ALL-AT-ONCE SOLUTION OF TIME-DEPENDENT PDE-CONSTRAINED OPTIMIZATION PROBLEMS

MARTIN STOLL[†] AND ANDY WATHEN[*]

Abstract. Time-dependent partial differential equations (PDEs) play an important role in applied mathematics and many other areas of science. One-shot methods try to compute the solution to these problems in a single iteration that solves for all time-steps at the same time. In this paper, we look at one-shot approaches for the optimal control of time-dependent PDEs and focus on the fast solution of these problems. The use of Krylov subspace solvers together with an efficient preconditioner allows for minimal storage requirements. We solve only approximate time-evolutions for both forward and adjoint problem and compute accurate solutions of a given control problem only at convergence of the overall Krylov subspace iteration. We show that our approach can give competitive results for a variety of problem formulations.

AMS subject classifications. Primary 65F10, 65N22, 65F50 Secondary 76D07

Key words. PDE-constrained optimization, Saddle point systems, Time-dependent PDE-constrained optimization, Preconditioning, Krylov subspace solver

1. Introduction. Optimization problems with constraints given by time-dependent partial differential equations (PDEs) arise in a variety of applications (see for example [16]). Throughout this paper, we consider minimization of a functional $J(y,u)$ (see Section 2) defined on $\Omega \times [0,T]$ where $\Omega \subset \mathbb{R}^d$ subject to a time-dependent state equation (PDE) that links the control u and the state y.

We will show that discretization of the problem and solution via stationarity (first order optimality) condition on a Lagrangian leads to a linear system in saddle point form

$$\underbrace{\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}}_{\mathcal{A}} \quad x = b \tag{1.1}$$

where $A \in \mathbb{R}^{n,n}$ is symmetric and positive definite or positive semi-definite and $B \in \mathbb{R}^{m,n}$, $m < n$, is a matrix of full rank. We assume that the linear system given in (1.1) is well defined and has a unique solution. This is guaranteed if the block A is positive definite on the kernel of B [2]. The system matrix $\mathcal{A}$ is symmetric and indefinite and a variety of methods exists to solve problems of this type efficiently (see [2] for a survey). In practice, the linear system $\mathcal{A}x = b$ usually is of sufficiently high dimension, when a one-shot approach for a time-dependent problem is considered that iterative solution methods are needed; it is never solved without the application of a preconditioner $\mathcal{P}$ chosen to enhance the convergence behavior of the iterative method. The aim of this paper is to describe a suitable one-shot formulation of the optimization problem and then to present appropriate solvers and preconditioners that allow a fast solution of the time-dependent problem. A similar approach was recently presented in [3].

The paper is organized as follows. The problem we are interested in will be presented in detail in Section 2. Our focus in this paper is to derive efficient preconditioners for the optimal control problems and hence we will introduce all methods from

[*]Numerical Analysis Group, Mathematical Institute, 24 - 29 St Giles', Oxford, OX1 3LB, United Kingdom (wathen@maths.ox.ac.uk)

[†]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)

a linear algebra perspective. In Section 3 we introduce the important situation where the desired state is only defined on a subset of the spatial domain. For simplicity this will be done in the time-independent case. Preconditioners for the time-dependent case are introduced in Section 4 and in Section 5 we present some analysis of our approach. The numerical results presented in Section 6 illustrate the performance of the presented method.

2. The problem and discretization. Throughout this paper, we are focusing on two different functionals given by

$$J_1(y,u) := \frac{1}{2}\int_{\Omega_1} \left(\mathrm{y(x,T)} - \bar{\mathrm{y}}\mathrm{(x)}\right)^2 dx + \frac{\beta}{2}\int_0^T \int_{\Omega_2} \left(\mathrm{u(x,t)}\right)^2 dxdt \tag{2.1}$$

and

$$J_2(y,u) := \frac{1}{2}\int_0^T \int_{\Omega_1} \left(\mathrm{y(x,t)} - \bar{\mathrm{y}}\mathrm{(x,t)}\right)^2 dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} \left(\mathrm{u(x,t)}\right)^2 dxdt. \tag{2.2}$$

Here T is the final time and we have a given desired state $\bar{\mathrm{y}}$ appropriately chosen for each problem. The goal of the optimization process is to drive the state y as close as possible to the desired state using the control u. Note, that both Ω_1 and Ω_2 are subdomains of a domain $\Omega \in \mathbb{R}^d$. We will frequently use $\Omega_2 = \Omega$ and occasionally $\Omega_1 \subsetneq \Omega$. For the remainder of this Section we will assume $\Omega_1 = \Omega_2 = \Omega$ and come back to the case $\Omega_1 \subsetneq \Omega$ in Section 3.

We consider the following parabolic PDE for the course of this paper

$$y_t - \triangle y = u \tag{2.3}$$

in $\Omega \times [0,T]$, with boundary conditions $y = 0$ on the spatial boundary $\partial\Omega$ and initial condition $y(x,0) = y_0(x)$. We also need the adjoint of this PDE, which for (2.1) is given by

$$\begin{aligned} -p_t - \triangle p &= 0 \\ p &= 0 \text{ on } \partial\Omega \\ p(x,T) &= \chi_{\Omega_1}(y(x,T) - \bar{y}(x)) \end{aligned} \tag{2.4}$$

and for (2.2) we have

$$\begin{aligned} -p_t - \triangle p &= \chi_{\Omega_1}(y - \bar{y}) \\ p &= 0 \text{ on } \partial\Omega \\ p(x,T) &= 0, \end{aligned} \tag{2.5}$$

where χ_{Ω_1} is an indicator function for the domain Ω_1 (see Chapter 3.6.4 in [32] for more details). In our approach we will follow a *discretize-then-optimize* strategy. We begin by discretizing the functional given in (2.1) and (2.2) using a standard finite-element approach in space. The choice of the discretization scheme in time will have a significant influence on whether the adjoint PDE is properly reflected in the discretize-then-optimize approach. The final time functional (2.1) then becomes

$$J_1(y,u) = \frac{1}{2}(\mathbf{y}^{(T)} - \bar{\mathbf{y}}^{(T)})^T \tilde{\mathcal{M}}_1 (\mathbf{y}^{(T)} - \bar{\mathbf{y}}^{(T)}) + \frac{\beta\tau}{2}\mathbf{u}^T \mathcal{M}_{1/2}\mathbf{u} \tag{2.6}$$

using the trapezoidal rule and

$$\tilde{\mathcal{M}}_1 = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & \tilde{M} \end{bmatrix} \text{ and } \mathcal{M}_{1/2} = \begin{bmatrix} \frac{1}{2}M & & & \\ & M & & \\ & & \ddots & \\ & & & \frac{1}{2}M \end{bmatrix}. \tag{2.7}$$

Here, $\mathbf{y}^{(T)} = [0, 0, \ldots, 0, \mathbf{y}_N^T]^T$, $\bar{\mathbf{y}}^{(T)} = [0, 0, \ldots, 0, \bar{\mathbf{y}}_N^T]^T$, $\mathbf{u} = [\mathbf{u}_1^T, \boldsymbol{p}_2^T, \ldots, \mathbf{u}_N^T]^T$ are the state and desired state at the last time step and the control at time-steps 1 to N of a backward Euler scheme (see below). If we instead use the (left) rectangular rule for the integral involving the control then the functional becomes

$$J_1(y, u) = \frac{1}{2}(\mathbf{y}^{(T)} - \bar{\mathbf{y}}^{(T)})^T \tilde{\mathcal{M}}_1 (\mathbf{y}^{(T)} - \bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}\mathcal{M}_0\mathbf{u}, \tag{2.8}$$

where $\mathcal{M}_0 = \text{blkdiag}(M, \ldots, M, 0)$. The discretization of the functional (2.2) can be done in a similar way, i.e., the trapezoidal rule for (2.2) leads to

$$J_2(y, u) = \frac{\tau}{2}(\mathbf{y} - \bar{\mathbf{y}})^T \tilde{\mathcal{M}}_{2,1/2} (\mathbf{y} - \bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}\mathcal{M}_{1/2}\mathbf{u} \tag{2.9}$$

where

$$\tilde{\mathcal{M}}_{2,1/2} = \begin{bmatrix} \frac{1}{2}\tilde{M} & & & & \\ & \tilde{M} & & & \\ & & \ddots & & \\ & & & \tilde{M} & \\ & & & & \frac{1}{2}\tilde{M} \end{bmatrix} \tag{2.10}$$

and the rectangular rule gives

$$J_2(y, u) = \frac{\tau}{2}(\mathbf{y} - \bar{\mathbf{y}})^T \tilde{\mathcal{M}}_{2,0} (\mathbf{y} - \bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}\mathcal{M}_0\mathbf{u}, \tag{2.11}$$

with $\tilde{\mathcal{M}}_{2,0} = \text{blkdiag}(\tilde{M}, \tilde{M}, \ldots, \tilde{M}, 0)$. Here, $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_N^T]^T$, $\boldsymbol{p} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \ldots, \boldsymbol{p}_N^T]^T$, $\boldsymbol{p}^{(T)} = [0, 0, \ldots, 0, \boldsymbol{p}_N^T]^T$ are the state and adjoint state (Lagrange multiplier) at time-steps 1 to N of a backward Euler scheme (see below). Further, $\tilde{M}$ represents a (lumped) mass matrix for our choice of finite elements on Ω_1 and M is the corresponding (lumped) mass matrix on Ω_2. Note that we use lumped mass matrices for convenience only; we believe that everything derived in this paper also holds for consistent mass matrices. We will come back to the discrete objective function after the discretization of the PDE. We are performing a time discretization of the PDE (2.3) using a backward Euler discretization

$$\frac{y^k - y^{k-1}}{\tau} - \triangle y^k = u^k \tag{2.12}$$

with time step τ and the finite-element discretization of the weak form then gives

$$M\mathbf{y}_k + \tau K\mathbf{y}_k = M\mathbf{y}_{k-1} + \tau M\mathbf{u}_k. \tag{2.13}$$

K is the stiffness matrix, i.e., the discrete negative Laplacian in the finite element context. So putting all of Equation (2.13) together, the one-shot discretization for N time-steps becomes

$$\underbrace{\begin{bmatrix} M+\tau K & & & & & \\ -M & M+\tau K & & & & \\ & -M & M+\tau K & & & \\ & & \ddots & \ddots & & \\ & & & -M & M+\tau K \end{bmatrix}}_{\mathcal{K}} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} - \tau\mathcal{M}\mathbf{u} = \begin{bmatrix} M\mathbf{y}_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.14}$$

With this, the Lagrangian for the functional $J_1(y,u)$ using the trapezoidal rule is given by

$$\frac{1}{2}(\mathbf{y}^{(T)} - \bar{\mathbf{y}}^{(T)})^T \tilde{\mathcal{M}}_1(\mathbf{y}^{(T)} - \bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}\mathcal{M}_{1/2}\mathbf{u} + \boldsymbol{p}^T(-\mathcal{K}\mathbf{y} + \tau\mathcal{M}\mathbf{u} + d), \tag{2.15}$$

with

$$d = \begin{bmatrix} M\mathbf{y}_0 + c \\ c \\ \vdots \\ c \end{bmatrix}$$

where c represents the boundary conditions of the PDE. The optimality system for this can be written as

$$\begin{bmatrix} \tilde{\mathcal{M}}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{M}}_1\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix}. \tag{2.16}$$

Please note that commonly the first equation in (2.16) is referred to as the adjoint equation, for the simple reason as it represents a discretization of the adjoint PDE. Namely, we get

$$\tilde{\mathcal{M}}_1\mathbf{y} - \mathcal{K}^T\boldsymbol{p} = \tilde{\mathcal{M}}_1\bar{\mathbf{y}},$$

which would correspond to a backward Euler discretization of the following type

$$-\frac{p^k - p^{k-1}}{\tau} - \triangle p^{k-1} = 0. \tag{2.17}$$

For the optimize-then-discretize approach to represent the adjoint PDE, also the final condition for (2.4) has to be represented by the first equation in (2.16). Note that the last line of the first-equation in (2.16) gives

$$\tilde{M}(\mathbf{y}^N - \bar{\mathbf{y}}^N) = Lp_N,$$

which does not necessarily coincide with final conditions of the adjoint PDE. However, note that in this case the discretize-then-optimize and the optimize-then-discretize approach would not necessarily give the same discretization. Note that for $\tau \to 0$ the

final condition is fulfilled; indeed the final condition of the adjoint equation is satisfied to first order accuracy in τ.

The discretization of the functional (2.2) can be derived in an analogous way with the final results being

$$\begin{bmatrix} \tau\tilde{\mathcal{M}}_{2,1/2} & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \tau\tilde{\mathcal{M}}_{2,1/2}\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix}. \tag{2.18}$$

Note that now the adjoint equation of our discretize-then-optimize approach will differ from a discretization of the adjoint PDE as the mass matrix in the $(1,1)$-block has the factor $\frac{1}{2}$ in the first and last block (from the trapezoidal rule) but this would in general not be obtained from the discretization of the right hand side of (2.5) in an Optimize-then-Discretize approach. As we pointed out earlier the approximation of the first integral in (2.2) via the rectangle rule would result in the following system

$$\begin{bmatrix} \tau\tilde{\mathcal{M}}_{2,0} & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \tau\tilde{\mathcal{M}}_{2,0}\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix} \tag{2.19}$$

where $\tilde{\mathcal{M}}_{2,0} = \text{blkdiag}(\tilde{M}, \tilde{M}, \ldots, \tilde{M}, 0)$. This would allow for a discretization that preserves the final condition of the adjoint PDE (2.5) as the last line of the first part in (2.19) gives

$$Lp_N = 0$$

and hence the correct final condition for the adjoint PDE. We will also address this issue in a different publication [29] for the one-shot treatment of the time-dependent Stokes equation (see [15] also for comments on Discretize-then-Optimize and Optimize-then-Discretize).

A system similar to (2.16) was considered in [3] where the observations were only given on a part of a domain ($\Omega_1 \subsetneq \Omega$) and a hyperbolic PDE was considered.

3. The subset case (non time-dependent case). Before we address the issue of how to precondition the time-dependent problems introduced in the last section, we address the case $\Omega_1 \subsetneq \Omega$ as this will play a significant role for the rest of the paper. For this problem we consider the minimization in the time-independent context of

$$J(y,u) := \frac{1}{2}\int_{\Omega_1} \left(\mathrm{y(x)} - \bar{\mathrm{y}}\mathrm{(x)}\right)^2 dx + \frac{\beta}{2}\int_{\Omega} \left(\mathrm{u(x)}\right)^2 dx. \tag{3.1}$$

subject to Poisson's equation

$$-\triangle y = u$$

with Dirichlet boundary condition $y = 0$ on $\partial\Omega$. The finite element discretization of this problem (see also [28, 21]) is now given by

$$J(y,u) := \frac{1}{2}\left(\mathbf{y} - \bar{\mathbf{y}}\right)^T \tilde{M}\left(\mathbf{y} - \bar{\mathbf{y}}\right) + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u}.$$

subject to

$$K\mathbf{y} = M\mathbf{u} + d$$

where K and M are the stiffness and lumped mass matrices for the domain Ω as above whereas $\tilde{M}$ is a lumped mass matrix for Ω_1. Note for this time-independent example $\mathbf{y}$, $\mathbf{u}$ and $\boldsymbol{p}$ are not vector valued as they are in the time-dependent case. Using a standard Lagrangian technique, we obtain the following optimality system

$$\begin{bmatrix} \tilde{M} & 0 & -K^T \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix}. \tag{3.2}$$

More details for the case $\Omega_1 = \Omega$ can be found in [32, 28, 23, 21]. In this paper we consider only iterative solvers for the saddle point system (3.2). Direct methods [6, 5] typically perform very well for $2D$ problems whereas they are likely to run of memory for discretizations of three dimensional problems (see [21]). As the system (3.2) as well as the systems (2.16) and (2.18) are symmetric and indefinite, the minimal residual method (MINRES) [20] will be our method of choice. MINRES is a method that constructs an increasing search space (Krylov subspace) given by

$$\text{span}\left\{r_0, \mathcal{A}r_0, \mathcal{A}^2 r_0, \ldots, \mathcal{A}^{k-1} r_0\right\}$$

and approximates the solution to the linear system in such a way that for every step $k = 1, 2, \ldots$ of the algorithm, the Euclidean norm of the residual r_k is minimized over the current search space. In general, iterative solvers will only be applied together with a preconditioner $\mathcal{P}$ that speeds up the convergence of the scheme (see [2] for a general overview). As MINRES is our method of choice, we need the preconditioner $\mathcal{P}$ to be symmetric and positive definite (see [7, 2].

A relatively simple observation by Murphy *et al.* [19] guides the search for good preconditioners, viz, a preconditioner only using the $(1,1)$-block of $\mathcal{A}$ and the Schur-complement $S = BA^{-1}B^T$ of $\mathcal{A}$ will lead to convergence in a small number of iterations as the matrix $\mathcal{P}^{-1}\mathcal{A}$ has only few distinct eigenvalues. For MINRES the preconditioner would in general be a block-diagonal matrix of the following form

$$\mathcal{P} = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}.$$

Another preconditioner that has proven to be very effective for saddle point problems is the block-triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix},$$

which for symmetric and positive definite A can be used with a non-standard CG method [4, 22]. For the case of semi-definite A we would have to use non-symmetric methods such as GMRES [25] or BICG [9] with the block-triangular preconditioner. Another alternative for problems of this type are the so-called constraint preconditioners, which themselves are of saddle point form. They were shown to be effective for certain time-independent PDE-constrained optimization problems in [21] and are attractive because of the small number of eigenvalue clusters of the preconditioned matrix (see [17, 18]).

For the subset problem described above we propose the following idealized block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} M & 0 & 0 \\ 0 & \beta M & 0 \\ 0 & 0 & KM^{-1}K^T \end{bmatrix}. \tag{3.3}$$

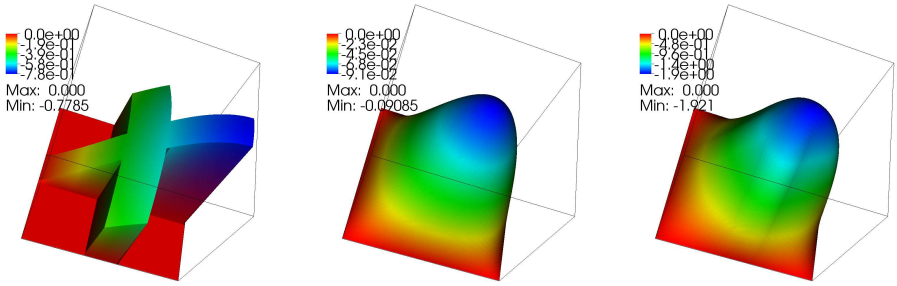
We illustrate the performance of the preconditioner on an example where

$$\bar{\mathbf{y}} = -x_1 e^{-(x_1-0.5)^2-(x_2-0.5)^2} \text{ in } 2D$$

or

$$\bar{\mathbf{y}} = -x_1 e^{-(x_1-0.5)^2-(x_2-0.5)^2-(x_3-0.5)^2} \text{ in } 3D$$

with $\Omega_1 = \{x : x_i \in [c_1, c_2] \text{ for all } i = 1, 2\}$. The regularization parameter β was set to 10^{-2}. Figure 3.1a depicts this data for $c_1 = 0.4$ and $c_2 = 0.6$. Note that whenever we plot the desired state in this paper we plot it as the solution of $\mathcal{M}^{-1}\tilde{\mathcal{M}}\bar{\mathbf{y}}$ or $M^{-1}\tilde{M}\bar{\mathbf{y}}$. Figure 3.1b and 3.1c show the computed state $\mathbf{y}$ and control $\mathbf{u}$. In Table



(a) Desired state (b) State (c) Control

Figure 3.1: Desired state, state and control for subset problem

3.1 we show results for the problem setup given above with simple inversion of the diagonal matrix M and the approximation of K by one AMG multigrid cycle [10] with 10 steps of a Chebyshev smoother. For $\Omega_1 = \Omega$ this approach is described in [21]. The implementation of our method was done in deal.II [1] and we discretized using $Q1$ finite elements. The stopping criterion for MINRES was set to 10^{-6} for the relative pseudo-residual. We show results for $2D$ and $3D$ in Table 3.1. We also want

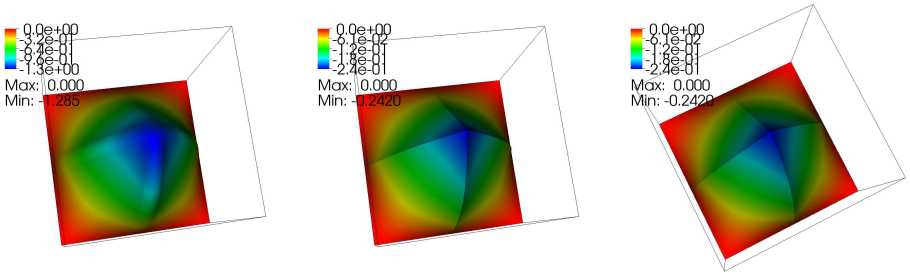
$2D$			$3D$		
DoF	MINRES	Time	DoF	MINRES	Time
1089	16	0.07			
4225	17	0.33	729	15	0.07
16641	20	1.54	4913	15	0.47
66049	26	11.8	35937	13	4.43
263169	32	69.67	274625	17	53.19
1050625	42	402.63	2146689	19	497.26

Table 3.1: Number of MINRES steps with CPU-time.

to illustrate the dependence of the iteration numbers on the width of the cross- given by $x_{1/2} \in [c_1, c_2]$ in Table 3.2. It can be seen that for this setup the width of the 'cross' does not have a negative effect on the iteration numbers.

	$c_1 = 0.45, c_2 = 0.55$	$c_1 = 0.49, c_2 = 0.51$	$c_1 = 0.495, c_2 = 0.505$
DoF	MINRES(T)	MINRES(T)	MINRES(T)
1089	15(0.08)	15(0.07)	15(0.07)
4225	17(0.3)	16(0.28)	16(0.28)
16641	21(1.53)	19(1.4)	18(1.36)
66049	24(11.36)	20(9.07)	20(9.33)
263169	31(67.42)	26(57.33)	24(53.81)

Table 3.2: The lemon juicer: The control **u** for decreasing width of Ω_1 and number of MINRES steps with CPU-time.



(a) $c_1 = 0.45, c_2 = 0.55$ (b) $c_1 = 0.49, c_2 = 0.51$ (c) $c_1 = 0.495, c_2 = 0.505$

Figure 3.2: The lemon juicer: Decreasing width of Ω_1

4. Solution of the linear systems and preconditioning for time-dependent problems.

4.1. The $J_1(y,u)$ case. In this Section we focus on a preconditioner for the system (2.16). We want to emphasize the fact that the matrix $\tilde{\mathcal{M}}$ is symmetric and positive semi-definite as most of its diagonal blocks are zero. For many problems in this so-called saddle point form, a Krylov subspace solver in combination with a block-preconditioner will give good results [2] as indicated in the previous section. As for our matrix,

$$\mathcal{A} = \begin{bmatrix} \tilde{\mathcal{M}}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \tag{4.1}$$

the left-upper block blkdiag$(\tilde{\mathcal{M}}_1, \mathcal{M})$ is only positive semi-definite, we propose the following preconditioner

$$\mathcal{P} = \begin{bmatrix} \hat{\mathcal{M}} & 0 & 0 \\ 0 & \beta\tau\mathcal{M} & 0 \\ 0 & 0 & S \end{bmatrix}. \tag{4.2}$$

The block $\hat{\mathcal{M}} = \text{blkdiag}(\gamma^{-1}I, \gamma^{-1}I, \ldots, \gamma^{-1}I, M)$ with a parameter $\gamma > 0$ is now assumed to be part of a perturbed problem and will be chosen as the preconditioner

for $\tilde{\mathcal{M}}_1$. Also we consider the alternative choice $\hat{\mathcal{M}} = \text{blkdiag}(M, M, \ldots, M, M)$ for the Schur complement approximation, i.e., $S = \mathcal{K}\hat{\mathcal{M}}^{-1}\mathcal{K}^T + \tau\beta^{-1}\mathcal{M}$ or even $\hat{\mathcal{M}} = \text{blkdiag}(\gamma M, \gamma M, \ldots, \gamma M, M)$. We will illustrate the difference between these choices in our numerical experiments (see Section 6.1).

It is obvious that we never want to form the Schur-complement explicitly as the storage requirements for realistic scenarios would not be feasible. This was done in an initial study in [3] but it was emphasized that this should be improved. We rather focus on a strategy where we approximate the Schur complement S by neglecting the term $\beta^{-1}\mathcal{M}$ as initially proposed for non time-dependent problems (see [21]). We then get $S_0 = \mathcal{K}\hat{\mathcal{M}}^{-1}\mathcal{K}$. Thorne recently analyzed the sensitivity of this approach with respect to varying β [31]. It has to be noted that we have a natural decomposition of S_0 that allows us to compute S_0^{-1} relatively cheaply. We have the following inverse of S_0,

$$S_0^{-1} = \mathcal{K}^{-T}\hat{\mathcal{M}}\mathcal{K}^{-1}, \tag{4.3}$$

where $\mathcal{K}$ is lower triangular. This results in the inverse of $\mathcal{K}$ being relatively cheap to implement. Note that $\mathcal{K}$ is a lower block triangular matrix which means that both the application of $\mathcal{K}^{-1}$ and $\mathcal{K}^{-T}$ are simple forward and backward substitutions, which only involve the solution of a linear system with the block $L = M + \tau K$. The aim of a preconditioner is to resemble the original matrix by also being easy to apply. Hence, a good choice for the approximation of $\mathcal{K}^{-1}$ is given by $\hat{\mathcal{K}}^{-1}$, which is defined as a forward substitution not using L but rather $\hat{L}$, a multigrid approximation of L that we will discuss in Section 4.3.

This now results in a good approximation $\hat{S} \approx S_0$ such that

$$\hat{S}^{-1} = \hat{\mathcal{K}}^{-T}\hat{\mathcal{M}}\hat{\mathcal{K}}^{-1}$$

with the preconditioner in its final form

$$\mathcal{P} = \begin{bmatrix} \hat{\mathcal{M}} & 0 & 0 \\ 0 & \beta\tau\mathcal{M} & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}. \tag{4.4}$$

4.2. The $J_2(y,u)$ case. In the case when the functional $J_2(y,u)$ is considered, the matrix $\tilde{\mathcal{M}}_{2,1/2}$ or $\tilde{\mathcal{M}}_{2,0}$ is simply denoted by $\tilde{\mathcal{M}}_2$

$$\mathcal{A} = \begin{bmatrix} \tau\tilde{\mathcal{M}}_2 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \tag{4.5}$$

is a block-diagonal matrix consisting of mass matrices for Ω_1. For this, we now propose the following preconditioner

$$\mathcal{P} = \begin{bmatrix} \tau\hat{\mathcal{M}} & 0 & 0 \\ 0 & \beta\tau\mathcal{M} & 0 \\ 0 & 0 & \hat{S} \end{bmatrix} \tag{4.6}$$

where $\hat{\mathcal{M}} = \text{blkdiag}(M, M, \ldots, \gamma^{-1}I)$. The Schur-complement S will now be approximated in the same way as in Section 4.1. We will use

$$\hat{S}^{-1} = \tau\hat{\mathcal{K}}^{-T}\hat{\mathcal{M}}\hat{\mathcal{K}}^{-1},$$

resulting in a good approximation $\hat{S} \approx S_0$, which again allows choosing an appropriate $\hat{M}$, here we use $\hat{\mathcal{M}} = \text{blkdiag}(M, M, \ldots, \gamma M)$. Our Schur-complement approximation $\hat{S}$ is also accounting for the factor τ in the $(1,1)$-block of the system matrix $\mathcal{A}$. We want to emphasize the fact that now the time-step τ also enters the $(1,1)$-block of the preconditioner as this will be strongly used in the eigenvalue analysis of the preconditioned matrix.

4.3. Approximating L. As we have seen in the previous sections, in order to have efficient preconditioners for the setups given in (2.1) and (2.2) we need to approximate the matrix $L = M + \tau K$ efficiently. Note that this matrix is symmetric and positive definite as M is positive definite and K positive (semi)definite depending on the imposed boundary condition. Eigenvalue bounds for the mass matrix are well studied and can be found in [33] or [7, Proposition 1.29]. The eigenvalues of the stiffness matrix are well analyzed and can be approximated based on the mesh-parameter and some discretization-dependent constants (see [7, Theorem 1.32]). For $2D$ examples with a moderate size, the factorization of L using direct methods would be feasible [6, 5] but for larger problems L should be approximated in a less expensive way. A key observation is that exact solution of systems with L is not necessary in the preconditioner. That is, exact evolution of the forward problem and backwards of the adjoint is *not* required here. Only in matrix×vector products with the coefficient matrix in the outer MINRES iteration is L used exactly simply to multiply.

A typical approximation for L would be the use of a multigrid method of geometric (GMG) [35, 12] or algebraic nature (AMG) [24, 8]. We will focus here on algebraic multigrid, namely, the smoothed aggregation AMG implemented as part of the Trilinos [13] ML- package [10]. The ML package allows for a variety of smoothers and in our experience 10 steps of the Chebyshev smoother were superior to similar or smaller numbers of Gauss-Seidel or Jacobi smoothing steps. One explanation for this behavior might be the importance of the mass matrix in L at least for small τ given that Chebyshev methods are known to be very efficient for mass matrices (see [34]). The use of one AMG V-cycle was always sufficient for our experiments and in fact we did not observe much improvement when a larger number of V-cycles was used. This is a key observation in our approach since an approximate evolution of the underlying PDE is made in a simple forward substitution: no attempt is made to accurately solve each time evolution through accurate or exact representation of L.

5. Eigenvalue Analysis. For symmetric systems the clustering of the eigenvalues plays a crucial role in the speed of convergence and we therefore analyze the generalized eigenvalue problem $\mathcal{A}v = \lambda \mathcal{P}v$ with the idealized block-preconditioner.

We start by analyzing a somewhat simplified case where $\Omega_1 = \Omega$ for the problem (2.2) and using the trapezoidal rule as for this problem the saddle point system has a symmetric and positive definite $(1,1)$ block and we can use $\hat{\mathcal{M}} = \mathcal{M}$. The eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ where $\mathcal{P}$ is the idealized block-diagonal preconditioner $\mathcal{P} = \text{blkdiag}(\tau\mathcal{M}, \beta\tau\mathcal{M}, \tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K})$ would then be given by 1 and eigenvalues depending on the term

$$\tau\mathcal{K}^{-T}\mathcal{M}\mathcal{K}^{-1}\left(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T + \tau\beta^{-1}\mathcal{M}\right), \tag{5.1}$$

which we analyze using a field of values approach [11]. The eigenvalues of (5.1) only have a benign dependence on τ and our aim is to show that they are independent of the discretization parameter h. Note that for the problem (3.1) with $\Omega_1 = \Omega$ the eigenvalues depend on the term $K^{-T}MK^{-1}M$, which is independent of

the mesh parameter h (see [21] for more details on block-diagonal preconditioners and [28] for block-triangular preconditioners). Now the crucial part in (5.1) is the term $\beta^{-1}\tau^2\mathcal{K}^{-T}\mathcal{M}\mathcal{K}^{-1}\mathcal{M}$ which is similar to $\beta^{-1}\tau^2\mathcal{M}^{1/2}\mathcal{K}^{-T}\mathcal{M}\mathcal{K}^{-1}\mathcal{M}^{1/2}$ and we get

$$\frac{\tau^2\beta^{-1}x^T\mathcal{M}^{1/2}\mathcal{K}^{-T}\mathcal{M}\mathcal{K}^{-1}\mathcal{M}^{1/2}x}{x^Tx} = \frac{\tau^2\beta^{-1}\left(z^T\mathcal{M}z\right)\left(x^T\mathcal{M}x\right)\left(y^T\mathcal{K}^{-T}\mathcal{K}^{-1}y\right)}{\left(x^Tx\right)\left(y^Ty\right)\left(z^Tz\right)}, \quad (5.2)$$

where $y = \mathcal{M}^{1/2}x$, $z = \mathcal{K}^{-1}y$. From this we can see that the eigenvalues of the overall system are determined by the maximal eigenvalues of $\mathcal{M}$ which are just the eigenvalues of the mass matrix M and the maximal eigenvalues of $\mathcal{K}^{-T}\mathcal{K}^{-1}$. But as we do not have an easy expression for this matrix we will instead look at the eigenvalues of

$$\mathcal{L} = \mathcal{K}\mathcal{K}^T = \begin{bmatrix} L^2 & -LM & & & & \\ -ML & L^2+M^2 & \ddots & & & \\ & -ML & L^2+M^2 & -LM & & \\ & & \ddots & \ddots & -LM & \\ & & & -ML & L^2+M^2 \end{bmatrix}, \quad (5.3)$$

which is a symmetric and positive definite matrix.

We now want to use this fact that $\mathcal{K}$ is a block triangular matrix in the analysis of the eigenvalues of $\mathcal{L}$ as the eigenvalues of this matrix are simply the squares of the singular values of $\mathcal{K}$. The structure of $\mathcal{K}$ is very beneficial for the determination of its eigenvalues but it is not obvious from the block-triangular matrix how to estimate the singular values. We suggest the following splitting $\mathcal{K} = D + R$ where $D = \text{blkdiag}\,(L, \ldots, L)$ and

$$R = \begin{bmatrix} 0 & & & \\ -M & 0 & & \\ & \ddots & 0 & \\ & & -M & 0 \end{bmatrix}. \quad (5.4)$$

We are now trying to find a bound for the relation

$$\left|\sigma_i^{(\mathcal{K})} - \sigma_i^{(D)}\right|$$

viewing R as a perturbation term. Using the results in [26, 36] we get that

$$\left|\sigma_i^{(\mathcal{K})} - \sigma_i^{(D)}\right| \leq \|R\|. \quad (5.5)$$

This means that the singular values of $\mathcal{K}$ lie in circles around the singular values of D (the eigenvalues of L) with a radius given by the spectral norm of R. For a 2-dimensional domain we can use results for the mass matrix [7, Proposition 1.29] and stiffness matrices [7, Theorem 1.32], i.e,

$$c_1h^2 \leq \frac{x^TMx}{x^Tx} \leq d_1h^2 \quad (5.6)$$

$$c_2h^2 \leq \frac{x^TKx}{x^Tx} \leq d_2 \quad (5.7)$$

$$(c_1+\tau c_2)h^2 \leq \frac{x^TLx}{x^Tx} \leq d_1h^2 + \tau d_2 \quad (5.8)$$

where d_1, d_2, c_1, and c_2 are all constants independent of h, to compute $\|R\|$ using the definition

$$\|R\| = \sqrt{\lambda_{max}(R^T R)}$$

and as $R^T R = \text{blkdiag}(0, M^2, M^2, \dots M^2)$ we see that

$$\|R\| = \sqrt{\lambda_{max}(M^2)}.$$

From (5.6) we know that

$$\|R\| \leq d_1 h^2. \tag{5.9}$$

Figure 5.1 shows the singular values of both $\mathcal{K}$ and D with the bound (5.9) incorporated. We recall now that we are interested in finding a bound for the maximal eigenvalue of $\mathcal{K}^{-T}\mathcal{K}^{-1}$ by using the minimal eigenvalue of $\mathcal{K}\mathcal{K}^T$ which is the square of the minimal singular value of $\mathcal{K}$. Hence, we get that

$$(c_1 + \tau c_2 - d_1)h^2 \leq \sigma(\mathcal{K}) \Rightarrow (c_1 + \tau c_2 - d_1)^2 h^4 \leq \lambda_{min}(\mathcal{K}\mathcal{K}^T$$

which results in the following bound

$$\lambda_{max}(\mathcal{K}^{-T}\mathcal{K}^{-1}) \leq \frac{1}{(c_1 + \tau c_2 - d_1)^2 h^4}.$$

Finally, the desired bound for

$$\lambda_{max}\left(\mathcal{M}^{1/2}\mathcal{K}^{-T}\mathcal{M}\mathcal{K}^{-1}\mathcal{M}^{1/2}\right) \leq \frac{d_1^2 h^4}{(c_1 + \tau c_2 - d_1)^2 h^4} = \frac{d_1^2}{(c_1 + \tau c_2 - d_1)^2}$$

which is independent of h and we can observe that the bound is virtually unchanged for varying τ. Note that for a three dimensional problem, we can proceed in the exact same way by replacing the bounds on mass and stiffness matrix with their three-dimensional equivalent (see [7] for details).

We now turn our attention to the case when the $(1,1)$ block is semi-definite and assume that the saddle point matrix $\mathcal{A}$ is permuted in such a way that the zero entries in $\tilde{\mathcal{M}}$ all appear in the left upper corner. For clarity of exposition, we neglect the scalar τ in the $(1,1)$-block of the saddle point system. Clearly, this depends on the choice of Ω_1 as in the case of $\Omega_1 = \Omega$ we would have $\tilde{\mathcal{M}} = \mathcal{M}$. In more detail, the system is of the following form

$$\breve{\mathcal{A}} = \begin{bmatrix} 0 & 0 & 0 & -\mathcal{K}_1^T \\ 0 & \mathcal{M}^{(2)} & 0 & -\mathcal{K}_2^T \\ 0 & 0 & \beta\tau\mathcal{M} & \tau\mathcal{M} \\ -\mathcal{K}_1 & -\mathcal{K}_2 & \tau\mathcal{M} & 0 \end{bmatrix} \tag{5.10}$$

where $\mathcal{M}^{(2)}$ represents the contributions from Ω_1 for problems (2.1), (2.2), and (3.1) also including possible τ factors. The preconditioner is of the following permuted form (as all the mass matrices are in lumped form)

$$\breve{\mathcal{P}} = \begin{bmatrix} \hat{\mathcal{M}}^{(1)} & 0 & 0 & 0 \\ 0 & \mathcal{M}^{(2)} & 0 & 0 \\ 0 & 0 & \beta\tau\mathcal{M} & 0 \\ 0 & 0 & 0 & \mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T \end{bmatrix}, \tag{5.11}$$

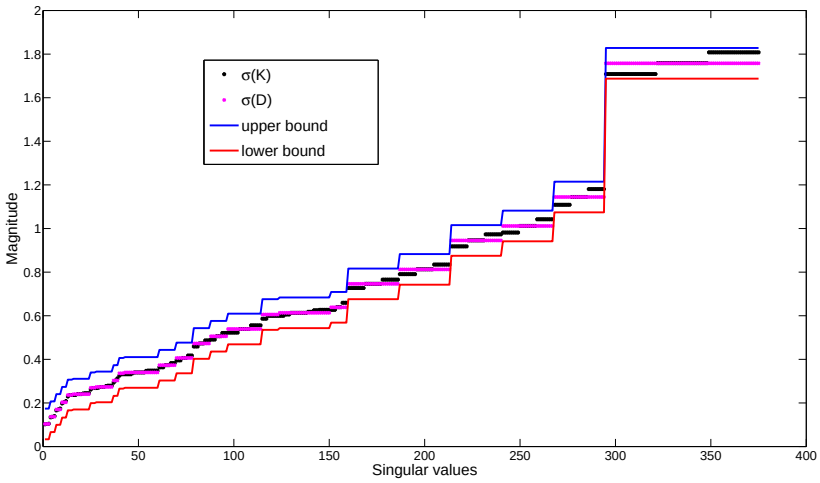


Figure 5.1: Singular values of $\mathcal{K}$ versus singular values of D with upped and lower bounds determined by $\|R\|$

where $\hat{S} = \mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T$ (any factor τ will be included in the matrix $\mathcal{M}$ coming from the $(1,1)$-block). We are now interested in the eigenvalues of $\breve{\mathcal{P}}^{-1}\breve{\mathcal{A}}$, which are equivalent to the eigenvalues of

$$\mathcal{H} = \breve{\mathcal{P}}^{-1/2}\breve{\mathcal{A}}\breve{\mathcal{P}}^{-1/2} = \begin{bmatrix} 0 & 0 & 0 & B_1^T \\ 0 & I & 0 & B_2^T \\ 0 & 0 & I & B_3^T \\ B_1 & B_2 & B_3 & 0 \end{bmatrix} \tag{5.12}$$

where $B_1 = \hat{S}^{-1/2}\mathcal{K}_1\left(\hat{\mathcal{M}}^{(1)}\right)^{-1/2}$, $B_2 = \hat{S}^{-1/2}\mathcal{K}_2\left(\hat{\mathcal{M}}^{(2)}\right)^{-1/2}$ and $B_3 = \beta^{-1/2}\tau^{1/2}\hat{S}^{-1/2}\mathcal{M}^{1/2}$. The eigenvalues of $\mathcal{H}$ are given by all values λ such that

$$(\mathcal{H} - \lambda I)v = \begin{bmatrix} -\lambda I & 0 & 0 & B_1^T \\ 0 & (1-\lambda)I & 0 & B_2^T \\ 0 & 0 & (1-\lambda)I & B_3^T \\ B_1 & B_2 & B_3 & -\lambda I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = 0. \tag{5.13}$$

From (5.13) we see that $\lambda = 1$ is an eigenvalue with the multiplicity determined by the nullspace of B_2 and B_3, i.e., for $\lambda = 1$ the vector $[0, v_2, v_3, 0]^T$ with $B_2 v_2 = 0$, $B_3 v_3 = 0$ satisfies (5.13). We now assume that $\lambda \neq 1$ and we get from (5.13) that

$$\left(\frac{1}{\lambda}B_1B_1^T + \frac{1}{\lambda-1}B_2B_2^T + \frac{1}{\lambda-1}B_3B_3^T - \lambda I\right)v_4 = 0. \tag{5.14}$$

In order to find λ for (5.14) we look at

$$(\lambda-1)B_1B_1^T + \lambda B_2B_2^T + \lambda B_3B_3^T - \lambda^2(\lambda-1)I$$

which can be simplified to give

$$\lambda^3 - \lambda^2 - \lambda x^T\left(B_1B_1^T + B_2B_2^T + B_3B_3^T\right)x - x^TB_1B_1^Tx. \tag{5.15}$$

We can now solve this equation using the rather complicated cubic formula, i.e.,

$$\lambda = \left\{ q + [q^2 + (r - p^2)^3]^{1/2} \right\}^{1/3} + \left\{ q - [q^2 + (r - p^2)^3]^{1/2} \right\}^{1/3} + p \tag{5.16}$$

where $p = 1/3$, $q = 1/27 + \left(x^T \left(B_1 B_1^T + B_2 B_2^T + B_3 B_3^T \right) x + 3 x^T B_1 B_1^T x \right) /6$ and $r = x^T \left(B_1 B_1^T + B_2 B_2^T + B_3 B_3^T \right) x/3$. Note that in order to obtain meaningful bounds for λ in (5.16) we have to find bounds for the terms

$$x^T \left(B_1 B_1^T + B_2 B_2^T + B_3 B_3^T \right) x = \hat{S}^{-1} (\mathcal{K} \hat{\mathcal{M}}^{-1} \mathcal{K}^T + \beta^{-1} \tau \mathcal{M}) \tag{5.17}$$

and

$$x^T B_1 B_1^T x = \hat{S}^{-1} \mathcal{K}_1 \left(\hat{\mathcal{M}}^{(1)} \right)^{-1} \mathcal{K}_1^T$$

where we now can proceed using the techniques presented for the case of a positive definite $(1,1)$ block in $\mathcal{A}$, i.e., Equation (5.17) is very similar to (5.2) and $\hat{\mathcal{M}}^{(1)}$ might include a parameter γ in which case $x^T B_1 B_1^T x$ goes to zero as γ goes to zero.

The eigenvalue analysis becomes more accessible if a block-triangular preconditioner

$$\breve{\mathcal{P}} = \begin{bmatrix} \hat{\mathcal{M}} & 0 & 0 \\ 0 & \beta\tau\mathcal{M} & 0 \\ -\mathcal{K} & \tau\mathcal{M} & -\hat{S} \end{bmatrix}$$

is used as the preconditioned matrix becomes

$$\breve{\mathcal{P}}^{-1} \breve{\mathcal{A}} = \begin{bmatrix} 0 & 0 & 0 & -\left(\hat{\mathcal{M}}^{(1)} \right)^{-1} \mathcal{K}_1^T \\ 0 & I & 0 & -\left(\mathcal{M}^{(2)} \right)^{-1} \mathcal{K}_2^T \\ 0 & 0 & I & \beta^{-1} I \\ -\hat{S}^{-1}\mathcal{K}_1 & 0 & 0 & \hat{S}^{-1}(\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T + \beta^{-1}\tau\mathcal{M}) \end{bmatrix}.$$

A preconditioned matrix of this form was analyzed in [3], where it was shown that $\breve{\mathcal{P}}^{-1}\breve{\mathcal{A}}$ has an eigenvalue cluster at 1 and the remaining eigenvalues satisfy a quadratic relation.

Note that in comparison to the case when the $(1,1)$ block is positive definite the polynomials that determine the convergence here are one degree higher than in the positive definite case. Namely, a linear polynomial would describe the eigenvalues of the block-triangular preconditioned matrix and a quadratic polynomial would determine the eigenvalues for block-diagonal preconditioners.

6. Numerical Experiments. The results presented in this Section are based on an implementation of the above described algorithms within the deal.II [1] framework. For the AMG preconditioner, we used the Trilinos ML package [10] that implements a smoothed aggregation AMG. Our implementation of MINRES was taken from [7] and was stopped with a tolerance of 10^{-4} for the relative pseudo residual. For all our experiments $T = 1$ and $\tau = 0.05$, which results in 20 time-steps. We discretized using $Q1$ finite elements for each of the state, control and Lagrange multiplier fields as well as the backward Euler scheme. Throughout, we use $\beta = 10^{-4}$ and $\gamma = 10^{-8}$. Also note that whenever we mention the degrees of freedom that in fact we are solving a linear system of the dimension 3 times the number of time-steps (N) times the degrees of freedom of the spatial discretization. For example, a spatial discretization with 274625 unknowns and 20 time-steps corresponds to an overall linear system of dimension 16477500.

6.1. The $J_1(y,u)$ case.

6.1.1. $\Omega_1 = \Omega$. From our experience, the results in this section typically follow the pattern that the control drives the state to zero for early times and only in the last time steps before reaching T do we see an increase in $\mathbf{y}$ towards the desired state $\bar{\mathbf{y}}$. As we already mentioned in Section 4, there exists a variety of preconditioners that can be chosen for this problem. From our experience (see Table 6.1) the $(1,1)$ block is best preconditioned by

$$\hat{\mathcal{M}} = \text{blkdiag}(\gamma^{-1}I, \gamma^{-1}I, \ldots, \gamma^{-1}I, M),$$

with small γ whereas the Schur-complement approximation best suited for the problems considered here is given by $\hat{S} = \hat{\mathcal{K}}\hat{\mathcal{M}}^{-1}\hat{\mathcal{K}}^T$ with

$$\hat{\mathcal{M}} = \text{blkdiag}(\gamma M, \gamma M, \ldots, \gamma M, M)$$

or

$$\hat{\mathcal{M}} = \text{blkdiag}(M, M, \ldots, M, M).$$

We start with the following setup: $y_0 = 0$ and

$$\bar{\mathbf{y}} = -x_1 e^{-(x_1-0.5)^2-(x_2-0.5)^2-(x_3-0.5)^2}.$$

The parameter γ was set to 10^{-8}. Table 6.1 shows the results for a sequence of three spatial meshes.

	γM	M	$\gamma M\ \beta = 1e-2$
DoF	MINRES(T)	MINRES(T)	MINRES(T)
4913	39(22)	73(41)	13(8)
35937	42(192)	77(345)	13(64)
274625	42(1550)	75(2721)	11(449)

Table 6.1: Different Schur-complement approximations and varying β: Number of MINRES iterations (CPU time).

6.1.2. $\Omega_1 \subsetneq \Omega$. We use $\hat{\mathcal{M}} = \text{blkdiag}(\gamma^{-1}I, \gamma^{-1}I, \ldots, \gamma^{-1}I, M)$ for the $(1,1)$ block and $\hat{\mathcal{M}} = \text{blkdiag}(\gamma M, \gamma M, \ldots, \gamma M, M)$ in the Schur-complement approximation. We have $\Omega_1 = \{x : x_1 \in [0.4, 0.8]\}$ with $y_0 = 0$ and desired final state

$$\bar{\mathbf{y}} = -x_1 e^{-(x_1-0.5)^2-(x_2-0.5)^2-(x_3-0.5)^2}.$$

The numerical results for the above setup are shown in Table 6.2. We also want to present another example using a different domain for Ω_1 to illustrate the mesh-independence of our solver. Hence, for the functional $J_1(y,u)$ we describe choose the desired state to be

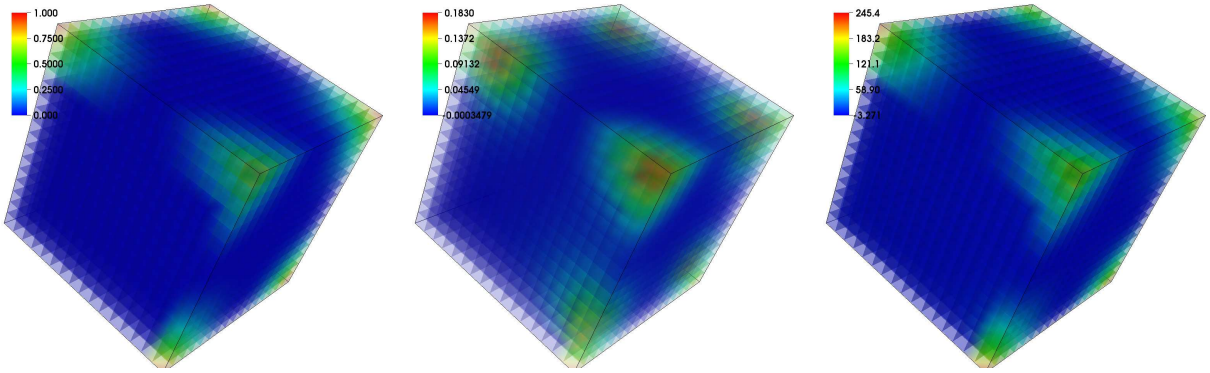
$$\bar{\mathbf{y}} = (2x_1 - 1)^2(2x_2 - 1)^2(2x_3 - 1)^2$$

with

$$\Omega_1 = \{x : x_1 \in [0, 0.1] \text{ or } x_2 \in [0.8, 1.0]\}$$

DoF	MINRES	Time
4913	99	55
35937	97	433
274625	104	3768

Table 6.2: Number of MINRES steps with CPU-time.



(a) Desired state at $t = 1$ (b) State at $t = 1$ (c) Control at $t = 1$

Figure 6.1: $3D$ problem with Dirichlet boundary condition

and $y_0 = 0$. Results are given in Table 6.3 and we show the desired state as well as the computed state and control in Figure 6.1. For both problems presented in this section we can see that the iteration numbers are higher than in the case of $\Omega_1 = \Omega$. But we still observe mesh-independence as well as an optimal scaling by a factor of 8 for each refinement of the mesh.

DoF	MINRES	Time
4913	138	77
35937	142	635
274625	138	5010

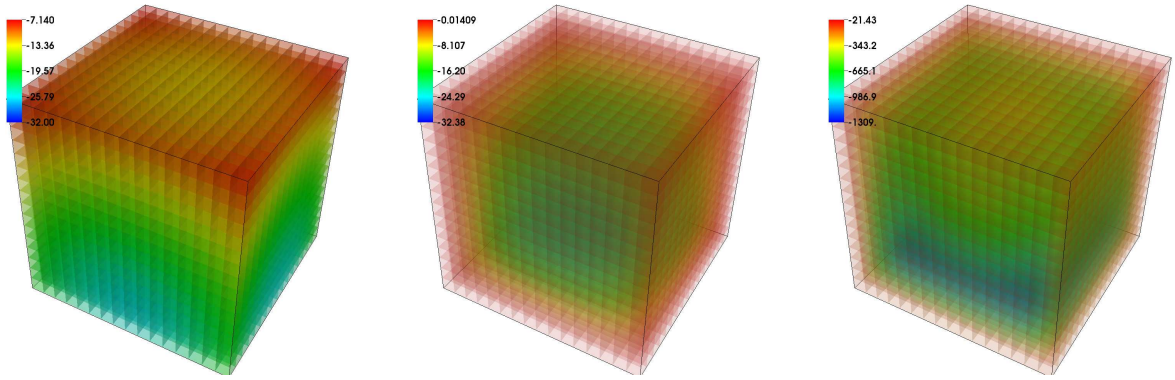
Table 6.3: Number of MINRES steps with CPU-time.

6.2. The $J_2(y, u)$ case. We now give numerical results for the case when the functional (2.2) is solved using the techniques described earlier.

6.2.1. $\Omega_1 = \Omega$. Note that for this setup, the $(1, 1)$ block of the system matrix $\mathcal{A}$ is symmetric and positive definite. In this setup we could, as explained earlier, also employ a Conjugate Gradients method [14] based on a non-standard inner product (see [4, 27, 22, 23, 30]) for details), though we have not done so here.

Our first example is determined by the desired state

$$\bar{\mathbf{y}} = -64te^{-(x_1-0.5)^2-(x_2-0.5)^2-(x_3-0.5)^2}$$



(a) Desired state at $t = 0.5$ (b) State at $t = 0.5$ (c) Control at $t = 0.5$

Figure 6.2: $3D$ problem with Dirichlet boundary condition

with $y_0 = 0$. We show the desired state (Figure 6.2a), the state (Figure 6.2b) and the control (Figure 6.2c) and give the iterations numbers and timings for MINRES applied to problems with a variety of mesh-sizes in Table 6.4. Table 6.5 shows results for two mesh-size if the change the time-step from $\tau = 0.05$ to $\tau = 0.01$.

DoF	MINRES	Time
4913	22	14
35937	20	100
274625	20	811

Table 6.4: Number of MINRES steps with CPU-time.

DoF	MINRES	Time
4913	32	99
35937	30	733

Table 6.5: Number of MINRES steps with CPU-time for $\tau = 0.01$.

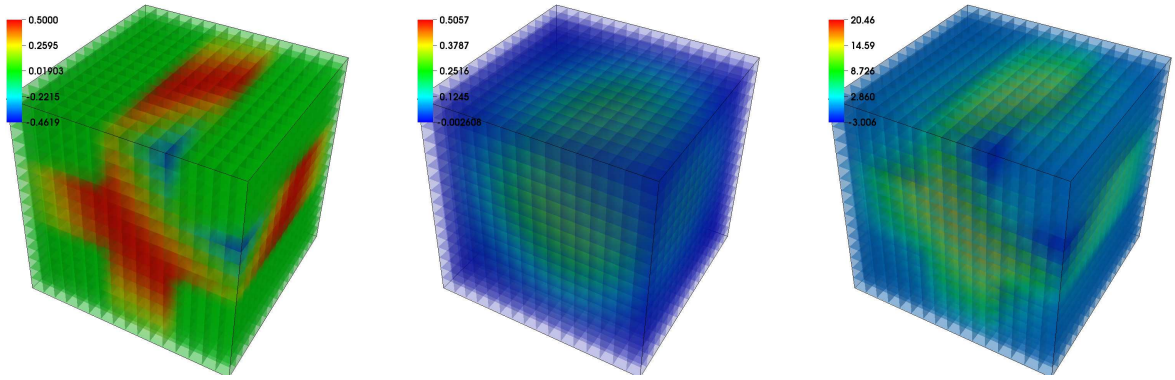
6.2.2. $\Omega_1 \subsetneq \Omega$**.** We now come to the case when we have Ω_1 as a proper subdomain of Ω. Our domain $\Omega_1 = \{x : x_i \in [0.4, 0.7] \text{ for } i = 1, 2\}$ with an initial state

$$\mathbf{y}_0 = -0.01e^{-(x_1-0.5)^2-(x_2-0.5)^2-(x_3-0.5)^2}$$

and the desired state being

$$\bar{\mathbf{y}} = t\sin(2\pi x_1 x_2 x_3).$$

Table 6.6 shows the number of MINRES iterations as well as the timings for the solution of the linear systems using the ML AMG preconditioner. Again, we show the desired state (Figure 6.3a), the state (Figure 6.3b) and the control (Figure 6.3c).



(a) Desired state at $t = 0.5$ (b) State at $t = 0.5$ (c) Control at $t = 0.5$

Figure 6.3: $3D$ problem with Dirichlet boundary condition

DoF	MINRES	Time
4913	61	36
35937	53	253
274625	56	2166

Table 6.6: Number of MINRES steps with CPU-time.

7. Conclusions. In this paper we presented formulations for two problems frequently occurring in the optimal control of time-dependent problems. We proposed a one-shot approach to solve these problem and showed that this leads to saddle point problems of very large dimensionality. The resulting linear system also has changed properties depending on the formulation of the control problem; frequently the state is only needed to be close to a desired state on a subdomain of the spatial domain or even only on a part of the time domain. This then leads to semi-definite $(1, 1)$ blocks in the saddle point system. We illustrated how a block-diagonal preconditioner for a time-independent problem gives very competitive results and showed that this technique can be extended to the case of time-dependent problems.

We were able to propose a preconditioner that does not need to solve the time evolution accurately (e.g. using a direct method) but rather only uses a AMG preconditioner (Trilinos AMG [10]). This not only speeds up the solution of the linear system but also allows for much larger problems to be solved as the AMG preconditioner can easily handle extremely large three-dimensional matrices.

We also presented an eigenvalue analysis for the preconditioners proposed and were able to show that our setup leads to a good clustering of the eigenvalues and hence fast convergence of the method in the spirit of [19].

Our numerical results showed competitive results and mesh-independent convergence behavior for all problems. We achieved this by never having to solve the time evolution accurately but rather using a cheap approximation at each step.

Acknowledgment. The authors would like to thank Roland Herzog and Frank Schmidt for pointing out some flaws in the initial version of this manuscript. This

publication is partially based on work supported by Award No. KUK-C1-013-04 , made by King Abdullah University of Science and Technology (KAUST)

REFERENCES

[1] 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.

[2] Michele Benzi, Gene H. Golub, and Jörg Liesen, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.

[3] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of pde-constrained optimization problems*, To appear in Advances in Computational Mathematics, (2010).

[4] James H. Bramble and Joseph E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp, 50 (1988), pp. 1–17.

[5] 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.

[6] I.S. Duff, *Sparse numerical linear algebra: direct methods and preconditioning*, Technical Report TR/PA/96/22, CERFACS, Toulouse, France, 1996. Also RAL Report RAL 96-047.

[7] 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.

[8] R.D. Falgout, *An Introduction to Algebraic Multigrid*, Computing in Science and Engineering, 8 (2006), pp. 24–33. Special Issue on Multigrid Computing.

[9] 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.

[10] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.

[11] 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.

[12] Wolfgang Hackbusch, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[13] Michael Heroux, Roscoe Bartlett, Vicki Howle Robert Hoekstra, Jonathan Hu, Tamara Kolda, Richard Lehoucq, Kevin Long, Roger Pawlowski, Eric Phipps, Andrew Salinger, Heidi Thornquist, Ray Tuminaro, James Willenbring, and Alan Williams, *An Overview of Trilinos*, Tech. Report SAND2003-2927, Sandia National Laboratories, 2003.

[14] Magnus R. Hestenes and Eduard Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand, 49 (1952), pp. 409–436 (1953).

[15] 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.

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

[17] C. Keller, N.I.M. Gould, and A.J. Wathen, *Constraint Preconditioning for Indefinite Linear Systems*, SIAM J. Matrix Anal. Appl, 21 (2000), pp. 1300–1317.

[18] Ladislav Lukšan and Jan Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.

[19] Malcolm F. Murphy, Gene H. Golub, and Andrew J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput, 21 (2000), pp. 1969–1972.

[20] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.

[21] 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.

[22] Tyrone Rees and Martin Stoll, *Block triangular preconditioners for PDE constrained optimization*, Accepted to Numer. Linear Algebra Appl., (2009).

[23] Tyrone Rees, Martin Stoll, and Andy Wathen, *All-at-once preconditioners for PDE-constrained optimization*, Accepted to Kybernetika, (2009).

[24] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.
[25] Youcef 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.
[26] GW Stewart, *Perturbation theory for the singular value decomposition*, SVD and Signal Processing, II: Algorithms, Analysis and Applications, (1990), pp. 99–109.
[27] Martin Stoll, *Solving Linear Systems using the Adjoint*, PhD thesis, University of Oxford, 2009.
[28] Martin Stoll and Andy Wathen, *Preconditioning for PDE-constrained optimization with control constraints*, Submitted, (2009).
[29] ———, *All-at-once solution of time-dependent stokes control*, In preparation, (2010).
[30] ———, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Submitted, (2010).
[31] H. S. Thorne, *Properties of linear systems in PDE-constrained optimization. Part I: Distributed control*, Submitted, (2010).
[32] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*, Vieweg Verlag, Wiesbaden, 2005.
[33] A. J. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.
[34] Andrew J. Wathen and Tyrone Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electronic Transactions in Numerical Analysis, 34 (2008-2009), pp. 125–135.
[35] Pieter Wesseling, *An introduction to multigrid methods*, Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.
[36] H. Weyl, *Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung)*, Mathematische Annalen, 71 (1912), pp. 441–479.