \tau \mathcal{M}_2 \end{bmatrix}$$

we see that an efficient preconditioner given for example by

$$\begin{bmatrix} D_1 \otimes \tau \hat{\mathcal{M}}_1 & 0 \\ 0 & D_2 \otimes \beta \tau \hat{\mathcal{M}}_2 \end{bmatrix}$$

where the mass matrices are approximated by the Chebychev semi-iteration [83], will naturally maintain the desired structure. But what can be said about the Schur-complement  $S$  of the above system. Starting from the previously used approximation

$$\hat{S} = \tau^{-1} \left( I_{n_t} \otimes \hat{L} + C \otimes M \right) \mathcal{M}_1 \left( I_{n_t} \otimes \hat{L} + C \otimes M \right)^T,$$

where  $\hat{L} = \left( \left( 1 + \frac{\tau}{\sqrt{\beta}} \right) M_1 + \tau K \right)$ , we see that there already exists an inherent tensor structure within this approximation. In [78] the authors observe that such a system can be easily solved as the matrix  $\left( I_{n_t} \otimes \hat{L} + C \otimes M \right)$  is of block-triangular nature. This means one can sequentially pass through the vectors associated with each grid-point in time. For our purpose the block-triangular nature will not be sufficient to guarantee the low-rank preserving nature of our algorithm. In simple terms, a low-rank factorization in time does not allow for a temporal decoupling of the time-steps as vector for each time-step are not readily identified. In mathematical terms we can see that it is not possible to explicitly write down the inverse of  $\left( I_{n_t} \otimes \hat{L} + C \otimes M \right)$ . Our starting point is a Block-Jacobi version of the Schur-complement approximation. This procedure is motivated by the fact that we can simply write

$$\left( I_{n_t} \otimes \hat{L} \right)^{-1} = \left( I_{n_t} \otimes \hat{L}^{-1} \right).$$

The last expression assures us that this preconditioner applied to any vector  $v = \text{vec}(RS^T)$  can be written as

$$\text{vec} \left( \hat{L}^{-1} RS^T I_{n_t} \right).$$

We can now simply use the Schur-complement approximation

$$\hat{S} = \tau^{-1} \left( I_{n_t} \otimes \hat{L} \right) \mathcal{M}_1 \left( I_{n_t} \otimes \hat{L} \right)^T$$

or when using  $\hat{S} = \tau^{-1} \left( I_{n_t} \otimes \hat{L} + C \otimes M \right) \mathcal{M}_1 \left( I_{n_t} \otimes \hat{L} + C \otimes M \right)^T$  approximate the inverse of  $\left( I_{n_t} \otimes \hat{L} + C \otimes M \right)$  by a small, fixed number of steps of a stationary iteration with the block-diagonal preconditioner  $\left( I_{n_t} \otimes \hat{L} \right)$ . As was noted in [77] a time-periodic control problem where  $y(0, \cdot) = y(T, \cdot)$  results in the matrix  $C$  having circulant structure and we can then make use of the Fourier transform to obtain a Schur-complement system with only block-diagonal matrices, that are now of complex nature.

Similar structures are obtained in [1] for the simultaneous discretization in space and time. Preconditioning results using tensor structures are found in [2, 23].

A possible and very likely more parameter-robust way for preconditioning is given by interpreting  $\left( I_{n_t} \otimes \hat{L} + C \otimes M \right)$  as the Kronecker representation of the generalized Sylvester operator  $\mathcal{S}(X) = \hat{L}X + MXC^T$ . As we already mentioned at the beginning, there exist several low-rank methods such as the ADI iteration, see e.g. [14], and projection-based methods, see e.g. [71], that allow to approximately solve linear matrix equations of this type. Hence, alternative preconditioning techniques would be to apply a few steps of the ADI iteration to the generalized Sylvester equation or solve a reduced Sylvester equation resulting from the projection onto a suitable

smaller (Krylov) subspace. However, due to limitation of the scope of this paper, here we refrain from these latter ideas and instead propose them as possible topics of future research. The incorporation of approximate Sylvester solvers is straightforward and should be complemented by the use of an outer flexible method.

**4. Existence result of low-rank solutions.** The previously derived low-rank method of course is only competitive if the solution to the optimal control problem exhibits a fast singular value decay, allowing to replace it by a low-rank approximation. It thus remains to show that this is indeed a reasonable assumption for problems of the form (2.7). For this reason, in this section we establish a direct connection between (2.7) and the more prominent Sylvester equation

$$AX + XB = \tilde{C}, \quad (4.1)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{m \times m}$  and  $\tilde{C} \in \mathbb{R}^{n \times m}$ . For the case that  $\tilde{C}$  is of low-rank, i.e.,  $\tilde{C} = W_{\tilde{C}} V_{\tilde{C}}^T$ ,  $W_{\tilde{C}} \in \mathbb{R}^{n \times k}$ ,  $V_{\tilde{C}} \in \mathbb{R}^{m \times k}$  and  $k \ll n, m$ , it is well-known, see e.g. [35, 34, 52], that there exist approximations  $X_r = W_X V_X^T \approx X$ , with  $W_X \in \mathbb{R}^{n \times r}$ ,  $V_X \in \mathbb{R}^{m \times r}$  and  $r \ll n, m$ . Moreover, recently there has been an increased interest in numerical methods that, rather than computing the true solution and computing an approximation afterwards, solely work on low-rank factors and iteratively construct approximations  $X_r$  converging to the true solution  $X$ , making these approaches feasible for dimensions  $n, m \sim 10^6$ . Popular methods are projection-based methods, see [9, 29, 72], ADI-based methods, see [8, 14, 12, 75] and multigrid methods, see [36].

Let us now consider the second block-row of (2.7) for which we obtain that

$$(D_2 \otimes \beta \tau M_2) u + (D_3 \otimes \tau N^T) p = 0.$$

Solving this equation for  $u$  and inserting the result into the third block-row of (2.7) gives

$$-(I_{n_t} \otimes L + C \otimes M) y - \frac{1}{\beta} (D_3 \otimes \tau N) (D_2^{-1} \otimes M_2^{-1}) (D_3 \otimes N^T) p = d$$

which, due to the properties of the Kronecker product and the definition of  $D_3$  can be simplified to

$$-(I_{n_t} \otimes L + C \otimes M) y - \frac{\tau}{\beta} (D_2^{-1} \otimes N M_2^{-1} N^T) p = d.$$

Together with the first block-row, we thus can reformulate (2.7) in matrix notation as

$$\begin{aligned} \tau M_1 Y D_1 - L P - M P C &= \tau M_1 Y_{obs} D_1 \\ -L Y - M Y C^T - \frac{\tau}{\beta} N M_2^{-1} N^T P D_2^{-1} &= D. \end{aligned}$$

So far, we have only eliminated the second block-row and rewritten the problem in its matrix form. For the connection to (4.1), we have to make some additional assumptions on our initial setup (2.1). Typically, in real-life applications we can only observe a small portion  $\tilde{y}$  of the state rather than the full  $y$ . In other words, the mass matrix  $M_1$  in this case can be replaced by a low-rank matrix  $\tilde{C}_{obs} \tilde{C}_{obs}^T = M_1$ , with  $\tilde{C}_{obs} \in \mathbb{R}^{n_1 \times \ell}$  determining the observable parts of  $y$ . Note that in the context of

classical control theory,  $\tilde{C}_{obs}$  simply denotes the measurable output quantity of interest within the state-space representation of a linear dynamical system, see [3, 41, 49]. Similarly, in case of boundary control, the rectangular matrix  $N \in \mathbb{R}^{n_1 \times m}$  usually contains significantly less columns than rows. In summary, this means that we are often interested in the solution  $[Y \ P]$  of the linear matrix equation

$$\begin{aligned} L [Y \ P] \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix} + M [Y \ P] \begin{bmatrix} 0 & -C^T \\ -C & 0 \end{bmatrix} + \tilde{C}_{obs} \tilde{C}_{obs}^T [Y \ P] \begin{bmatrix} \tau D_1 & 0 \\ 0 & 0 \end{bmatrix} \\ + N M_2^{-1} N^T [Y \ P] \begin{bmatrix} 0 & 0 \\ 0 & -\frac{\tau}{\beta} D_2^{-1} \end{bmatrix} = \begin{bmatrix} \tau M_1 Y_{obs} D_1 \\ D \end{bmatrix}. \end{aligned} \quad (4.2)$$

Pre- and post-multiplying the previous equation by  $M^{-1}$  and  $\begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix}$  leads to a generalized Sylvester equation of the form

$$A\mathcal{X} + \mathcal{X}B + \mathcal{Q}_1\mathcal{X}\mathcal{R}_1 + \mathcal{Q}_2\mathcal{X}\mathcal{R}_2 = \mathcal{E}_1\mathcal{F}_2^T,$$

where

$$\begin{aligned} A &= M^{-1}L, \quad B = \begin{bmatrix} C^T & 0 \\ 0 & C \end{bmatrix}, \quad \mathcal{Q}_1 = M^{-1}\tilde{C}_{obs}\tilde{C}_{obs}^T, \\ \mathcal{R}_1 &= \begin{bmatrix} 0 & -\tau D_1 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{Q}_2 = M^{-1}N M_2^{-1}N^T, \quad \mathcal{R}_2 = \begin{bmatrix} 0 & 0 \\ -\frac{\tau}{\beta} D_2^{-1} & 0 \end{bmatrix}, \\ \mathcal{E}_1 &= [M_1 W_{Y_{obs}} \quad W_D], \quad \mathcal{F}_1 = \begin{bmatrix} D_1 V_{Y_{obs}} & 0 \\ 0 & V_D \end{bmatrix}. \end{aligned}$$

Note that we assumed  $Y_{obs} = W_{Y_{obs}} V_{Y_{obs}}^T$  and  $Y_D = W_D V_D^T$  to be the low-rank representations for the right hand side.

In what follows, we proceed as in [10, 21] and use the Sherman-Morrison-Woodbury formula [31] to simplify the previous equation. Since  $\mathcal{Q}_1 = \mathcal{U}_1 \mathcal{V}_1^T$  and  $\mathcal{Q}_2 = \mathcal{U}_2 \mathcal{V}_2^T$ , for the Kronecker structured linear system, we subsequently obtain

$$(\mathcal{I} \otimes A + \mathcal{B}^T \otimes \mathcal{I} + \mathcal{R}_1^T \otimes \mathcal{Q}_1 + \mathcal{R}_2^T \otimes \mathcal{Q}_2) \text{vec}(\mathcal{X}) = \text{vec}(\mathcal{E}_1 \mathcal{F}_1^T)$$

which can be rewritten as

$$\left( \underbrace{\mathcal{I} \otimes A + \mathcal{B}^T \otimes \mathcal{I}}_{\tilde{A}} + \underbrace{[\mathcal{I} \otimes \mathcal{U}_1 \quad \mathcal{I} \otimes \mathcal{U}_2]}_{\tilde{U}} \underbrace{\begin{bmatrix} \mathcal{R}_1^T \otimes \mathcal{V}_1^T \\ \mathcal{R}_2^T \otimes \mathcal{V}_2^T \end{bmatrix}}_{\tilde{V}^T} \right) \text{vec}(\mathcal{X}) = \text{vec}(\mathcal{E}_1 \mathcal{F}_1^T).$$

According to the Sherman-Morrison-Woodbury formula, we alternatively get

$$\tilde{A} \text{vec}(\mathcal{X}) = \text{vec}(\mathcal{E}_1 \mathcal{F}_1^T) - \underbrace{\tilde{U} (I + \tilde{V}^T \tilde{A}^{-1} \tilde{U})^{-1} \tilde{V}^T \tilde{A}^{-1}}_{\text{vec}(\mathcal{Y})} \text{vec}(\mathcal{E}_1 \mathcal{F}_1^T).$$

Since we have

$$\begin{aligned} \tilde{U} \text{vec}(\mathcal{Y}) &= \tilde{U} \text{vec} \left( \begin{bmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \end{bmatrix} \right) \\ &= [\mathcal{I} \otimes \mathcal{U}_1 \quad \mathcal{I} \otimes \mathcal{U}_2] \text{vec} \left( \begin{bmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \end{bmatrix} \right) \\ &= \text{vec}(\mathcal{U}_1 \mathcal{Y}_1) + \text{vec}(\mathcal{U}_2 \mathcal{Y}_1) \end{aligned}$$

we can conclude that

$$\tilde{U} \operatorname{vec}(\mathcal{Y}) =: \operatorname{vec}(\mathcal{E}_2 \mathcal{F}_2^T),$$

with  $\mathcal{E}_2 \in \mathbb{R}^{n_1 \times (l+m)}$ ,  $\mathcal{F}_2 \in \mathbb{R}^{2n_t \times (l+m)}$ . In particular, this implies

$$\tilde{A} \operatorname{vec}(\mathcal{X}) = \operatorname{vec}(\mathcal{E}_1 \mathcal{F}_1^T) - \operatorname{vec}(\mathcal{E}_2 \mathcal{F}_2^T)$$

or, in other words,  $\mathcal{X}$  can also be derived as the solution to a regular Sylvester equation of the form

$$\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = [\mathcal{E}_1 \quad -\mathcal{E}_2] \begin{bmatrix} \mathcal{F}_1^T \\ \mathcal{F}_2^T \end{bmatrix}.$$

We have now established that the PDE-constrained optimization problem can be written in form of a classical Sylvester equation for which we can use the existence results for a low-rank solution introduced in [35]. Note that we do not claim to actually proceed this way in order to compute the solution matrix  $\mathcal{X}$ . Obviously, determining the intermediate solution  $\operatorname{vec}(\mathcal{Y})$  would be a challenge on its own. The previous steps rather should be understood as a theoretical evidence for the assumption that  $\mathcal{X}$  indeed exhibits a very strong singular value decay. Keep in mind that we had to assume that the desired final state  $Y_{obs}$  as well as  $D$  are of low-rank and that  $l, m \ll n_1$  which is a reasonable assumption for realistic control problems.

**A special case.** One might argue that for the distributed control case, i.e.,  $N$  begin square together with (almost) entirely observable state, i.e.,  $C_{obs} = M_2$ , the previous low-rank assumptions do no longer hold true. Consequently, applying the Sherman-Morrison-Woodbury formula will not simplify (4.2) and we thus will have to deal with a linear matrix equation of the form

$$\sum_{i=1}^4 \mathcal{A}_i \mathcal{X} \mathcal{B}_i = \mathcal{E}_1 \mathcal{F}_1^T, \quad (4.3)$$

where we cannot benefit from additional structure in  $\mathcal{A}_i$  and  $\mathcal{B}_i$ . Still, as has already been (numerically) observed and partially discussed in [10, 11, 21] for the special Lyapunov type case, i.e.,  $\mathcal{B}_i = \mathcal{A}_i^T$ , the solution matrix  $\mathcal{X}$  still seems to exhibit similar low-rank properties.

Although the most general case certainly is an interesting topic of future research, we want to conclude by pointing out that for the special case  $M_2 = M_1 = N = M$  we immediately get an analogous (in fact even stronger) low-rank existence result for (4.2). This is due to the fact that here (4.2) is equivalent to the Sylvester equation

$$L [Y \quad P] \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix} + M [Y \quad P] \begin{bmatrix} \tau D_1 & -C^T \\ -C & -\frac{\tau}{\beta} D_2^{-1} \end{bmatrix} = \begin{bmatrix} \tau M Y_{obs} D_1 \\ D \end{bmatrix}$$

for which we again can apply the low-rank existence results from [35].

## 5. Other State equations.

**Stokes equation.** In addition to the heat equation as a test problem we here also consider the Stokes equation. The discretization of the Stokes control problem can

be performed similarly to the case for the heat equation and we refer the interested reader to [26]. The Stokes equations are given by

$$y_t - \nu \Delta y + \nabla p = u \text{ in } [0, T] \times \Omega \quad (5.1)$$

$$-\nabla \cdot y = 0 \text{ in } [0, T] \times \Omega \quad (5.2)$$

$$y(t, \cdot) = g(t) \text{ on } \partial\Omega, t \in [0, T] \quad (5.3)$$

$$y(0, \cdot) = y^0 \text{ in } \Omega, \quad (5.4)$$

and the functional we are interested in is defined as

$$J(y, u) = \frac{1}{2} \int_0^T \int_{\Omega_1} (y - \bar{y})^2 dxdt + \frac{\beta}{2} \int_0^T \int_{\Omega_2} u^2 dxdt. \quad (5.5)$$

Our goal is to build a discrete Lagrangian using an all-at-once approach [79] where we setup a discrete problem within the space-time cylinder. Using the trapezoidal rule and  $Q2/Q1$  finite elements in space leads to the following discrete objective function

$$J(y, u) = \frac{\tau}{2} (y - y_{obs})^T \mathcal{M}_1 (y - y_{obs}) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u \quad (5.6)$$

with

$$\mathcal{M}_1 = \text{blkdiag} \left( \frac{1}{2} M_1, 0, M_1, 0, \dots, M_1, 0, \frac{1}{2} M_1, 0 \right),$$

$$\mathcal{M}_2 = \text{blkdiag} \left( \frac{1}{2} M_2, M_2, \dots, M_2, \frac{1}{2} M_2 \right)$$

where we reuse the notation for the heat equation. 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.,

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

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

$$\mathcal{K}y - \tau \mathcal{N}u = d \quad (5.7)$$

where

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

Here,

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

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

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

represents the distributed control term control term where  $N = M$ , 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 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. 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 using a Lagrangian with Lagrange multiplier  $p$  lead to the following system

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \mathcal{N}^T \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1 y_{obs} \\ 0 \\ d \end{bmatrix}, \quad (5.8)$$

where again we can switch to a Kronecker structure defined by

$$\begin{bmatrix} D_1 \otimes \tau\mathcal{M} & 0 & -(I_{n_t} \otimes \mathcal{L} + C^T \otimes \mathcal{M}) \\ 0 & D_2 \otimes \beta\tau\mathcal{M}_2 & D_3 \otimes \mathcal{N}^T \\ -(I_{n_t} \otimes \mathcal{L} + C \otimes \mathcal{M}) & D_3 \otimes \mathcal{N} & 0 \end{bmatrix}. \quad (5.9)$$

We can now in a similar way to before use the low-rank MINRES method. Again, we here apply a block-diagonal preconditioner of the form

$$P = \begin{bmatrix} D_1 \otimes \tau\hat{M}_1 & 0 & 0 \\ 0 & D_2 \otimes \beta\tau\hat{M}_2 & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}. \quad (5.10)$$

Here  $\hat{M} = \text{blkdiag}(\hat{M}_1, \gamma I)$ , with  $\gamma = \beta\tau h^d$  (see [79] for details). Here  $d$  is the dimension of the problem ( $d=2,3$ ) and  $h$  the mesh parameter. The matrices  $M_1$  and  $M_2$  are approximated via a Chebychev semi-iteration [32, 33, 83] or in case of lumped mass matrices we trivially have  $\hat{M}_{1,2} = M_{1,2}$ . The approximation of the Schur-complement is much more tricky in this case as for the indefinite  $\mathcal{M}_1$  the Schur-complement is not well-defined. Thus, we again use the approximation  $\hat{M} = \text{blkdiag}(\hat{M}_1, \gamma I)$  to form an approximate Schur-complement

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

with  $\hat{\mathcal{M}}_1$  a block-diagonal involving  $\hat{M}$ . We in turn approximate this via

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

where  $\hat{\mathcal{M}} = \text{blkdiag}\left(\frac{1}{\sqrt{\beta}}M_1, 0, \dots, \frac{1}{\sqrt{\beta}}M_1, 0\right)$  for the distributed control case. As in Section 3 we note that the matrix

$$(\mathcal{K} + \hat{\mathcal{M}}) = (I_{n_t} \otimes \tilde{\mathcal{L}} + C \otimes \mathcal{M})$$

with  $\tilde{\mathcal{L}} = \begin{bmatrix} (\tau^{-1} + \beta^{-1/2})M_1 + K & B^T \\ B & 0 \end{bmatrix}$ . We now proceed in the following way. A stationary iteration scheme with a fixed number of steps is used to approximately solve  $(I_{n_t} \otimes \tilde{\mathcal{L}} + C \otimes \mathcal{M})$  with preconditioner  $(I_{n_t} \otimes \tilde{\mathcal{L}})$  and within this preconditioner systems with  $\tilde{\mathcal{L}}$  are approximately solved using another Uzawa scheme with a fixed but small number of iterations. For this inner Uzawa iteration the inverse of the preconditioner is given by

$$\begin{bmatrix} [(\tau^{-1} + \beta^{-1/2})M_1 + K]_{MG}^{-1} & 0 \\ 0 & (\tau^{-1} + \beta^{-1/2})[K_p]_{MG}^{-1} + [M_p]_{MG}^{-1} \end{bmatrix}$$

where the  $[\dots]_{MG}^{-1}$  indicates the use of a geometric [38, 84] or algebraic multigrid method [66, 28]. Preconditioners of this type are of so-called Cahouet-Chabard [20] and the derivation can be done using a least squares commutator approach [26, 79].

Again, for more robustness more sophisticated Sylvester solvers should be used in the future to guarantee robustness with respect to the system parameters.

**Convection-diffusion equation.** Before coming to the numerical results we quickly want to introduce the last state equation considered here. The PDE constraint is now given by the convection diffusion equation

$$y_t - \varepsilon \Delta y + w \cdot \nabla y = u \text{ in } \Omega \quad (5.11)$$

$$y(:, x) = g \text{ on } \partial\Omega \quad (5.12)$$

$$y(0, :) = y_0 \quad (5.13)$$

as the constraint to the following objective function

$$J(y, u) = \frac{1}{2} \int_0^T \int_{\Omega_1} (y - \bar{y})^2 dxdt + \frac{\beta}{2} \int_0^T \int_{\Omega_2} u^2 dxdt. \quad (5.14)$$

Note that the parameter  $\varepsilon$  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 [64, 40, 59] and for brevity we do not discuss the possible pitfalls regarding the discretization. We here focus on a discretize-then-optimize scheme using a streamline upwind Galerkin (SUPG) approach introduced in [19]. Note that other schemes such as discontinuous Galerkin methods [80] or local projection stabilization [59] are typically more suitable discretizations for the optimal control setup as they often provide the commutation between optimize first or discretize first for the first order conditions. Nevertheless, our approach will also work for these discretizations. Once again we employ a trapezoidal rule in connection with finite elements and now additionally an SUPG stabilization the discretized objective function and state equation are given by

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

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 before

$$\mathcal{K}y - \tau\mathcal{N}u = d \quad (5.15)$$

with

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

Note that due to the SUPG test functions used we now have  $M_s$  which is obtained entry-wise from evaluating the integrals

$$(M_s)_{ij} = \int_{\Omega} \phi_i \phi_j + \delta \int_{\Omega} \phi_i (w \cdot \nabla \phi_j)$$

where  $\phi$  are the finite element test functions and  $\delta$  is a parameter coming from the use of SUPG [19, 26]. We then have  $L_s = M_s + \tau K_s$  where  $K_s$  is the standard nonsymmetric matrix representing the SUPG discretization of the convection-diffusion equation. We can now see that this again is of the desired Kronecker-form

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -(I_{n_t} \otimes L_s + C \otimes M_s)^T \\ 0 & D_2 \otimes \beta \tau M_2 & D_3 \otimes \tau M_s^T \\ -(I_{n_t} \otimes L_s + C \otimes M_s) & D_3 \otimes \tau M_s & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} D_1 \otimes \tau M_1 y_{obs} \\ 0 \\ d \end{bmatrix}. \quad (5.16)$$

Again, we employ the low-rank version of MINRES to solve this system. Note that for nonsymmetric formulations such as the one obtained from an optimize-then-discretize strategy we can also use low-rank versions of nonsymmetric Krylov solvers such as GMRES [69] or BICG [30]. A preconditioner is of the form

$$P = \begin{bmatrix} D_1 \otimes \tau \hat{M}_1 & 0 & 0 \\ 0 & D_2 \otimes \beta \tau \hat{M}_2 & 0 \\ 0 & 0 & \hat{S} \end{bmatrix} \quad (5.17)$$

where the two blocks involving mass matrices are as before and the Schur-complement of  $\mathcal{A}$

$$\begin{aligned} \hat{S} &= (I_{n_t} \otimes L_s + C \otimes M_s) (D_1^{-1} \otimes \tau^{-1} M_1^{-1}) (I_{n_t} \otimes L_s + C \otimes M_s)^T \\ &\quad + (D_3 \otimes \tau M_s) (D_2^{-1} \otimes \beta^{-1} \tau^{-1} M_2^{-1}) (D_3 \otimes \tau M_s^T). \end{aligned} \quad (5.18)$$

The technique introduced in [60] for the steady case can now be extended here as well to give the following Schur-complement approximation

$$\hat{S} = \left( I_{n_t} \otimes \hat{L}_s + C \otimes M_s \right) (D_1^{-1} \otimes \tau^{-1} M_1^{-1}) \left( I_{n_t} \otimes \hat{L}_s + C \otimes M_s \right)^T \quad (5.19)$$

where

$$\hat{L}_s = \left( 1 + \frac{1}{\sqrt{\beta}} \right) M_s + \tau K_s.$$

As with the previous problems a multigrid approximation [63] is suitable for the evaluation of the inverse of  $L_s$ . These preconditioners need to be adapted when other discretizations are used but we expect our results to carry over to these cases as well. We again use a stationary iteration to approximate the parts of the Schur-complement and a more sophisticated solver is recommended for the future.

**6. Numerical Results.** We are now going to present results for the low-rank in time solution of certain PDE-constrained optimization problems. The results presented in this section are based on an implementation of the above described algorithms within MATLAB<sup>®</sup> whereas we perform the discretization of the PDE-operators within the deal.II [7] framework using  $Q1$  finite elements for the heat equation and convection diffusion equation. The Stokes equation is discretized with  $Q2$  elements for the velocity and  $Q1$  elements for the pressure component. For the algebraic multigrid approximation, we used HSL ML\_20 [18]. For some preconditioner we used backslash within MATLAB<sup>®</sup> for the innermost solution within the preconditioner. Our implementation of MINRES is based on a vector version presented in [26] and was stopped with a tolerance of  $10^{-4}$  for the relative pseudo-residual. Our experiments are performed for a final time  $T = 1$  with a varying number of time-steps. As the domain  $\Omega$  we consider the unit cube but other domains are of course possible. We specify the boundary conditions for each problem separately. Throughout the results section we fixed the truncation at  $10^{-8}$  for which we observed good results. Additionally, we also performed not listed experiments with a tolerance of  $10^{-10}$  for which we also observed fast convergence. Larger tolerances should be combined with a deeper analysis of the algorithms and a combination with flexible outer solvers. All results are performed on a standard Ubuntu desktop Linux machine with Intel(R) Xeon(R) CPU W3503 @ 2.40GHz and 6GB of RAM.

### 6.1. The heat equation.

**Distributed Control.** As the first example shown in this section we use the heat equation with a distributed control term. We choose the boundary conditions for this problem to be of zero Dirichlet type. We first show how well the desired state

$$y_{obs} = -64 \exp\left(-\left((x_0 - 0.5t)^2 + (x_1 - 0.5t)^2\right)\right)$$

is approximated in low-rank form. Figure 6.1 illustrates this for grid point 10 in time where the right hand side  $\text{vec}^{-1}(\tau \mathcal{M}_1 y_{obs}) = B_{11} B_{12}^T$  is approximated by low-rank factors of rank 2.

Table 6.1 shows first results for the comparison of the full-rank MINRES versus the low-rank version. We want to point out that here we use the backslash operator in MATLAB<sup>®</sup> to evaluate the matrix  $L$  within the preconditioner but this can easily be replaced by a multigrid approximation and in fact is done later. Before interpreting the results it has to be noted that the comparison is not entirely fair as due to the poor performance within MATLAB<sup>®</sup> using for loops, we implemented both the preconditioner and the matrix-vector multiplication in a matrix way as shown in (3.2). For simplicity we only used a block-diagonal approximation for the matrix  $\mathcal{K}$  and hence the scaling with respect to changes in the number of time-steps is not given. This would typically be the case and our results using deal.II and C++ in [78, 58] indicate robustness with respect to the number of time-steps. Nevertheless, every increase in the number of time-steps also results in an increase in the matrix size and so one would expect when the number of time-steps is increased five-fold that the same

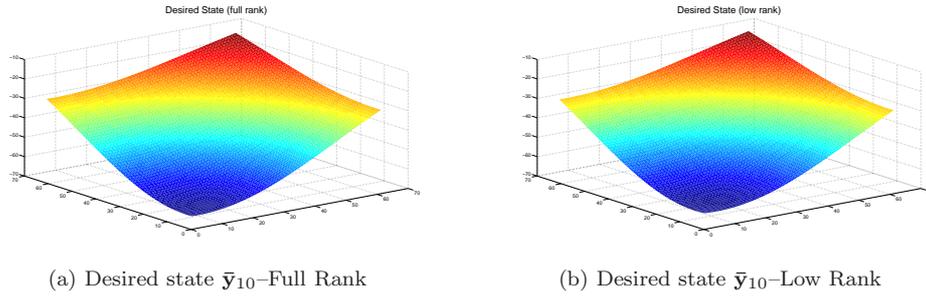


Fig. 6.1: Desired state in full-rank and low-rank form.

happens for the time needed to solve the linear system. Going back to the results in Table 6.1 where both the timings and iteration numbers are shown for a variety of mesh-sizes and two different orders of grid points in time, that both methods perform mesh-independent and that we can see that the low-rank method shows almost no increase when the number of time-steps is drastically increased. Note also the degrees of freedom given here are only for the spatial discretization. The overall dimension of the linear system is then given by  $3nn_t$  where  $n$  represents the spatial degrees of freedom. We see that the iteration times for the full rank solver go up and using the non-optimal preconditioners we additionally see that the times increase more than just by a factor of five. We also see that due to the cost of performing a low-rank truncation the full-rank method always outperforms the low-rank scheme for a small number of time steps. Nevertheless, the low-rank method can easily solve problems that are no longer tractable for full-rank methods.

DoF	FR (20) # it(t)	FR (100) # it(t)	LR (20) # it(t)	LR (100) # it(t)
289	17(0.2)	31(1.1)	15(5.6)	19(7.6)
1089	19(0.7)	33(5.56)	17(8.9)	21(11.9)
4225	19(3.5)	35(26.6)	17(19.7)	23(26.3)
16641	21(17.5)	35(125.8)	19(72.1)	23(97.6)
66049	23(81.8)	OoM	19(324.8)	25(427.4)

Table 6.1: Results for full-rank (FR) MINRES vs. low-rank (LR) MINRES for 20 or 100 time-steps and a variety of different meshes. Both iteration numbers and computing times in seconds are listed. OoM indicates Out of Memory in MATLAB<sup>®</sup>.

Next we compare how both the full-rank and the low-rank method perform when the number of time-steps is further increased. We therefore consider a fixed mesh for a varying time-discretization. Table 6.2 shows the results for both the full-rank and the low-rank method. We additionally show the rank of the three components of the state, control, and adjoint state. We started computing the truncation process using a maximum size of the truncated SVD of 20, which was sufficient for all discretizations in time using a truncation tolerance of  $10^{-8}$ . In order to keep the iteration numbers from growing too much with an increase in the number of time-steps we increased

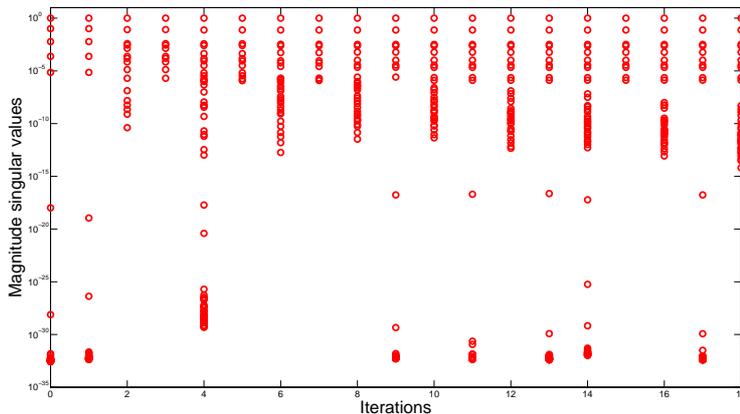


Fig. 6.2: Singular values of the approximate solution during the iteration before truncation.

the number of stationary iterations for the preconditioner from 2 to 3 for the last two columns in Table 6.2. We see again that the full-rank method exceeds the memory limit in MATLAB<sup>®</sup>. It can also be seen that the increase in rank and computing time is typically moderate. Note that the system dimension considering a full-rank solution is ranging from 998460 to 29953800 unknowns.

DoF	20	100	200	400	600
16641	# it(t)				
LR	19(108.2)	21(307.8)	25(432.7)	43(671.9)	61(937.3)
(ranks)	8/10/10	10/11/11	12/13/13	11/14/14	14/15/15
FR	21(18.3)	35(124.0)	63(434.3)	OoM	OoM

Table 6.2: Results for full-rank (FR) MINRES vs. low-rank (LR) MINRES for a fixed mesh with 16641 unknowns in space. We show varying time-steps and additionally the rank of the state/control/adjoint state. Both iteration numbers and computing times in seconds are listed. OoM indicates Out of Memory in MATLAB<sup>®</sup>. Results are shown for  $\beta = 10^{-4}$ .

In order to illustrate the distribution of the singular values we show in Figure 6.2 how the relative value of the singular values behaves throughout the iteration. Shown are the scaled singular values ( $\sigma_j/\sigma_1$ ) of the approximation to the state for the problem with 4225 unknowns and 100 grid-points in time. So far the preconditioners introduced have used a direct solver for the solution of the systems with

$$\hat{L} = \left( \left( 1 + \frac{\tau}{\sqrt{\beta}} \right) M_1 + \tau K \right)$$

both in the full-rank method and the low-rank one. We now illustrate that we can easily approximate this matrix using an algebraic multigrid technique by also showing

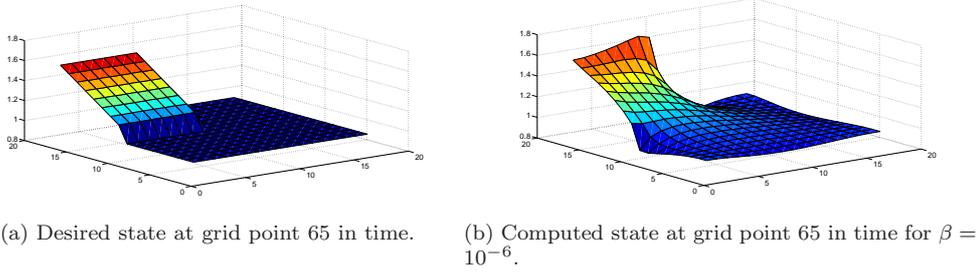


Fig. 6.3: Desired state and computed state for a boundary control problem.

that our preconditioner performs robustly with respect to the regularization parameter  $\beta$ . We here compute the truncated singular value decomposition up to the order 20 and then cut off corresponding to the truncation tolerance. We additionally increased the number of stationary iteration steps for the matrix  $(I_{n_t} \otimes \hat{L} + C \otimes M)$  with preconditioner  $(I_{n_t} \otimes \hat{L})$  to 4.

DoF	1089 (100)	4225 (100)	16641 (100)
$\beta$	# it(t)	# it(t)	# it(t)
$10^{-2}$	37(79.1)	37(226.1)	39(943.7)
$10^{-4}$	17(36.6)	19(111.6)	19(448.2)
$10^{-6}$	17(34.7)	17(94.4)	19(406.7)

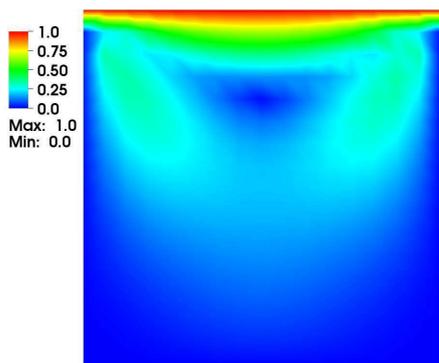
Table 6.3: Results for low-rank (LR) MINRES with 100 time-steps and a varying regularization parameter on three different meshes. Both iteration numbers and computing times in seconds are listed.

**Boundary Control.** In the following we quickly demonstrate that our approach also works for the case of a boundary control problem. The desired state is shown in Figure 6.3a and the computed state wanting to approximate this in Figure 6.3b. In Table 6.4 we show results for the low-rank MINRES approximation for a variety of mesh-parameters and regularization parameters. Details on the preconditioners used can be found in [58]. As in the last example for the distributed control case we choose 4 Uzawa iterations and a tolerance of  $10^{-4}$  for the iterative solver. We here evaluate  $\hat{L}$  again using the backslash operator in MATLAB<sup>®</sup> but the use of AMG is straightforward.

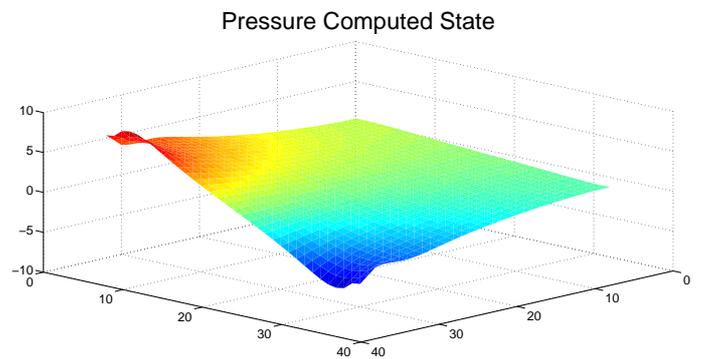
**6.2. Stokes equation.** The configuration for the Stokes equation is taken from [79] and originally appeared in [43]. The spatial domain is the unit cube  $\Omega = [0, 1]^d$  with a time domain  $[0, 1]$ . The target flow is the solution for an unsteady Stokes equation with Dirichlet boundary conditions, i.e.  $y = (1, 0)$  when the second spatial component  $x_2 = 1$  and  $y = (0, 0)$  on the remaining boundary for the two-dimensional case. For the control problem we now consider the following time-dependent boundary conditions. For the top-boundary where  $x_2 = 1$  we get  $y = (1 + \frac{1}{2} \cos(4\pi t - \pi), 0)$  and

DoF	289 (100)	4225 (100)	16641 (100)
$\beta$	# it(t)	# it(t)	# it(t)
$10^{-2}$	49(137.3)	61(236.18)	79(802.7)
$10^{-4}$	67(179.8)	99(406.6)	151(1510.6)
$10^{-6}$	63(169.2)	95(380.4)	147(1448.6)

Table 6.4: Results for low-rank (LR) MINRES with 100 time-steps and a varying regularization parameter on three different meshes for a boundary control example. Both iteration numbers and computing times in seconds are listed.



(a) Desired state at grid point 10 in time.



(b) Pressure of the computed state for  $\beta = 1e - 1$ .

Fig. 6.4: Desired state and computed pressure for the Stokes flow problem.

zero elsewhere in two space dimensions and we set viscosity to  $1/100$ . For the results shown in Table 6.5 we note that we needed to set the number of stationary iteration steps for both the outer-iteration to 30 as well as the inner one for the small saddle point system to 5. We believe that the outer iteration can be replaced by a robust Sylvester solver.

DoF	578+81 (100)	2178+289 (100)	8450+1089 (100)
$\beta$	# it(t)	# it(t)	# it(t)
$10^{-1}$	11(224.4)	12(624.8)	14(3601.9)
$10^{-5}$	15(290.2)	15(737.6)	17(4091.5)

Table 6.5: Results for low-rank (LR) MINRES with 100 time-steps and a varying regularization parameter on three different meshes for a Stokes control example. Both iteration numbers and computing times in seconds are listed.

Apart from the approximation of the Neumann-Laplacian on the pressure space whose inverse was evaluated using an algebraic multigrid scheme we simply used

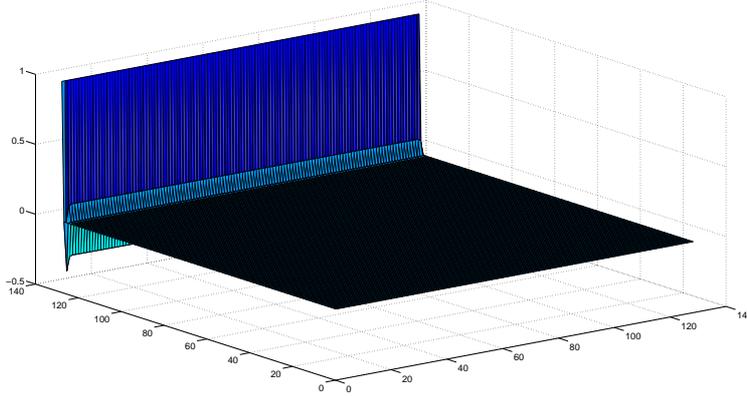


Fig. 6.5: Computed state for  $\beta = 10^{-6}$  at grid point 10 in time.

the backslash operator to evaluate the remaining components. A further increase in computational efficiency can be gained when these are replaced by multigrid approximations.

**6.3. Convection diffusion equation.** The configuration for the convection-diffusion equation is taken from [26] and is typically referred to as the double glazing problem. The spatial domain is the unit cube  $\Omega = [-1, 1]^2$  with a time domain  $[0, 1]$ . The wind  $w$  is given by

$$w = (2y(1 - x^2), -2x(1 - y^2))$$

We here set the parameter to  $\varepsilon$  to  $1/200$  and the boundary condition is a Dirichlet zero condition with the exception of  $y = 1$  when  $x_2 = 1$ . The desired state is set to zero throughout the domain [59]. In Figure 6.5 we show the computed state for grid point 10 in time. Due to the nonsymmetric nature of the PDE-operator we have not employed the recommended multigrid technique [63] and simply used the backslash operator here. The results shown in Table 6.6 indicate a robust performance of the low-rank MINRES method. We here set the number of stationary iterations to 15.

DoF	1089 (100)	4225 (100)	16641 (100)	4225 (200)	16641 (200)
$\beta$	# it(t)	# it(t)	# it(t)	# it(t)	# it(t)
$10^{-2}$	18(58.2)	18(195.5)	18(1278.1)	32(371.6)	32(2266.5)
$10^{-5}$	8(26.2)	10(95.3)	12(820.7)	10(115.4)	12(858.8)

Table 6.6: Results for low-rank (LR) MINRES with 100 or 200 time-steps and a varying regularization parameter on a variety of meshes for a convection-diffusion control example. Both iteration numbers and computing times in seconds are listed.

**7. Outlook.** We believe that the research presented here opens some interesting angles that should be studied in the future. The incorporation of additional

constraints such as control and state constraints is typically very important for real-world scenarios. We plan to investigate a technique introduced in [42] where the state and adjoint state are computed first and hence amenable to low-rank techniques and then the constrained control is computed. It is further desired to investigate more complicated discretizations in time. Of particular interest we want to study backward differentiation formulas (BDF) [5] as these can be easily incorporated simply modifying the  $C$  matrix in (3.1). We further plan to incorporate more sophisticated generalized Sylvester equation solvers for  $(I_{n_t} \otimes L + C \otimes M)$ , which we believe allows for more robustness with respect to the system parameters and should be combined with a flexible outer method [68]. It is further crucial to investigate how the low-rank techniques can be extended to incorporate non-linearities of both the objective function and the PDE-constraint such as [17].

**8. Conclusions.** In this paper we proposed the use of a low-rank methodology for the solution to PDE-constrained optimization problems. We in particular introduced a low-rank in time approach that allows to significantly reduce the storage requirements in time for a one-shot solution of the optimal control problem. We were also able to rewrite the problem in such a way that we can obtain low-rank existence results from classical Sylvester equation theory. We additionally discussed a stationary iteration as a preconditioner for the Schur-complement approximation within the overall block-diagonal preconditioner. We further illustrated that this technique can be used for many well-known PDEs. Our numerical results illustrated that even with the rather crude Schur-complement approximation a rather robust performance could be obtained. The low-rank method presented enabled computations that are no longer possible to perform with the full-rank approach, which we see as a crucial feature of our methodology.

#### REFERENCES

- [1] ROMAN ANDREEV, *Space-time discretization of the heat equation. a concise matlab implementation*, arXiv preprint arXiv:1212.6037, (2012).
- [2] ROMAN ANDREEV AND CHRISTINE 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] A.C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005.
- [4] A.C. ANTOULAS, D.C. SORENSEN, AND Y. ZHOU, *On the decay rate of Hankel singular values and related issues*, Systems & Control Letters, 46 (2002), pp. 323–342.
- [5] URI M ASCHER, ROBERT MM MATTHEIJ, AND ROBERT D RUSSELL, *Numerical solution of boundary value problems for ordinary differential equations*, vol. 13, Society for Industrial and Applied Mathematics, 1987.
- [6] JAMES BAGLAMA AND LOTHAR REICHEL, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput., 27 (2005), pp. 19–42.
- [7] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II—a general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. Art. 24, 27.
- [8] U. BAUR AND P. BENNER, *Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic*, 78 (2006).
- [9] P. BENNER AND T. BREITEN, *On optimality of interpolation-based low-rank approximations of large-scale matrix equations*, MPI Magdeburg Preprints MPIMD/11-10, 2011.
- [10] ———, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numerische Mathematik, 124 (2013), pp. 441–470.
- [11] P. BENNER AND T. DAMM, *Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems*, SIAM Journal on Control and Optimization, 49 (2011), pp. 686–711.

- [12] PETER BENNER AND PATRICK KÜRSCHNER, *Computing real low-rank solutions of Sylvester equations by the factored ADI method*, MPI Magdeburg Preprint MPIMD/13-05, May 2013.
- [13] P. BENNER, J.-R. LI, AND T. PENZL, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numerical Linear Algebra with Applications, 15 (2008), pp. 755–777. Based on an unpublished manuscript from 1999.
- [14] P. BENNER, R.C. LI, AND N. TRUHAR, *On the ADI method for Sylvester equations*, Journal of Computational and Applied Mathematics, 233 (2009), pp. 1035–1045.
- [15] P. BENNER AND E.S. QUINTANA-ORTÍ, *Solving stable generalized Lyapunov equations with the matrix sign function*, 20 (1999), pp. 75–100.
- [16] MICHELE BENZI, GENE H. GOLUB, AND JÖRG LIESEN, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.
- [17] 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.
- [18] J. BOYLE, M. D. MIHAJLOVIC, AND J. A. SCOTT, *HSL\_MI20: an efficient AMG preconditioner*, Tech. Report RAL-TR-2007-021, Rutherford Appleton Laboratory (CCLRC), 2007.
- [19] ALEXANDER N BROOKS AND THOMAS JR 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.
- [20] J. CAHOUET AND J.P. CHABARD, *Some fast 3D finite element solvers for the generalized Stokes problem*, International Journal for Numerical Methods in Fluids, 8 (1988), pp. 869–895.
- [21] T. DAMM, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numerical Linear Algebra with Applications, 15 (2008), pp. 853–871.
- [22] T.A. DAVIS, *Umfpack version 4.4 user guide*, tech. report, Dept. of Computer and Information Science and Engineering Univ. of Florida, Gainesville, FL, 2005.
- [23] SERGEY DOLGOV, BORIS N KHOROMSKIJ, IVAN OSELEDETS, AND EUGENE TYRTYSHNIKOV, *A reciprocal preconditioner for structured matrices arising from elliptic problems with jumping coefficients*, Linear Algebra and its Applications, 436 (2012), pp. 2980–3007.
- [24] V. DRUSKIN, L. KNIZHNERMAN, AND V. SIMONCINI, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1875–1898.
- [25] 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.
- [26] HOWARD C. ELMAN, DAVID J. SILVESTER, AND ANDREW 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.
- [27] A. EPPLER AND M. BOLLHÖFER, *A structure preserving FGMRES method for solving large Lyapunov equations*, in Progress in Industrial Mathematics at ECMI 2010, M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel, eds., vol. 17 of Mathematics in Industry, Springer Verlag, Berlin/Heidelberg, 2012, pp. 131–136.
- [28] R.D. FALGOUT, *An Introduction to Algebraic Multigrid*, Computing in Science and Engineering, 8 (2006), pp. 24–33. Special Issue on Multigrid Computing.
- [29] G. FLAGG AND S. GUGERCIN, *On the ADI method for the Sylvester equation and the optimal- $\mathcal{H}_2$  points*, Applied Numerical Mathematics, 64 (2013), pp. 50–58.
- [30] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, pp. 73–89. Lecture Notes in Math., Vol. 506.
- [31] GENE H. GOLUB AND CHARLES F. VAN LOAN, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.
- [32] GENE H. GOLUB AND RICHARD S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I*, Numer. Math., 3 (1961), pp. 147–156.
- [33] ———, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II*, Numer. Math., 3 (1961), pp. 157–168.
- [34] L. GRASEDYCK, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.
- [35] L. GRASEDYCK, *Existence of a low rank or -matrix approximant to the solution of a sylvester equation*, Numerical Linear Algebra with Applications, 11 (2004), pp. 371–389.
- [36] L. GRASEDYCK AND W. HACKBUSCH, *A multigrid method to solve large scale sylvester equations*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 870–894.

- [37] A. GRIEWANK AND A. WALTHER, *Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation*, ACM Transactions on Mathematical Software (TOMS), 26 (2000), pp. 19–45.
- [38] WOLFGANG HACKBUSCH, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.
- [39] MATTHIAS 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.
- [40] MATTHIAS HEINKENSCHLOSS AND DMITRIY 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.
- [41] D. HINRICHSSEN AND A.J. PRITCHARD, *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, vol. 1, Springer Verlag, 2005.
- [42] M. HINZE, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, Comput. Optim. Appl., 30 (2005), pp. 45–61.
- [43] MICHAEL HINZE, MICHAEL KÖSTER, AND STEFAN TUREK, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, tech. report, SPP1253-16-01, 2008.
- [44] M. HINZE, R. PINNAU, M. ULBRICH, AND S. ULBRICH, *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.
- [45] MICHIEL E. HOCHSTENBACH, *A Jacobi-Davidson type SVD method*, SIAM J. Sci. Comput., 23 (2001), pp. 606–628. Copper Mountain Conference (2000).
- [46] KAZUFUMI ITO AND KARL 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.
- [47] I.M. JAÏMOUKHA AND E.M. KASENALLY, *Krylov subspace methods for solving large Lyapunov equations*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 227–251.
- [48] K. JBILOU AND A. J. RIQUET, *Projection methods for large Lyapunov matrix equations*, Linear Algebra and its Applications, 415 (2006), pp. 344–358.
- [49] H.W. KNOBLOCH AND H. KWAKERNAAK, *Lineare Kontrolltheorie*, Springer-Verlag, Berlin, 1985. In German.
- [50] MARKUS KOLLMANN AND MICHAEL KOLMBAUER, *A Preconditioned MinRes Solver for Time-Periodic Parabolic Optimal Control Problems*, Submitted, Numa-Report 2011-06, (August 2011).
- [51] D. KRESSNER AND C. TOBLER, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714.
- [52] D. KRESSNER AND C. TOBLER, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1288–1316.
- [53] J.-R. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM Journal on Matrix Analysis and Applications, 24 (2002), pp. 260–280.
- [54] TAREK P. MATHEW, MARCUS SARKIS, AND CHRISTIAN E. SCHAEERER, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM J. Sci. Comput., 32 (2010), pp. 1180–1200.
- [55] JORGE NOCEDAL AND STEPHEN J. WRIGHT, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.
- [56] C. C. PAIGE AND M. A. SAUNDERS, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
- [57] JOHN W. PEARSON, MARTIN STOLL, AND ANDY WATHEN, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, To Appear Numerical Linear Algebra with Applications, (2012).
- [58] JOHN W. PEARSON, MARTIN STOLL, AND ANDREW J. WATHEN, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.
- [59] J. W. PEARSON AND A. J. WATHEN, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Submitted, (2011).
- [60] JOHN W. PEARSON AND ANDREW 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.
- [61] T. PENZL, *A cyclic low rank Smith method for large, sparse Lyapunov equations with applications in model reduction and optimal control*, Tech. Report SFB393/98-6, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, FRG, 1998. Available from <http://www.tu-chemnitz.de/sfb393/sfb98pr.html>.

- [62] ———, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems & Control Letters, 40 (2000), pp. 139–144.
- [63] A RAMAGE, *A multigrid preconditioner for stabilised discretisations of advection–diffusion problems*, Journal of computational and applied mathematics, 110 (1999), pp. 187–203.
- [64] TYRONE REES, *Preconditioning Iterative Methods for PDE Constrained Optimization*, PhD thesis, University of Oxford, 2010.
- [65] TYRONE REES, H. SUE DOLLAR, AND ANDREW J. WATHEN, *Optimal solvers for PDE-constrained optimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 271–298.
- [66] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.
- [67] Y. SAAD, *Numerical solution of large Lyapunov equation*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhauser, 1990, pp. 503–511.
- [68] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific Computing, 14 (1993), pp. 461–461.
- [69] YOUSEF SAAD AND MARTIN H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput, 7 (1986), pp. 856–869.
- [70] JOACHIM SCHÖBERL AND WALTER ZULEHNER, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.
- [71] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1268–1288.
- [72] ———, *The extended Krylov subspace for parameter dependent systems*, Applied Num. Math, 60 (2010), pp. 550–560.
- [73] V. SIMONCINI, *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems*, To appear in Computational Optimization and Applications, (2012).
- [74] V. SIMONCINI, *Computational methods for linear matrix equations*, tech. report, 2013.
- [75] D.C. SORENSEN AND A.C. ANTOULAS, *The Sylvester equation and approximate balanced reduction*, Linear Algebra and its Applications, 351–352 (2002), pp. 671–700.
- [76] MARTIN STOLL, *A Krylov-Schur approach to the truncated SVD*, Linear Algebra and its Applications, 436 (2012), pp. 2795–2806.
- [77] MARTIN STOLL, *All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, Accepted IMA Journal of numerical analysis, (2013).
- [78] MARTIN STOLL AND ANDY WATHEN, *All-at-once solution of time-dependent PDE-constrained optimization problems*, tech. report, University of Oxford, 2010.
- [79] ———, *All-at-once solution of time-dependent Stokes control*, Journal of Computational Physics, 232 (2013), pp. 498–515.
- [80] TONGJUN 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.
- [81] F. TRÖLTZSCH, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Amer Mathematical Society, 2010.
- [82] B. VANDEREYCKEN AND S. VANDEWALLE, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2553–2579.
- [83] ANDREW J. WATHEN AND TYRONE REES, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electronic Transactions in Numerical Analysis, 34 (2008), pp. 125–135.
- [84] PIETER WESSELING, *An introduction to multigrid methods*, Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.