Hamburger Beiträge
zur Angewandten Mathematik

Nested multigrid methods for time-periodic, parabolic optimal control problems

Dirk Abbeloos, Moritz Diehl,
Michael Hinze, Stefan Vandewalle

Nr. 2011-03
January 2011

Noname manuscript No.
(will be inserted by the editor)

Nested multigrid methods for time-periodic, parabolic optimal control problems.

Dirk Abbeloos · Moritz Diehl · Michael Hinze · Stefan Vandewalle

the date of receipt and acceptance should be inserted later

Abstract We present a nested multigrid method to optimize time-periodic, parabolic, partial differential equations (PDE). We consider a quadratic tracking objective with a linear parabolic PDE constraint. The first order optimality conditions, given by a coupled system of boundary value problems can be rewritten as an Fredholm integral equation of the second kind, which is solved by a multigrid of the second kind. The evaluation of the integral operator consists of solving sequentially a boundary value problem for respectively the state and the adjoints. Both problems are solved efficiently by a time-periodic space-time multigrid method.

1 Introduction

We consider optimal control problems constrained by a time-periodic parabolic partial differential equation (PDE). Applications of such problems include the design of reverse flow reactors (Logist et al., 2007), cyclically steered (bio-) reactors (Houska et al., 2009) and energy-producing kites (Houska and Diehl, 2010).

The goal of the optimization is to find the optimal control which minimizes the objective over a set of admissible controls and in addition induces a prescribed cyclic state with the correct period. More precisely the state of the system at a certain instance of time should be identical to the one a fixed period of time later. In this paper we do not assume the existence of autonomous periodic behavior of the PDE and as a consequence the time-periodicity should be enforced by the applied control. Our problem differs from the initial value problem, where the goal is to find the optimal control to steer the system from a given state to a final state.

The proposed method relies on the reduction of the first order optimality conditions to a Fredholm integral equation of the second kind. The resulting equation will be solved by a multigrid method of the second kind, as proposed in (Hemker and Schippers, 1981; Hackbusch, 1985). The cost of this method is dominated by the evaluation of the kernel of the integral, which involves solving the linearized state and the corresponding adjoint equation. For the evaluation of these two subproblems a space-time multigrid approach of the first kind will be applied, as proposed in (Janssen and Vandewalle, 1993, 1996). Combining both multigrid methods results in a nested multigrid algorithm where an outer method is used for solving the integral equation and an inner method to evaluate the integral kernel.

Multigrid has proved highly successful for a variety of problems (Hackbusch, 1985; Trottenberg et al., 2001). Multigrid algorithms of the first and the second kind for solving time-periodic parabolic boundary value problems have been developed in (Horton and Vandewalle, 1995; Vandewalle and Piessens, 1993; Hackbusch, 1981). For their use towards PDE-constrained optimization we refer to the review in (Borzí and Schulz, 2009). Often the multigrid strategy is designed to exploit the structure of the underlying PDE-system as in the one

Dirk Abbeloos
Department of Computerscience, K.U.Leuven, Celestijnenlaan 200, 3001 Heverlee, Belgium

Moritz Diehl
Electrical Engineering Department ESAT, K.U.Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

Michael Hinze
Fachbereich Mathematik, Universität Hamburg, Bundesstr. 55, 20146 Hamburg, Germany

Stefan Vandewalle
Department of Computerscience, K.U.Leuven, Celestijnenlaan 200, 3001 Heverlee, Belgium

shot approach presented in (Borzí, 2003), but not specific the structure of the optimality system. The nested technique we propose here is based on the technique described in (Hackbusch, 1979) and does exploit the structure of both the PDE and the optimization problem.

In the next section we introduce the optimization problem. Its optimality conditions are derived and transformed into an integral equation in §3. Afterwards, we discretize the problem in §4. Next, the nested multigrid algorithm is presented in §5 and we illustrate the numerical performance of the nested multigrid method on two examples in §6.

2 Model Problem

We shall adhere to the optimal control notation, i.e. the state and control of the system are respectively denoted by y and u. The spatial domain is given by an open, bounded domain $\Omega \subset \mathbb{R}^d$ where $\partial\Omega = \Gamma$ denotes its boundary. Due to the periodicity, the time domain can be represented by only considering the reference interval $[0, T]$ where T is a predefined constant, called the period. The state y is a function of space and time. The domain of the control depends on the application. Here, we focus on distributed control, which makes u space and time dependent. The function spaces to which the state and the control belong are denoted by Y and U, respectively. These spaces will be defined more precisely further on. In this notational framework our model problem is given by,

$$\begin{cases} \min_{y,u} J(y) + \frac{\alpha}{2} \|u\|_U^2 \\ \text{s.t.} \qquad \mathbf{e}(y,u) = F. \end{cases} \tag{1}$$

Here, $J(y)$ is a cost functional, α is a positive constant weighting the control cost and equation $\mathbf{e}(y,u) = F$ is an abstract representation of the time-periodic PDE constraint.

2.1 The PDE-constraint

We consider the linear, parabolic PDE constraint,

$$\partial_t y - \triangle y - u = f \qquad \text{in } \Omega \times (0,T),$$

where $\triangle$ denotes the Laplacian. On $\partial\Omega \times (0,T)$ we either set Robin boundary conditions or homogeneous Dirichlet boundary conditions. The periodicity of the problem is imposed by equating the states at the beginning and the end of the reference interval, i.e.

$$y|_{t=T} = y|_{t=0} \qquad \text{on } \Omega.$$

The operator $\mathbf{e}$ corresponds to the weak formulation of the PDE, the spatial boundary conditions and the time-periodicity constraint. For a proper setting we introduce the following Gelfand triple of spatial function spaces,

$$H(\Omega) \hookrightarrow L^2(\Omega) \hookrightarrow H^*(\Omega),$$

where the space $H(\Omega)$ equals $H_0^1(\Omega)$ or $H^1(\Omega)$ for respectively Dirichlet and Robin boundary conditions. For brevity reasons, the Ω-dependency will be dropped from our notation. The symbol $\hookrightarrow$ denotes a continuous embedding and H^* denotes the dual space of H with duality pairing $\langle \cdot, \cdot \rangle_{H^*,H}$. The inner product on $L^2(\Omega)$ is given by $(\cdot, \cdot)_L$.

The state space is

$$Y = \left\{ v \mid v \in L^2((0,T), H),\ \partial_t v \in L^2((0,T), H^*) \right\}.$$

Note that we do not include the periodicity in the state space, but leave it as a separate constraint. This will result in a separate multiplier in the optimality conditions later on. Thanks to the continuous embedding,

$$Y \hookrightarrow C\left([0,T]; L^2(\Omega)\right),$$

the evaluation of y at a given instance of time is well defined. This is necessary to make the time-periodicity constraint meaningful. We require $f \in L^2\left((0,T); L^2(\Omega)\right)$, $u \in L^2\left((0,T); L^2(\Omega)\right)$. We also define the space $P = L^2((0,T); H^*(\Omega)) \times L^2(\Omega)$, where the first and second component are used for testing respectively the PDE and the time-periodicity constraint. The exact formulation of the operator $\mathbf{e}$ is then given by,

$$\mathbf{e} : Y \times U \to P : (y,u) \to \mathbf{e}(y,u), \tag{2}$$

with $e(y,u)$ given by,

$$\begin{bmatrix} \int_0^T \left(\langle \partial_t y, \cdot \rangle_{H^*,H} + a(t,y,\cdot) - (u,\cdot)_L \right) dt \\ (y(0) - y(T), \cdot)_L \end{bmatrix}^T \tag{3}$$

and

$$a(t,y,\phi) = (\nabla y, \nabla \phi)_L + \text{boundary terms}.$$

The right-hand side F in (1) is defined as,

$$F = \begin{bmatrix} \int_0^T (f,\cdot)_L \, dt + \text{boundary terms} \\ 0 \end{bmatrix}^T.$$

This completes the weak formulation of the PDE-constraint. We assume for every u a unique corresponding solution y to $\mathbf{e}(y,u) = F$, which leads us to define the solution operator,

$$y = \mathbf{S}(u).$$

2.2 The objective

We consider objective functionals of the form,

$$J(y) = \frac{1}{2}\|y-z\|^2_{L^2((0,T);L^2(\Omega))} + \frac{\nu}{2}\|y(T)-z_\Omega\|^2_{L^2(\Omega)} \quad (4)$$

where $z \in L^2\left((0,T);L^2(\Omega)\right)$ and $z_\Omega \in L^2(\Omega)$ are specified target functions, and $\nu \geq 0$. The parameter α in the model problem (1) allows tuning the trade-off between the conflicting objectives of both minimizing the tracking error and the control cost. For example, decreasing the value of α results in improved tracking performance at the cost of an increased control effort.

2.3 The reduced model problem

We use the solution operator $\mathbf{S}$ to transform problem (1) into the reduced problem,

$$\min_u \hat{J}(u) := J\left(\mathbf{S}(u)\right) + \frac{\alpha}{2}\|u\|^2_U \quad (5)$$

where $\hat{J}$ is the reduced cost functional. In this way we obtain an unconstrained optimization problem for the control variable u.

3 Reformulation of the optimality conditions as an integral equation.

We formally derive the optimality conditions of problem (1) with the Lagrangian technique. For details see (Tröltzsch, 2010; Hinze et al., 2009).

3.1 The Lagrangian

The Lagrangian associated to problem (1) is given by,

$$\mathcal{L} : Y \times U \times P^* \to \mathbb{R} : (y,u,p) \to \mathcal{L}(y,u,p)$$

with

$$\mathcal{L}(y,u,p) = J(y) + \frac{\alpha}{2}\|u\|^2_U - \langle p,\ \mathbf{e}(y,u) - F\rangle_{P^*,P}.$$

The function p is called the adjoint variable. Note that p has two components, one related to the PDE-constraint and another one to the periodicity constraint.

3.2 The optimality conditions

The first order optimality conditions are sufficient conditions and are given by,

$$\begin{bmatrix} \langle \mathcal{L}_y(y,u,p), \phi\rangle_{Y^*,Y} \\ \langle \mathcal{L}_u(y,u,p), \psi\rangle_{U^*,U} \\ \langle \mathcal{L}_p(y,u,p), \varphi\rangle_{P,P^*} \end{bmatrix} = 0,$$

for all $(\phi,\psi,\varphi) \in Y \times U \times P^*$. The equivalent expressions in terms of the operators and functionals appearing in our model problem, are deduced as follows,

$$\begin{aligned} &\langle \mathcal{L}_y(y,u,p),\ \phi\rangle_{Y^*,Y} \\ &\quad = \langle J_y(y),\ \phi\rangle_{Y^*,Y} - \langle p,\ \mathbf{e}_y(y,u)\phi\rangle_{P^*,P} \\ &\quad = \langle J_y(y),\ \phi\rangle_{Y^*,Y} - \left\langle \mathbf{e}_y^*(y,u)p,\ \phi\right\rangle_{Y^*,Y} \\ &\quad = \left\langle J_y(y) - \mathbf{e}_y^*(y,u)p,\ \phi\right\rangle_{Y^*,Y} \end{aligned}$$

$$\begin{aligned} &\langle \mathcal{L}_u(y,u,p),\ \psi\rangle_{U^*,U} \\ &\quad = (\alpha u,\ \psi)_U - \langle p,\ \mathbf{e}_u(y,u)\psi\rangle_{P^*,P} \\ &\quad = (\alpha u,\ \psi)_U - \langle \mathbf{e}_u^*(y,u)p,\ \psi\rangle_{U^*,U} \\ &\quad = \langle (\alpha u,\ \cdot)_U - \mathbf{e}_u^*(y,u)p,\ \psi\rangle_{U^*,U} \end{aligned}$$

$$\begin{aligned} &\langle \mathcal{L}_p(y,u,p),\ \varphi\rangle_{P,P^*} \\ &\quad = -\langle \mathbf{e}(y,u) - F,\ \varphi\rangle_{P,P^*} \end{aligned}$$

Summarizing we obtain,

$$\begin{cases} \mathbf{e}_y^*(y,u)\,p = J_y(y) & \text{in } Y^* \\ \mathbf{e}_u^*(y,u)\,p = (\alpha u,\ \cdot)_L & \text{in } U^* \\ \quad \mathbf{e}(y,u) = F & \text{in } P. \end{cases} \quad (6)$$

These three equations are often referred to as the adjoint, design and state equation, respectively. The first and last of these equations are boundary value problems in space and time. This is obvious for the state equation, while for the adjoint equation we refer to the derivation in the appendix.

Since $U = L^2\left((0,T);L^2(\Omega)\right)$ which is equal to U^*, the second condition is equivalent to

$$\nabla \hat{J}(u) = \alpha u - \mathbf{e}_u^*(y,u)p = 0, \quad (7)$$

where we have identified $\mathbf{e}_u^*(y,u)p$ with its Riesz representator. This expresses then the optimality of the solution.

3.3 The integral equation

The PDE-constraint in (1) can be rewritten as

$$\mathbf{e}(y,u) = Ay + Bu = F.$$

where A and B are linear operators. After reordering of (6), this results in the following symmetric system,

$$\begin{cases} J_y(y) & -A^*p & & = 0 \\ -Ay & & -Bu & = -F \\ & -B^*p & +\alpha u & = 0, \end{cases}$$

representing the first order optimality conditions. This symmetric problem can be reduced to a Fredholm integral equation of the second kind for the control variable. This is done by solving the first and the second equation for respectively the adjoint and the state, i.e.,

$$p = A^{-*}J_y(y) \quad \text{and} \quad y = A^{-1}\left(F - Bu\right),$$

and substituting these expressions in the last condition,

$$B^*A^{-*}J_y\left(A^{-1}Bu\right) + \alpha u = B^*A^{-*}J_y\left(A^{-1}F\right).$$

This integral equation is the infinite dimensional extension of the well known Schur-complement.

3.4 Conclusion

In this section we showed the equivalence of finding a minimizer of (1) and solving an integral equation for the control, of the form,

$$\alpha u + \mathcal{K}u = d. \tag{8}$$

Because finding an analytical solution of (8) is generally not possible, we will focus further on on numerical methods. In the next section, we first discuss the discretization of the integral equation.

4 Discretization

Due to the dense nature of the integral equation, we do not intend to discretize it directly. In stead, we will discretize it implicitly by performing the numerical analogue of the reduction process, given in §3.3. More precisely, we only discretize the non-reduced system of PDEs (6) and evaluate every application of the integral operator by solving for the different equations as explained in the reduction process. In this setting it is necessary to preserve the adjoint properties of the different operators also on the discrete level. If one fails to do so, the approximation of the reduced Hessian will be insufficiently accurate.

In addition, our approach will lead to a discretization that commutes with the optimization. More precisely, both the first optimize then discretize (OD) strategy and the first discretize then optimize (DO) strategy will produce identical results.

In order to archieve our goals, we adhere to Galerkin methods in space and time (Becker et al., 2007; Meidner and Vexler, 2007, 2008). More precisely, we choose a continuous Galerkin method of degree one (cg-1) in space, and a discontinuous Galerkin method of zero degree (dg-0) in time. This allows to use the inner product of the space $L^2((0,T);L^2(\Omega))$ on both the continuous and the discrete level.

4.1 A continuous Galerkin method in space

For simplicity, we consider a rectangular spatial domain Ω with boundaries parallel to the coordinate axes. We assume a coarse, regular partitioning of the spatial domain into N_e open quadrilaterals

$$\mathcal{T} = \{K_n\}_{n=1}^{N_e} \quad \text{with} \quad \bar{\Omega} = \bigcup_{K\in\mathcal{T}} \bar{K}.$$

From this coarse mesh $\mathcal{T}$, which we refer to as $\mathcal{T}_1$, we define a hierarchy of finer spatial meshes denoted by, $\{\ \mathcal{T}_l\ \}_{l=1}^{L}$, where $\mathcal{T}_{l+1}$ is recursively obtained from $\mathcal{T}_l$ by a uniform refinement, i.e. every quadrilateral is split into four new quadrilaterals by bisecting every edge. The index l is called the level, and indicates the fineness of the mesh. With x_i $(i = 1 \dots N_l)$ we denote the vertices of the partition $\mathcal{T}_l$. For every level l, we define a finite dimensional approximation of $H(\Omega)$, i.e., $H_l(\Omega)$, the space of piecewise bilinear finite elements on $\mathcal{T}_l$. By construction the above hierarchy of discrete function spaces is conformal to H and nested, i.e. $H_1 \subset H_2 \subset \ldots \subset H_L \subset H$. The nesting implicitly defines canonical interpolation operators between the different meshes. More precisely, the identity operator is the natural choice for the interpolation operator. As a basis for the discrete spaces, we choose the standard Lagrangian finite elements. With N_l the number of nodes in the mesh $\mathcal{T}_l$ and x_i the coordinate of the i-th node, these are defined as,

$$\phi_i(x_j) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \quad \forall\, i,j:\ 1 \leq i,j \leq N_l$$

with $\phi_i \in H_l$.

4.2 A discontinuous Galerkin method in time

For discretizing the time dimension we consider the following set of time points,

$$0 = t_0 < t_1 < \ldots < t_{M-1} < t_M = T$$

with a corresponding partitioning of the time interval,

$$[0,T] = \{0\} \cup I_1 \cup \ldots \cup I_M \qquad \text{where} \quad I_m = (t_{m-1}, t_m]\,.$$

The PDE will be discretized on each interval I_m, while the singleton $\{0\}$ is used for specifying the initial value or the time-periodicity constraint.

We create a hierarchy of refined time meshes by dividing every interval into four subintervals of equal length. The level of refinement is denoted by k, where $k = 1$ is the initial level. The number of intervals on level k will be denoted as $M_k = M4^k$. On every refinement level k we define a fully discrete function space,

$$W_k\left([0,T]\,; H_l\right) = \big\{ v \in L^2((0,T)\,; H_l) \mid \\ v|_{I_m} \in \mathbb{P}^r(I_m; H_l) \text{ with } 1 \le m \le M_k \text{ and } v(0) \in H_l \big\}\,.$$

where $\mathbb{P}^r(I_m; H_l)$ is the set of H_l-valued polynomials with maximal degree r. In short, this space contains the piecewise polynomials on the time partitioning, augmented with a value at the point $t = 0$. By setting $r = 0$, we obtain the necessary function spaces for the desired dg(0)-cg(1)-discretization. The fully discretized state, control and test spaces are defined as,

$$\begin{aligned} Y_{kl} &= W_k\left([0,T]\,; H_l\left(\Omega\right)\right), \\ U_{kl} &= W_k\left([0,T]\,; H_l\left(\Omega\right)\right), \\ P_{kl} &= W_k\left([0,T]\,; H_l\left(\Omega\right)\right) \times H_l\left(\Omega\right), \end{aligned}$$

respectively. We also define the notation,

$$y_m^- := \lim_{t \to 0^-} y(t_m + t) \quad \text{and} \quad y_m^+ := \lim_{t \to 0^+} y(t_m + t).$$

with $v_0^- = v(0)$ as a special case. The weak partial time derivative of a function $y \in Y_{kl}$ applied to a function $(\lambda,\ \mu) \in P_{kl}$ is then given by,

$$\sum_{m=1}^{M_k} \int_{I_m} \langle \partial_t y,\ \lambda \rangle_{H^*,H}\, dt + \left(y_{m-1}^+ - y_{m-1}^-,\ \lambda_{m-1}^+\right)_L . \quad (9)$$

When $r = 0$, the first terms vanishes, leaving only the terms taking into account the jump from one interval to the next. The basis functions ψ_m for the dg(0) discretization are the piecewise constant functions, i.e.

$$\psi_m(t) = \begin{cases} 1 & t \in I_m \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \psi_0(t) = \begin{cases} 1 & t = 0 \\ 0 & \text{else} \end{cases}$$

The elements of the state space Y_{kl} are then of the form,

$$y = \sum_{n=1}^{N_l} \sum_{m=0}^{M_k} y_{nm}\ \phi_n \psi_m \quad \text{with} \quad y_{nm} \in \mathbb{R}. \qquad (10)$$

4.3 The full discretization

In order to obtain a discrete version of the operator $\mathbf{e}$, we substitute the time derivative in (2) by expression (9) and, in addition we replace all function spaces by their discrete analogues. The discrete first order conditions (6) for $(y,u,p) \in Y_{kl} \times U_{kl} \times P_{kl}^*$ are then given by,

$$\begin{cases} \tilde{\mathbf{e}}_y^*\left(y,u\right) p = J_y\left(y\right) & \text{in } Y_{kl} \\ \tilde{\mathbf{e}}_u^*\left(y,u\right) p = \left(\alpha u, \cdot\right)_L & \text{in } U_{kl} \\ \quad \tilde{\mathbf{e}}\left(y,u\right) = F & \text{in } P_{kl}\ . \end{cases} \qquad (11)$$

with $\tilde{\mathbf{e}}$ a discrete approximation to $\mathbf{e}$. Next we rewrite the discrete operators in (11) in terms of the representation (10). The variables y_{nm} will be stacked in a vector $\mathbf{y} \in \mathbb{R}^{M_k N_l}$,

$$\mathbf{y} = \left[y_{11}, \ldots, y_{1M_k}, y_{21}, \ldots y_{N_l M_k}\right]^T$$

Note that we drop the terms y_{n0} as they will be eliminated later on. A similar vector notation is selected for the discrete control u. As for the discretization of the adjoint $p = (\lambda, \mu)$, we will use the notation $\mathbf{p}$ for the discretization of λ. We are now ready to rewrite the discrete optimality conditions in matrix-vector notation using the vectors $\mathbf{y}$, $\mathbf{u}$ and $\mathbf{p}$.

First, we consider the weak formulation of the time derivative combined with the periodicity constraint in (2), i.e.

$$\int_0^T \langle \partial_t y,\ \lambda \rangle_{H^*,H}\ dt + \left(y(0) - y(T), \mu\right)_L . \qquad (12)$$

Substituting the discrete time derivative (9) and the representation (10) of the discrete state y, yields,

$$\begin{aligned} &\textstyle\sum_{m=1}^{M_k} \int_{I_m} \langle \partial_t y,\ \lambda \rangle_{H^*,H}\, dt \\ &+ \textstyle\sum_{m=1}^{M_k} \left(y_{m-1}^+ - y_{m-1}^-,\ \lambda_{m-1}^+\right)_L + \left(y_0^- - y_M^-,\ \mu\right)_L \\ &= \textstyle\sum_{n,i=1}^{N} \sum_{m,j=1}^{M} y_{nm}\lambda_{ij}\ \ (\phi_n,\phi_i)_L\ \ \int_0^T \frac{d\psi_m}{dt}\psi_j\ dt \\ &+ \textstyle\sum_{n,i=1}^{N} \sum_{m=1}^{M} \left(y_{nm} - y_{nm-1}\right) \lambda_{im}\left(\phi_n,\phi_i\right)_L \\ &+ \textstyle\sum_{n,i=1}^{N} \left(y_{n0} - y_{nM}\right) \mu_i\left(\phi_n,\phi_i\right)_L . \end{aligned} \qquad (13)$$

We will eliminate the terms y_{n0} by using the discretized weak periodicity constraint in (2). Testing this condition with different discrete μ yields,

$$\sum_{n=1}^{N} y_{n0}\left(\phi_n,\phi_i\right)_L = \sum_{n=1}^{N} y_{nM}\left(\phi_n,\phi_i\right)_L .$$

After substituting the above condition in (13) and simplifying we obtain the following discrete representation for (12):

$$\lambda^T \left(A_\mathrm{s} \otimes B_\mathrm{t}\right) \mathbf{y},$$

where $\otimes$ is the Kronecker product, and A_s and B_t are matrices defined as,

$$A_{\mathrm{s}ij} = \left(\phi_i, \phi_j\right)_L, \quad \text{and} \quad B_\mathrm{t} = \begin{bmatrix} 1 & & & -1 \\ -1 & \ddots & & \\ & \ddots & 1 & \\ & & -1 & 1 \end{bmatrix}.$$

The remaining terms in (2) are discretized as follows,

$$\int_0^T \left(u,\ \lambda\right)_L \ dt = \lambda^T \left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{u},$$

$$\begin{aligned} \int_0^T a\left(t, y, \lambda\right) dt &= \left(\nabla y, \nabla \lambda\right)_L + \text{ boundary terms} \\ &= \lambda^T \left(B_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{y}. \end{aligned}$$

Here, we used the matrices A_t and B_s defined as

$$A_{\mathrm{t}ij} = \int_0^T \psi_i \ \psi_j \ dt \quad \text{for} \quad 1 \le i, j \le M,$$

$$B_{\mathrm{s}ij} = \left(\nabla \phi_i, \nabla \phi_j\right)_L + \text{boundary terms}.$$

Collecting all terms yields the following representation of the discrete operator $\tilde{\mathbf{e}}$ in (11),

$$\left(A_\mathrm{s} \otimes B_\mathrm{t} + B_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{y} - \left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{u}.$$

In order to derive the adjoint operator $\tilde{\mathbf{e}}^*$ and find the discretization of the first equation in (11), we start from the first expression in (13), i.e.

$$\begin{aligned} &\textstyle\sum_{m=1}^M \int_{I_m} \left\langle \partial_t y,\ \lambda \right\rangle_{H^*,H} dt \\ &+ \textstyle\sum_{m=1}^M \left(y_{m-1}^+ - y_{m-1}^-,\ \lambda_{m-1}^+\right)_L + \left(y_0^- - y_M^-,\ \mu\right)_L. \end{aligned}$$

Applying integration by parts and reordering, results in

$$\begin{aligned} -\textstyle\sum_{m=1}^M \int_{I_m} &\left\langle \partial_t \lambda,\ y \right\rangle_{H^*,H} dt \\ &+ \left(y_M^-,\ \lambda_M^-\right)_L + \textstyle\sum_{m=1}^{M-1} \left(y_m^-,\ \lambda_m^- - \lambda_m^+\right)_L \\ &+ \left(y_0^-,\ \lambda_0^+\right)_L + \left(y_0^-,\ \mu\right)_L - \left(y_M^-,\ \mu\right)_L. \end{aligned}$$

Elimination of y_0^-, reduces this to,

$$\begin{aligned} -\textstyle\sum_{m=1}^M \int_{I_m} &\left\langle \partial_t \lambda,\ y \right\rangle_{H^*,H} dt \\ &+ \textstyle\sum_{m=1}^{M-1} \left(y_m^-,\ \lambda_m^- - \lambda_m^+\right)_L \\ &+ \left(y_M^-,\ \lambda_M^- - \lambda_0^+\right)_L. \end{aligned}$$

In our discrete framework this is equivalent to,

$$\mathbf{y}^T \left(A_\mathrm{s} \otimes B_\mathrm{t}^T\right) \lambda.$$

Note that matrix A_s is symmetric.

The remaining terms in (11) can be discretized in a similar way. This eventually results in the following system of discrete first order conditions corresponding to (11),

$$\begin{cases} \left(A_\mathrm{s} \otimes B_\mathrm{t}^T + B_\mathrm{s}^T \otimes A_\mathrm{t}^T\right) \mathbf{p} \\ \qquad - \left(A_\mathrm{s} \otimes \left(A_\mathrm{t} + \nu E_\mathrm{t}\right)\right) \mathbf{y} = Z_\mathrm{st} \\ \alpha \left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{u} + \left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{p} = 0 \\ \left(A_\mathrm{s} \otimes B_\mathrm{t} + B_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{y} \\ \qquad - \left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{u} = F_\mathrm{st}, \end{cases} \tag{14}$$

where the matrix E_t is defined as $E_{\mathrm{t}ij} = \delta_{i,M}\delta_{j,M}$ with $\delta_{k,l}$ the Kronecker delta. Furthermore, the vectors F_st and Z_st are given by,

$$\begin{aligned} F_{\mathrm{st}\ (nM_k+m)} &= F\left(\phi_n \psi_m\right), \\ Z_{\mathrm{st}\ (nM_k+m)} &= \int_0^T \left(-z,\ \phi_n \psi_m\right)_L dt \\ &\quad + \nu \delta_{m,M} \left(-z_\Omega,\ \phi_n\right)_L. \end{aligned}$$

5 The nested multigrid solver

We shall present a nested multigrid algorithm for solving the integral equation (8). The outer iteration is a multigrid method of the second kind and solves the integral equation, while the inner iteration is a multigrid method of the first kind, and is used to evaluate the various components of the integral kernel.

As is well known, the multigrid strategy involves two complementary components, a smoothing step and a coarse grid correction step. In a multigrid method of the second kind the smoothing step may sometimes be discarded, when the integral operator possesses inherent smoothing properties. Examples of such operators are the inverse parabolic PDE operator and its adjoint. Also the kernel $\mathcal{K}$ of the integral operator (8) has this property.

5.1 A multigrid method of the first kind for evaluating the integral kernel.

As already mentioned the kernel of the integral operator in (8) is evaluated by performing the numerical analogue of the reduction process given in §3.3. More precisely, in order to evaluate the discretized integral equation for a given control $\mathbf{u}$, we first solve the third equation of (14) for the state $\mathbf{y}$, i.e.

$$\mathbf{y} = \mathcal{A}^{-1} \left(\left(A_\mathrm{s} \otimes A_\mathrm{t}\right) \mathbf{u} + F_\mathrm{st}\right),$$

where $\mathcal{A} = (A_\mathrm{s} \otimes B_\mathrm{t} + B_\mathrm{s} \otimes A_\mathrm{t})$. Then we solve the first equation of (14) for the adjoint $\mathbf{p}$, i.e.

$$\mathbf{p} = \mathcal{A}^{-*} \left((A_\mathrm{s} \otimes (A_\mathrm{t} + \nu E_\mathrm{t})) \, \mathbf{y} + Z_\mathrm{st} \right),$$

and finally, evaluating the left-hand side of the second equation of (14) yields an evaluation of the discretized integral equation, i.e.

$$\alpha \, (A_\mathrm{s} \otimes A_\mathrm{t}) \, \mathbf{u} + (A_\mathrm{s} \otimes A_\mathrm{t}) \, \mathbf{p}.$$

The computational cost of this evaluation is dominated by sequentially solving the state and adjoint equation. These problems are time-periodic boundary value problems corresponding to a parabolic PDE. Here, both problems are solved with the space-time multigrid method first presented in (Vandewalle and Piessens, 1993), and later on extended towards finite element methods in (Janssen and Vandewalle, 1996) and (Janssen and Vandewalle, 1993). The algorithm is an acceleration of the dynamic iteration method with a multigrid strategy in the space dimension.

The multigrid method uses a series of operators $\mathcal{A}^{(kl)}$ and $\mathcal{A}^{*(kl)}$, where the parameters k and l refer to the underlying discrete function spaces, as explained in §4.3. To solve the model problem on a given grid with parameters K and L, the method considers a hierarchy of semi-coarsened discrete problems. More precisely the method uses semi-coarsening in space, i.e. a hierarchy of discrete problems where k remains equal to K and l varies from L to 1. An outline of the method is given in Algorithm 1, where the parameter γ determines the cycle type.

Algorithm 1: mgm1st(k, l, $\mathcal{A}^{(kl)}$, $\mathbf{f}^{(kl)}$, $\mathbf{y}^{(kl)}$)

```
if l = 1 then
    Solve on coarsest grid
        A^(k1) y^(k1) = f^(k1)
else
    Perform μ1 smoothing iterations
        A^(kl) y^(kl) = f^(kl)
    Compute the residual
        r^(kl) ← f^(kl) − A^(kl) y^(kl)
    Restrict the residual to the lower level
        f^(kl−1) ← R^(kl) r^(kl) and y^(kl−1) ← 0
    Perform recursion
    for γ times do
        mgm1st(k, l − 1, A^(kl−1), f^(kl−1), y^(kl−1))
    end
    Interpolate the correction and correct
        y^(kl) ← y^(kl) + P^(kl) y^(kl−1)
    Perform μ2 smoothing iterations
        A^(kl) y^(kl) = f^(kl)
end
```

In order to fully characterize our implementation we shall specify the different multigrid components in the remaining of this section. First, the coarse grid operators are selected to be rediscretizations of the original problem on the coarser meshes. As direct solver on the coarsest grid we used an optimized sparse direct solver.

As a smoother we choose a four color timeline Gauss-Seidel method. This method is derived from the corresponding four color Gauss-Seidel iteration by grouping all degrees of freedom linked to the same spatial node. This comes down to solving a time discretized ODE with a time-periodicity constraint at each nodel point. This can easily be done with an optimized direct sparse solver. To formulate this iteration algebraically, we apply the standard four color splitting on the matrices A_s and B_s, resulting in,

$$B_\mathrm{s} = B_\mathrm{s}^+ - B_\mathrm{s}^- \quad \text{and} \quad A_\mathrm{s} = A_\mathrm{s}^+ - A_\mathrm{s}^-.$$

The timeline smoother is then obtained by

$$(A_\mathrm{s}^+ \otimes B_\mathrm{t} + B_\mathrm{s}^+ \otimes A_\mathrm{t}) \, \mathbf{y}^{(\nu)} = \mathbf{f} + (A_\mathrm{s}^- \otimes B_\mathrm{t} + B_\mathrm{s}^- \otimes A_\mathrm{t}) \, \mathbf{y}^{(\nu-1)}.$$

To transfer a discrete approximation $\mathbf{y}^{(kl)}$ from Y_{kl-1} to Y_{kl}, we use the canonical interpolation operator. Algebraically this operation is given by,

$$\mathcal{P}^{(kl)} = P_\mathrm{s}^{(l)} \otimes I_\mathrm{t}^{(k)}$$

with $P_\mathrm{s}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ the bilinear prolongation matrix and $I_\mathrm{t}^{(k)} \in \mathbb{R}^{m_k \times m_k}$ the identity matrix. As restriction we used the transpose of the prolongation operator, i.e.

$$\mathcal{R}^{(kl)} = \mathcal{P}^{(kl)T} = P_\mathrm{s}^{(l)T} \otimes I_\mathrm{t}^{(k)}.$$

To speed up the evaluation of the integral kernel, we implemented the full multigrid method to evaluate both boundary value problems, see Algorithm 2 for a reference implementation. Technical details as for example, the number of post V-cycles are not considered here. Furthermore, we took over the interpolation and restriction operators from Algorithm 1.

Finally, we can formulate the method to evaluate the integral equation. It is given in Algorithm 3. Here, the first call to Algorithm 2 solves the state equation while the second call solves the adjoint equation. The result of the evaluation is stored in the vector $\mathbf{r}^{(kl)}$. In order to evaluate only the kernel of the integral, the terms F_st and Z_st should be ignored.

Algorithm 2: Fmgm1st(k, L, $\mathcal{A}^{(kL)}$, $\mathbf{f}^{(kL)}$, $\mathbf{y}^{(kL)}$)

for $l = L$ *down to* 1 **do**
  Restrict the rhs to the lower level
    $\mathbf{f}^{(kl-1)} \leftarrow \mathcal{R}^{(kl)}\mathbf{f}^{(kl)}$
end
Solve on coarsest grid
  mgm1st(k, 1, $\mathcal{A}^{(k1)}$, $\mathbf{f}^{(k1)}$, $\mathbf{y}^{(k1)}$)
for $l = 1$ *up to* L **do**
  Interpolate the approximation of the solution
    $\mathbf{y}^{(kl)} \leftarrow \mathbf{y}^{(kl)} + \mathcal{P}^{(kl)}\mathbf{y}^{(kl-1)}$
  Perform a multigrid iteration
    mgm1st(k, l, $\mathcal{A}^{(kl)}$, $\mathbf{f}^{(kl)}$, $\mathbf{y}^{(kl)}$)
end

Algorithm 3: Eval(k, l, $\mathbf{u}^{(kl)}$, $\mathbf{r}^{(kl)}$)

Assemble the rhs of the state equation
  $\mathbf{f}^{(kl)} \leftarrow (A_{\mathrm{s}}^{(kl)} \otimes A_{\mathrm{t}}^{(kl)})\mathbf{u}^{(kl)} + F_{\mathrm{st}}$
Evaluate the state equation
  Fmgm1st(k, l, $\mathcal{A}^{(kl)}$, $\mathbf{f}^{(kl)}$, $\mathbf{y}^{(kl)}$)
Assemble the rhs of the adjoint equation
  $\mathbf{f}^{(kl)} \leftarrow \left(A_{\mathrm{s}}^{(kl)} \otimes \left(A_{\mathrm{t}}^{(kl)} + \nu E_{\mathrm{t}}^{(kl)}\right)\right)\mathbf{y}^{(kl)} + Z_{\mathrm{st}}$
Evaluate the adjoint equation
  Fmgm1st(k, l, $\mathcal{A}^{*(kl)}$, $\mathbf{f}^{(kl)}$, $\mathbf{r}^{(kl)}$)

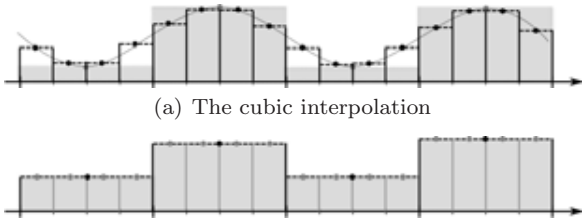
5.2 A multigrid method of the second kind for solving the integral equation

To solve the integral equation (8) we use a multigrid method of the second kind. Different variants of this method were developed independently by Hackbusch (Hackbusch, 1985) and Schippers (Hemker and Schippers, 1981). An outline of the variant by Hackbusch (v1) and by Schippers (v2) are given in Algorithm 4 and 5 respectively. Note the difference in the smoothing step compared to Algorithm 1. The multigrid method of the second kind exploits the smoothing characteristics of the integral operator by applying a simple Picard iteration in Algorithm 4 or an adaptation thereof in Algorithm 5.

As mentioned, the integral operators required on the different levels are not constructed explicitly but only evaluated by using Algorithm 3. As a solver on the coarsest level we choose a conjugate gradient method. This method proved to converge in just a few, cheap iterations.

In order to transfer a discrete approximation $\mathbf{y}^{(kl)}$ from Y_{k-1l-1} to Y_{kl} a tensor product of the canonical interpolation operator in space and a cubic interpolation operator in time with periodic boundary conditions is used. Algebraically this operation is given by,

$$\tilde{\mathcal{P}}^{(kl)} = P_{\mathrm{s}}^{(l)} \otimes P_{\mathrm{t}}^{(l)}.$$



(a) The cubic interpolation

(b) The restriction

Fig. 1 An illustration of the action of the cubic prolongation P_{t} (a) and the canonical restriction R_{t} (b). For the prolongation (restriction), the symbols • and • denote a degree of freedom of the fine (coarse) or coarse (fine) grid respectively. In addition, the coarse (fine) grid approximation is represented by the gray rectangles. The cubic interpolation with periodic boundary conditions is illustrated by a solid line.

Here, $P_{\mathrm{s}}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the bilinear prolongation matrix and $P_{\mathrm{t}}^{(k)} \in \mathbb{R}^{m_k \times m_{k-1}}$ is the periodic, cubic prolongation matrix. Its actions is illustrated in Figure 1. As restriction operator we use

$$\tilde{\mathcal{R}}^{(kl)} = R_{\mathrm{s}}^{(l)} \otimes R_{\mathrm{t}}^{(k)}.$$

where $R_{\mathrm{s}}^{(l)} \in \mathbb{R}^{n_{l-1} \times n_l}$ is the boundary adapted full weighting matrix and $R_{\mathrm{t}}^{(k)} \in \mathbb{R}^{m_{k-1} \times m_k}$ is the scaled adjoint of the canonical interpolation operator. Its action is illustrated in Figure 1.

To speed up the outer iteration, a full multigrid variant similar to Algorithm 2 is used.

Algorithm 4: mgm2nd1(k, l, $d^{(kl)}$, $u^{(kl)}$)

if $l = 1$ **then**
  Solve on coarsest grid
    $\alpha\mathbf{u}^{(11)} + \mathcal{K}^{(11)}\mathbf{u}^{(11)} = \mathbf{d}^{(11)}$
else
  Apply a Picard iteration
    $\mathbf{u}^{(kl)} \leftarrow \frac{1}{\alpha}\mathbf{d}^{(kl)} - \frac{1}{\alpha}\mathcal{K}^{(kl)}\mathbf{u}^{(kl)}$
  Compute the residual
    $\mathbf{r}^{(kl)} \leftarrow \mathbf{d}^{(kl)} - \alpha\mathbf{u}^{(kl)} - \mathcal{K}^{(kl)}\mathbf{u}^{(kl)}$
  Restrict the residual to the lower level
    $\mathbf{d}^{(k-1l-1)} \leftarrow \tilde{\mathcal{R}}^{(kl)}\mathbf{r}^{(kl)}$ and $\mathbf{u}^{(k-1l-1)} \leftarrow 0$
  Perform the recursion
  for 2 *times* **do**
    mgm2nd1($k-1$, $l-1$, $\mathbf{d}^{(k-1l-1)}$, $\mathbf{u}^{(k-1l-1)}$)
  end
  Interpolate the correction and correct
    $\mathbf{u}^{(kl)} \leftarrow \mathbf{u}^{(kl)} + \tilde{\mathcal{P}}^{(kl)}\mathbf{u}^{(k-1l-1)}$
end

6 Numerical results

In this section we elaborate on the numerical performance of the nested multigrid method. More precisely,

Algorithm 5: mgm2nd2(k, l, $d^{(kl)}$, $u^{(kl)}$)

if $l = 1$ **then**
  Solve on coarsest grid
    $\alpha \mathbf{u}^{(11)} + \mathcal{K}^{(11)}\mathbf{u}^{(11)} = \mathbf{d}^{(11)}$
else
  Compute the residual
    $\mathbf{r}^{(kl)} \leftarrow \mathbf{d}^{(kl)} - \alpha\mathbf{u}^{(kl)} - \mathcal{K}^{(kl)}\mathbf{u}^{(kl)}$
  Perform restriction and smoothing
    $\mathbf{u}^{(kl)} \leftarrow \mathbf{u}^{(kl)} - \left(I - \tilde{\mathcal{P}}^{(kl)}\tilde{\mathcal{R}}^{(kl)}\right)\mathbf{r}^{(kl)}$
    $\mathbf{d}^{(k-1l-1)} \leftarrow$
    $\left(\left(I - \mathcal{K}^{(k-1l-1)}\right)\tilde{\mathcal{R}}^{(kl)} - \tilde{\mathcal{R}}^{(kl)}\mathcal{K}^{(kl)}\right)\mathbf{r}^{(kl)}$
    $\mathbf{u}^{(k-1l-1)} \leftarrow 0$
  Perform the recursion
  for 2 *times* **do**
    mgm2nd2($k-1$, $l-1$, $\mathbf{d}^{(k-1l-1)}$, $\mathbf{u}^{(k-1l-1)}$)
  end
  Interpolate the correction and correct
    $\mathbf{u}^{(kl)} \leftarrow \mathbf{u}^{(kl)} + \tilde{\mathcal{P}}^{(kl)}\mathbf{u}^{(k-1l-1)}$
end

we first numerically illustrate the convergence behavior in terms of the spatial discretization parameter h and the parameter α. Next, we compare the multigrid methods with Krylov subspace methods, i.e. the outer multigrid iteration is replaced by a Krylov subspace method, like the conjugate gradient method (CG) and the generalized minimal residual method (GMRES). All algorithms have been implemented in C++. The Krylov subspace implementations were taken from the software package GLAS (Meerbergen et al., 2009).

6.1 Overview of the test problems

To illustrate the numerical performance of the nested multigrid algorithm, we consider the following test problems.

- Example I is the homogeneous problem, i.e. we set $\nu = 0,\ z = 0,\ z_\Omega = 0$ and $f = 0$.
- In example II we chose a nontrivial, continuous source term $f = x^2(1-x)^2\ y^2(1-y)^2\ \sin(2\pi t)$. Furthermore, we set $\nu = 0,\ z = 0,$ and $z_\Omega = 0$.
- In example III we chose the unreachable target state $z = \chi_{\left[\frac{1}{4},\ \frac{3}{4}\right]}(t)\ \chi_{\left[\frac{1}{2},\ 1\right]^2}(x)$ where χ_S denotes the characteristic function of the set S. This target state is unreachable because $z \notin Y$. The other parameters were chosen to be $\nu = 0,\ f = 0,$ and $z_\Omega = 0$.

In addition, we imposed homogeneous Robin boundary conditions, i.e. we required $\frac{\partial y}{\partial \nu} + y = 0$ on $\partial\Omega \times (0, T)$ where $\frac{\partial \cdot}{\partial \nu}$ is the derivative in the direction of the outer unit normal $\nu(x)$ of $\partial\Omega$ at x. In every experiment the discrete approximation of the control was initialized with Gaussian white noise. An example solution of example III is shown in Figure 2.

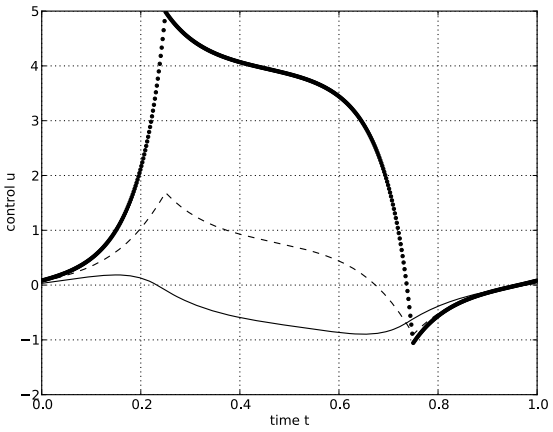


Fig. 2 The optimal control u of example III plotted as a function of time for the spatial coordinates $x = (0.25,\ 0.25)$ with a solid line $(-)$, $(0.5,\ 0.5)$ with a dashed line $(--)$ and $(0.75,\ 0.75)$ with dotted line $(\cdots)$.

6.2 The dependency of the convergence of the multigrid method on the fineness of the mesh

Figure 3 shows the convergence history of both nested multigrid methods, i.e. the variant of Hackbusch and Schippers denoted by v1 and v2 respectively, with different values for the spatial discretization parameter $h \in [\frac{1}{8}, \frac{1}{64}]$. The discretization parameter in the time dimension denoted by Δt, was chosen to satisfy the relation $\Delta t = h^2$. Furthermore, we set the parameter $\alpha = 0.01$. Figure 3 illustrates that the convergence of both nested multigrid methods improves with the fineness of the mesh, which makes the method especially suitable for very large scale problems. Table 1 reports the convergence factor ρ in function of the spatial grid parameter h for example I. Here, we define the convergence factor ρ as,

$$\rho = \sqrt[n]{\frac{\|\mathbf{r}_n\|_{L^2(\Omega\times[0,T])}}{\|\mathbf{r}_0\|_{L^2(\Omega\times[0,T])}}}, \tag{15}$$

with n the number of multigrid iterations applied and $\mathbf{r}_i$ the residual of the integral equation after i iterations. Clearly, the multigrid convergence improves dramatically for finer meshes. For example, a reduction of the parameter h from $\frac{1}{16}$ to $\frac{1}{64}$ improves the convergence rate by approximately a factor 100. Similar results were obtained in other experiments. The convergence factor of the nested multigrid method for example II and III are reported in Table 2.

Furthermore, the numerical experiments indicate that the multigrid variant of Hackbusch seems to converge faster than the variant of Schippers. This statement is confirmed by more elaborated experiments, although

Table 1 The convergence factor ρ of the multigrid algorithms in function of the discretization parameter h for example I.

	Example I	
h	v1	v2
$\frac{1}{8}$	$1.9\ 10^{-2}$	$1.9\ 10^{-2}$
$\frac{1}{16}$	$1.7\ 10^{-3}$	$6.3\ 10^{-3}$
$\frac{1}{32}$	$2.6\ 10^{-4}$	$5.5\ 10^{-4}$
$\frac{1}{64}$	$1.6\ 10^{-5}$	$2.5\ 10^{-5}$

Table 2 The convergence factor ρ of the multigrid algorithms in function of the discretization parameter h for example II and III.

	Example II		Example III	
h	v1	v2	v1	v2
$\frac{1}{8}$	$2.7\ 10^{-2}$	$2.7\ 10^{-2}$	$1.9\ 10^{-2}$	$1.9\ 10^{-2}$
$\frac{1}{16}$	$2.2\ 10^{-3}$	$1.2\ 10^{-2}$	$1.7\ 10^{-3}$	$6.3\ 10^{-3}$
$\frac{1}{32}$	$9.6\ 10^{-4}$	$1.7\ 10^{-3}$	$2.6\ 10^{-4}$	$5.4\ 10^{-4}$
$\frac{1}{64}$	$1.2\ 10^{-4}$	$1.8\ 10^{-4}$	$1.6\ 10^{-5}$	$2.4\ 10^{-5}$

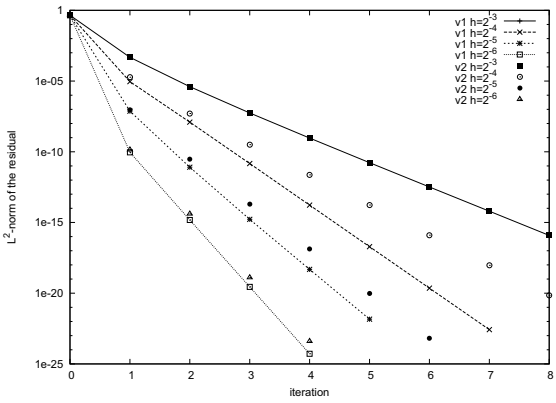


Fig. 3 The convergence history of the nested multigrid methods applied on example I with different values for the spatial mesh parameter h. The multigrid variant of Hackbusch is denoted by a solid line.

the variant of Schipper seems to be more robust. More precisely, there are cases where the method of Hackbusch diverges while the variant of Schippers is still convergent.

6.3 The dependency of the convergence of the multigrid method on the parameter α

It is known that the integral equation (8) is a compact perturbation of the identity operator, resulting in a mesh independent condition number, when discretized fine enough. Furthermore, the condition number depends strongly on the parameter α, more specific, a decrease of α results in a strong increase of the condition number and the other way round. As a consequence we can expect the multigrid method to deteriorate for decreasing values of α, which is confirmed by our numerical experiments.

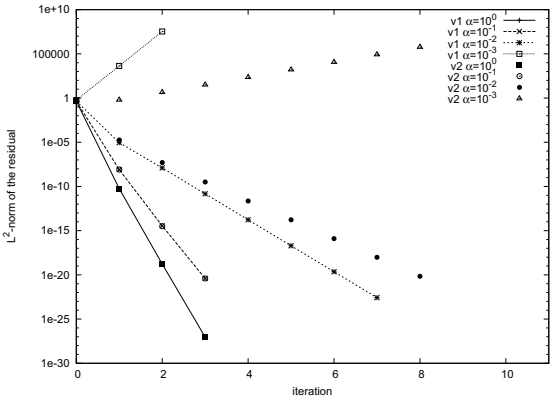


Fig. 4 The convergence history of the multigrid methods applied on example I with different values for the parameter α. The multigrid variant of Hackbusch is denoted by a solid line.

Figure 4 illustrates the convergence history of the nested multigrid methods applied on example I with different values for the parameter $\alpha \in \left[10^{-4}, 1\right]$. Here, we chose the discretization parameters as $h = 2^{-4}$ and $\Delta t = 2^{-8}$. Clearly, for decreasing values of α the convergences deteriorates and even diverges for $\alpha < 0.005$. The convergence factors of the multigrid methods for example I and II are given in Table 3 and Table 4 respectively. The numerical results indicate a severe influence of the parameter α on the convergence of the multigrid methods. In the optimal control framework, α is typically a fixed parameter. When α is not too small, the convergence rate is restored when the mesh is refined.

The numerical results indicate that the convergence rate of both multigrid variants is identical for large values of $\alpha \approx 1$, while for a smaller α the method of Hackbusch becomes relatively faster compared to the variant of Schippers. The convergence rates also indicate that for large values of α both methods are extremely fast. Often only 2 iterations will be needed to solve a given problem.

6.4 A comparison with Krylov subspace methods

To illustrate the effectiveness of the nested multigrid method, we repeated the numerical experiments from sections §6.2 and §6.3 with a CG and GMRS solver. Since the convergence of both Krylov subspace methods is solely determined by the condition number, we expect a mesh independent convergence behavior for

Table 3 The convergence factor ρ of the multigrid algorithms in function of the parameter α for example I. The symbol $*$ indicates that the method failed to converge.

	Example I	
α	v1	v2
1	$2.1\ 10^{-7}$	$2.1\ 10^{-7}$
0.1	$9.7\ 10^{-6}$	$9.6\ 10^{-6}$
0.01	$1.7\ 10^{-3}$	$6.3\ 10^{-3}$
0.001	*	*

Table 4 The convergence factor ρ of the multigrid algorithms in function of the parameter α for example III.

	Example III	
α	v1	v2
0.5	$5.8\ 10^{-7}$	$5.8\ 10^{-7}$
0.1	$9.7\ 10^{-6}$	$9.6\ 10^{-6}$
0.05	$4.9\ 10^{-5}$	$4.9\ 10^{-5}$
0.01	$1.7\ 10^{-3}$	$6.3\ 10^{-3}$

fixed α and a strong deterioration of the convergence rate for decreasing α. Both statements were confirmed in our numerical experiments as shown in Figure 5 and Figure 6. In the last figure we only plotted the convergence history of the CG-solver, because the GMRES-solver shows a similar result. From the experiments we conclude that the GMRES method converges faster in terms of number of iterations, but every iteration is also more expensive in terms of integral operator evaluations than the according CG-iteration. Comparing these results with Figures 3 and 4 indicates that for $\alpha \geq 10^{-2}$ the convergence of the nested multigrid method is much faster then of the Krylov subspace methods. In most cases only a few multigrid iterations are needed to solve a problem, where every multigrid iteration requires only two evaluations of the integral kernel on every level. Moreover, the convergence of the Krylov subspace methods are mesh independent while the convergence of the nested multigrid method improves with increasing fineness of the mesh.

7 Conclusion

This paper presented a nested multigrid algorithm for solving optimal control problems constrained by a time-periodic parabolic PDE. The method is based on the reduction of the optimality conditions to a Fredholm integral equation of the second kind. The resulting equation is then discretized using a space-time finite element framework, which preserves the structure of both the

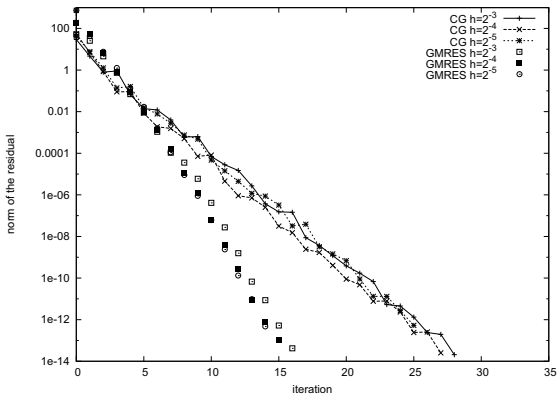


Fig. 5 The convergence history of the CG and GMRES solver applied to example I with different values for the spatial mesh parameter h. The solid line denotes the CG-solver, while the points corresponds to the GMRES-solver.

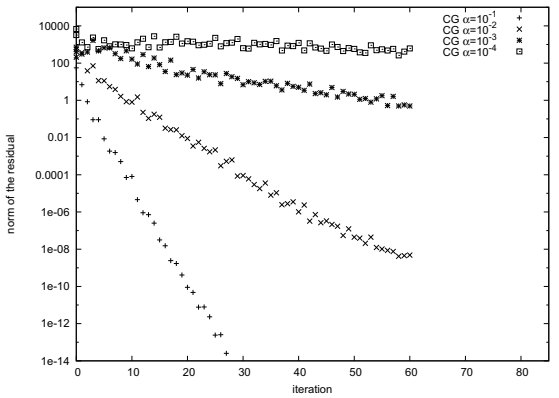


Fig. 6 The convergence history of the CG solver applied to example I with different values for the parameter α.

PDE-constraint and the optimality problem on the discrete level. The problem is solved with a nested multigrid method resulting in fast convergence. Moreover, the convergence factor improves with the fineness of the mesh, which makes the method especially suitable for very large scale problems.

Acknowledgements The authors would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), the K.U.Leuven Center of Excellence: Optimization in Engineering and the DEASE-project.

8 Appendix: Derivation of equations

In this appendix we derive the adjoint equation for the example included in the text, i.e. the first equation of

(6). The derivation starts from the expression,

$$\langle \mathcal{L}_y\left(y,u,p\right),\ \phi\rangle_{Y^*,Y} = \langle J_y(y),\ \phi\rangle_{Y^*,Y} - \langle p,\ \mathbf{e}_y(y,u)\phi\rangle_{P^*,P},$$

which after substituting expressions (2) and (4), results in,

$$\begin{aligned}&\langle \mathcal{L}_y\left(y,u,p\right),\ \phi\rangle_{Y^*,Y} = \\ &+\int_0^T c\left(y-z,\phi\right)_L - \langle \partial_t\phi,\lambda\rangle_{H^*,H} - a_y\left(t,y,\lambda\right)\phi\ dt \\ &+\left(y(T)-z_\Omega,\phi(T)\right)_L - \left(\phi(0)-\phi(T)_L,\mu\right)_L\end{aligned}$$

with $p=(\lambda,\mu)\in P^*$. The exact form of the functional $a_y(t,y,\lambda)$ is given by,

$$a_y\left(t,y,\lambda\right)\phi = \left(\nabla\phi,\nabla\lambda\right)_L + \text{boundary terms}.$$

The above expression for $a_y(t,y,\lambda)$ is missing some boundary conditions, which we do not include here for brevity reasons. In order to derive the adjoint of the derivative with respect to the time operator, we use the integration by parts formula, i.e.,

$$\begin{aligned}-\langle \partial_t\phi,\lambda\rangle_{H^*,H} &= \langle \partial_t\lambda,\phi\rangle_{H^*,H} \\ &\quad -\left(\lambda(T),\phi(T)\right)_L + \left(\lambda(0),\phi(0)\right)_L.\end{aligned}$$

After substitution in the original equation, this leads to the following expression,

$$\begin{aligned}&\langle \mathcal{L}_y\left(y,u,p\right),\ \phi\rangle_{Y^*,Y} = \\ &+\int_0^T \left(y-z,\phi\right)_L + \langle \partial_t\lambda,\phi\rangle_{H^*,H} - a_y\left(t,y,\lambda\right)\phi\ dt \\ &-\left(\lambda(T),\phi(T)\right)_L + \left(\lambda(0),\phi(0)\right)_L \\ &+\nu\left(y(T)-z_\Omega,\phi(T)\right)_L - \left(\phi(0)-\phi(T)_L,\mu\right)_L.\end{aligned}$$

To derive the adjoint equations in a simple form, we eliminate the variable μ by considering variations of $\phi(0)$ only. This leads to $\mu=\lambda(0)$ in the L^2-sense. Substituting this back and after simplification we obtain,

$$\begin{aligned}&\langle \mathcal{L}_y\left(y,u,p\right),\ \phi\rangle_{Y^*,Y} = \\ &+\int_0^T \left(y-z,\phi\right)_L + \langle \partial_t\lambda,\phi\rangle_{H^*,H} - a_y\left(t,y,\lambda\right)\phi\ dt \\ &+\nu\left(y(T)-z_\Omega,\phi(T)\right)_L - \left(\lambda(T)-\lambda(0),\phi(T)\right)_L.\end{aligned} \tag{16}$$

Finally, we may conclude that the above expression is the weak formulation of

$$\begin{cases} -\partial_t\lambda - \triangle\lambda - y = -z \qquad \text{in } \Omega\times(0,T) \\ \lambda(T) - \lambda(0) - \nu\ y(T) = -\nu\ z_\Omega \text{ on } \Omega, \end{cases}$$

This is a boundary value problem in space and time.

References

Becker, R., Meidner, D., and Vexler, B. (2007). Efficient numerical solution of parabolic optimization problems by finite element methods. *Optimization Methods and Software*, 22:813–833.

Borzí, A. (2003). Multigrid methods for parabolic distributed optimal control problems. *Journal of Computational and Applied Mathematics*, 157:365–382.

Borzí, A. and Schulz, V. (2009). Multigrid methods for PDE optimization. *SIAM review*, 51:361–395.

Hackbusch, W. (1979). On the fast solving of parabolic boundary control problems. *SIAM Journal on Control and Optimization*, 17:231–244.

Hackbusch, W. (1981). Fast numerical solution of time-periodic parabolic problems by a multigrid method. *SIAM Journal on Scientific and Statistical Computing*, 2:198–206.

Hackbusch, W. (1985). *Multigrid methods and applications*. Springer-verslag.

Hemker, P. and Schippers, H. (1981). Multiple grid methods for the solution of fredholm integral equations of the second kind. *Mathematics of computation*, 36:215–232.

Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. (2009). *Optimization with PDE constraints*. Springer-Verlag.

Horton, G. and Vandewalle, S. (1995). A space-time multigrid method for parabolic partial differential equations. *SIAM Journal on Scientific Computing*, 16:848–864.

Houska, B. and Diehl, M. (2010). Robustness and stability optimization of power generating kite systems in a periodic pumping mode. In *Proc. of the IEEE Multi-Conference on Systems and Control*, pages 2172 – 2177, Yokohama.

Houska, B., Logist, F., Impe, J. V., and Diehl, M. (2009). Approximate robust optimization of time-periodic stationary states with application to biochemical processes. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 6280 – 6285, Shanghai.

Janssen, J. and Vandewalle, S. (1993). Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM Journal on Numerical Analysis*, 33:456–474.

Janssen, J. and Vandewalle, S. (1996). Multigrid waveform relaxation on spatial finite element meshes: the continuous-time case. *SIAM Journal on Numerical Analysis*, 33:456–474.

Logist, F., Wouwer, A. V., Smets, I., and Impe, J. V. (2007). Optimal temperature profiles for tubular reactors implemented through a flow reversal strategy.

Chemical Engineering Science, 62:4675–4688.

Meerbergen, K., Fresl, K., and Knapen, T. (2009). C++ bindings to external software libraries with examples from BLAS, LAPACK, UMFPACK, and MUMPS. *ACM Transactions on Mathematical Software*, 36:1–23.

Meidner, D. and Vexler, B. (2007). Adaptive space-time finite element methods for parabolic optimization problems. *SIAM Journal*, 46:116–142.

Meidner, D. and Vexler, B. (2008). A priori error estimates for space-time finite element discretization of parabolic optimal control problems part i: Problems without control constraints. *SIAM Journal on Control and Optimization*, 47:1150–1177.

Tröltzsch, F. (2010). *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. American Mathematical Society, Providence, Rhode Island.

Trottenberg, U., Oosterlee, C., and Schüller, A. (2001). *Multigrid*. Academic Press, London.

Vandewalle, S. and Piessens, R. (1993). On dynamic iteration methods for solving time-periodic differential equations. *SIAM Journal on Numerical Analysis*, 30:286–303.