

Direct Multiple Shooting for Parabolic PDE Constrained Optimization

Andreas Potschka

Abstract Direct Multiple Shooting is a flexible and efficient method to solve difficult optimal control problems constrained by ordinary differential equations or differential-algebraic equations. The aim of this article is to concisely summarize the main conceptual and methodological approaches to solve also optimal control problems with parabolic partial differential equations constraints via a Direct Multiple Shooting method. The main obstacle is the sheer size of the discretized optimization problems. We explain a typical direct discretization approach and discuss an inexact SQP method based on two-grid Newton-Picard preconditioning. Special attention is given to a-posteriori κ -estimators that monitor contraction and to the structure-exploiting treatment of the resulting large-scale quadratic programming subproblems, including an extended condensing technique that exploits Multiple Shooting and two-grid Newton-Picard structures. Finally, we present numerical results for an advection-diffusion and a bacterial chemotaxis example.

1 Introduction

The early approaches for optimal control for ordinary differential equations (ODEs) were mostly based on Dynamic Programming [3] or Pontryagin's Maximum Principle [16]. Due to the curse of dimensionality, Dynamic Programming is not applicable to problems with more than a few state variables. The Maximum Principle is an indirect method in the sense that the controls are given as a closed form representation of adjoint states, the so-called co-states, which satisfy an adjoint differential equation and suitable boundary constraints. The resulting boundary value problems in the states and co-states can then be discretized by numerical methods for boundary value problems like Multiple Shooting [15] or Collocation [21]. The Maximum Principle is a typical example for an *optimize-then-discretize* approach. It was found out later that ideas from numerical methods for boundary value problems can also be used on the optimal control problem itself instead of

A. Potschka (✉)

Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany
e-mail: potschka@iwr.uni-heidelberg.de

© Springer International Publishing Switzerland 2015

T. Carraro et al. (eds.), *Multiple Shooting and Time Domain Decomposition Methods*, Contributions in Mathematical and Computational Sciences 9,
DOI 10.1007/978-3-319-23321-5_6

159

its optimality conditions, thus disposing of the need to formulate adjoint equations, which can be a time-consuming affair (*discretize-then-optimize*). Furthermore, this approach lends itself quite naturally to a direct formulation, in which the discretized controls are not eliminated in favor of adjoint variables. For Direct Collocation, we refer the reader to [4, 25] and for Direct Multiple Shooting to [6]. Direct formulations have been found to reflect the real conditioning of the optimal control problem better, in particular when the mapping of the adjoint to the control has a large Lipschitz constant. We want to remark that there are also direct optimize-then-discretize methods for which the control is kept in the system of optimality conditions and its discretization.

In optimal control for partial differential equations (PDEs), indirect optimize-then-discretize methods prevail (see, e.g., [24]). This is mostly due to the fact that indirect methods usually yield more accurate representations of the control (see also [11, Chap. 3]).

The contribution of this article is to summarize a Direct Multiple Shooting method for parabolic PDE constrained problems from Potschka [17, 18] in a concise form and to present numerical results for problems that could also be treated well with indirect methods. The main challenge in this approach is to tame the large-scale nature of the discretized Nonlinear Programming problems (NLPs), which can be mastered by the use of inexact Sequential Quadratic Programming (SQP) with two-grid Newton-Picard preconditioning.

The main strengths of the Direct Multiple Shooting approach can unfold in particular for practitioners who want to solve parabolic PDE constrained optimization problems and need an accurate representation of the system dynamics, but can live with lower resolution of the optimal control. This situation is not unusual due to physical restrictions of manipulator hardware. These practitioners can then benefit from time savings in the problem setup, because no adjoint equations need to be derived, but also in the problem solution, because the method can be easily parallelized on the basis of the Multiple Shooting structures. Furthermore, the solutions delivered by Direct Multiple Shooting can serve as initial guesses for other methods with higher accuracy.

2 Problem Formulation

Let $\Omega \subset \mathbb{R}^d$ be a bounded d -dimensional spatial domain with sufficiently regular boundary $\partial\Omega$. Regarding time, we only consider the fixed interval $(0, 1)$ here for simplicity. We assume that the control satisfies

$$q \in L^2((0, 1); Q), \quad \text{where } Q \subseteq L^2(\Omega)^{n_q^d} \times L^2(\partial\Omega)^{n_q^b},$$

with n_q^d distributed and n_q^b boundary controls. We would already like to point out that the pure boundary control case $n_q^d = 0$ is especially advantageous in a Multiple Shooting approach, as we describe in Sect. 3.

We consider two kinds of coupled state variables: ODE states $v \in C^1([0, 1]; \mathbb{R}^{n_v})$, which are not spatially distributed, and PDE states $u \in W(0, 1)^{n_u}$, for which we use the standard Hilbert space for parabolic PDEs

$$W(0, 1) = \{u \in L^2((0, 1); V) \mid \partial_t u \in L^2((0, 1); V^*)\},$$

where $(V, L^2(\Omega), V^*)$ is a Gelfand triple and $\partial_t u$ denotes the time derivative of u in the sense of vectorial distributions (see, e.g., [29, Chap. IV]). We assume that the spatial regularity of u is $V \subseteq H^1(\Omega)$. According to [7, Chap. XVIII, Theorem 1], we can trade some spatial regularity for higher temporal regularity, because $W(0, 1)$ is continuously embedded in $C^0([0, 1]; L^2(\Omega))$. Thus, we have existence of the trace $u(t) \in L^2(\Omega)^{n_u}$ for all $t \in [0, 1]$, which is important for the formulation of boundary constraints in time.

We assume that the time derivative \dot{v} equals a sufficiently regular nonlinear function $f^{\text{ODE}} : Q \times L^2(\Omega)^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_v}$ and $-\partial_t u$ equals a possibly nonlinear elliptic differential operator $A : Q \times V \times \mathbb{R}^{n_v} \rightarrow V^*$. Furthermore, we allow for generalized temporal boundary constraints, mixed control-ODE-state path constraints and final ODE state constraints via the functions

$$r^b : L^2(\Omega)^{n_u} \times \mathbb{R}^{n_v} \rightarrow L^2(\Omega)^{n_u} \times \mathbb{R}^{n_v}, \quad r^c : Q \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_c}, \quad r^e : \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_e}.$$

The objective function $\Phi : L^2(\Omega)^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ depends on the final values of the ODE and PDE states. Finally, we can state the problem of interest of this article as

$$\begin{aligned} & \underset{\substack{q \in L^2((0,1); Q) \\ u \in W(0,1)^{n_u} \\ v \in C^1([0,1]; \mathbb{R}^{n_v})}}{\text{minimize}} & \Phi(u(1), v(1)) & (1a) \end{aligned}$$

$$\text{s. t.} \quad \partial_t u = -A(q(t), u(t), v(t)), \quad t \in (0, 1), \quad (1b)$$

$$\dot{v} = f^{\text{ODE}}(q(t), u(t), v(t)), \quad t \in (0, 1), \quad (1c)$$

$$(u(0), v(0)) = r^b(u(1), v(1)), \quad (1d)$$

$$r^c(q(t), v(t)) \geq 0, \quad t \in (0, 1), \quad (1e)$$

$$r^e(v(1)) \geq 0. \quad (1f)$$

More general problems with free initial and final time or integral-type objectives can be equivalently reformulated using extra v variables to fit problem class (1). Further assumptions concerning regularity requirements on the occurring functions above is a delicate issue. From a practical point of view, we require that (1) is well-posed and that there exists an appropriate discretization for the involved variables

and functions such that the solution of the resulting finite dimensional optimization problem is consistent with (1).

3 Discretization

In Potschka [17], problem (1) is discretized in space first and then in time. We explain here a different route, where time is discretized before space. In both cases, though, the resulting discretized problem (3) exhibits the same structure.

3.1 Time Discretization

We start out by discretizing the controls in time. To this end, let $0 = t^0 < \dots < t^{n_{\text{MS}}} = 1$ denote a partition of the interval $[0, 1]$. On the shooting intervals $I^i := (t^i, t^{i+1})$, $i = 0, \dots, n_{\text{MS}} - 1$, we perform a piecewise discretization of the controls in time. The piecewise nature is important for decoupling properties to be exploited in the numerics. We restrict ourselves here to the simplest case of a piecewise constant discretization in time such that

$$\tilde{q}(t) = \sum_{j=0}^{n_{\text{MS}}-1} q^j \chi_{I^j}(t), \quad \text{with } q^i \in Q, i = 0, \dots, n_{\text{MS}} - 1,$$

where χ_{I^i} denotes the characteristic function of I^i .

In the next step, we parametrize the state variables by local initial values $u^i, v^i, i = 0, \dots, n_{\text{MS}}$: We assume that the local initial value problems of the form (1b)–(1c) on I^i with initial values u^i, v^i admit a unique solution denoted by

$$\begin{pmatrix} \bar{u}^i(t; q^i, u^i, v^i) \\ \bar{v}^i(t; q^i, u^i, v^i) \end{pmatrix} \in L^2(\Omega)^{n_u} \times \mathbb{R}^{n_v}, \quad \text{for all } t \in I^i, i = 0, \dots, n_{\text{MS}} - 1.$$

Continuity of the states on the full time horizon $(0, 1)$ must then be enforced with matching conditions of the form

$$\bar{u}^i(t; q^i, u^i, v^i) = u^{i+1}, \quad \bar{v}^i(t; q^i, u^i, v^i) = v^{i+1}, \quad i = 0, \dots, n_{\text{MS}} - 1. \quad (2)$$

At this stage of the discretization procedure, we have arrived at a problem depending only on variables that are either real vectors or functions distributed in space only, namely $q^i, u^i, v^i, i = 0, \dots, n_{\text{MS}}$. Note, that we have en passant added an additional control variable $q^{n_{\text{MS}}}$ to lend a common structure to the optimization variables corresponding to each shooting node t_i . We have to eliminate these additional degrees of freedom later via (3f).

3.2 Space Discretization

Now, we can discretize q^i and u^i in space, e.g., by Finite Differences, Finite Elements, or Finite Volumes. We denote the corresponding vectors by q^i and u^i . The quantities \bar{u}^i and \bar{v}^i can then be discretized by an appropriate time stepping method. We denote the result by $\bar{u}^i(t^{i+1}; q^i, u^i, v^i)$ and $\bar{v}^i(t^{i+1}; q^i, u^i, v^i)$. It is not mandatory to use the same grids for the controls or states on all shooting intervals, or even within one shooting interval for the time steps of states in case of a Runge-type time stepping scheme. According to [10], a good compromise for balancing good mesh adaptivity with few Finite Element matrix assembly calls is to fix the state mesh on each shooting interval and discretize the matching conditions (2) in a variational manner. In this case, a Method of Lines discretization in time seems to be a flexible and efficient approach, because existing ODE solvers can be used. For a more detailed discussion of discretization issues of parabolic PDEs, we refer the reader to [23].

For simplicity, we assume here that q^i and u^i are based on the same spatial discretization in each shooting node. However, we assume that there is a hierarchy of nested discretizations $V_h^1 \subset V_h^2 \subset \dots \subset V$ for the states, which in turn yields on each level $\ell = 1, 2, \dots$ discretized states and shooting solutions. The path constraint is discretized on the shooting grid only. For more sophisticated methods, see Potschka [19].

All remaining functions of problem (1) need to be appropriately discretized as well. Finally, we arrive at a highly structured, finite dimensional optimization problem on each spatial discretization level ℓ

$$\underset{(q^i, u^i, v^i)_{i=0}^{n_{\text{MS}}}}{\text{minimize}} \Phi(u^{n_{\text{MS}}}, v^{n_{\text{MS}}}) \quad (3a)$$

$$\text{s. t. } r_u^b(u^{n_{\text{MS}}}, v^{n_{\text{MS}}}) - u^0 = 0, \quad (3b)$$

$$\bar{u}^i(t^i; q^{i-1}, u^{i-1}, v^{i-1}) - u^i = 0, \quad i = 1, \dots, n_{\text{MS}}, \quad (3c)$$

$$r_v^b(u^{n_{\text{MS}}}, v^{n_{\text{MS}}}) - v^0 = 0, \quad (3d)$$

$$\bar{v}^i(t^i; q^{i-1}, u^{i-1}, v^{i-1}) - v^i = 0, \quad i = 1, \dots, n_{\text{MS}}, \quad (3e)$$

$$q^{n_{\text{MS}}} - q^{n_{\text{MS}}-1} = 0, \quad (3f)$$

$$r^i(q^{i-1}, v^{i-1}) \geq 0, \quad i = 1, \dots, n_{\text{MS}}, \quad (3g)$$

$$r^e(v^{n_{\text{MS}}}) \geq 0. \quad (3h)$$

The numerical evaluation of (3c) and (3e) is expensive, because it comprises the solution of an initial value problem for the spatially discretized PDE. To this end, we use an adaptive backward differentiation formula with monitoring strategy [1, 2], which is implemented in the C++ code DAESOL-II. It also supplies efficient first and second order forward and backward directional derivatives of the solutions of the initial value problems on the basis of Internal Numerical Differentiation [5] and

Algorithmic Differentiation [9, 28], for which the backwards mode of Algorithmic Differentiation delivers an automated and efficient way to compute gradients in the style of adjoint equations.

We want to remark that special attention has to be paid to the question of consistency of the discretized problem (3) with the original problem (1) for fine discretizations. For instance, it is necessary to use correct weighting matrices for the discrete approximations of Hilbert space inner products and their corresponding norms. Their inverses must also be used for correct discrete Riesz representations of discrete variables that really live in the dual space, for example the Lagrange multipliers of (3b) and (3c). Then, it is typically possible to prove consistency of (3) and (1) for the particular problem at hand.

4 Newton-Picard Inexact SQP

When disregarding its special structure for a moment, we see that problem (3) is a Nonlinear Programming (NLP) problem of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{4a}$$

$$\text{s. t.} \quad g_i(x) = 0, \quad i \in \mathcal{E}, \tag{4b}$$

$$g_i(x) \geq 0, \quad i \in \mathcal{I}, \tag{4c}$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$. We assume throughout that f and g are twice continuously differentiable functions and that the sets \mathcal{E} and \mathcal{I} form a partition of $\{1, \dots, m\} = \mathcal{E} \dot{\cup} \mathcal{I}$. For $z = (x, y) \in \mathbb{R}^{n+m}$, we can then define the Lagrangian function

$$\mathcal{L}(z) = f(x) - \sum_{i=1}^m y_i g_i(x).$$

For the theory of NLP we refer the reader to [14].

4.1 The Equality Constrained Case

We first consider the equality constrained case $\mathcal{I} = \{\}$. If $x^* \in \mathbb{R}^n$ satisfies a constraint qualification and is a local minimizer of (3), then there exists a Lagrange multiplier $y^* \in \mathbb{R}^m$ such that with $z^* = (x^*, y^*)$

$$F(z^*) = \begin{pmatrix} F_1(z^*) \\ F_2(z^*) \end{pmatrix} = \begin{pmatrix} \nabla_x \mathcal{L}(z^*) \\ g(x^*) \end{pmatrix} = 0. \tag{5}$$

In practice, a Newton-type method is used to solve the necessary optimality condition (5) in order to obtain a critical point z^* . In general we must then check (second order) sufficient conditions to ensure that x^* is really a minimizer and not a maximizer or saddle-point.

Because of the large-scale nature of (3), it is a good idea to employ iterative methods for the solution of the linearized subproblems within the Newton-type method. We recede to a rather simple iterative method here, namely a Linear Iterative Splitting Approach (LISA), because we can generalize it from linear systems to Quadratic Programming (QP) problems in Sect. 4.4.

Let $N = n + m$, $J(z) = \frac{dF}{dz}(z)$, and $\hat{M} : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times N}$ be given such that $\hat{M}(z)J(z) \approx \mathbb{I}_N$. Then, the LISA-Newton iterates are defined via

$$z^{k+1} = z^k + \alpha_k \Delta z^k, \quad z^0 \text{ given}, \quad (6)$$

where $\alpha_k \in (0, 1]$ is an appropriately chosen damping factor to ensure global convergence and Δz^k is computed via the inner LISA iteration

$$\Delta z_{i+1}^k = \Delta z_i^k - \hat{M}(z^k) [J(z^k) \Delta z_i^k + F(z^k)], \quad \Delta z_0^k = 0. \quad (7)$$

The iteration (7) converges for all values of $F(z^k)$ and Δz_0^k if and only if the spectral radius κ_{lin} of $A_k := \mathbb{I}_N - \hat{M}(z^k)J(z^k)$ is less than 1, see [22, Theorem 4.1].

The choice of the damping factors α_k exceeds the scope of this article. The interested reader is referred to the discussion in [8] and its specialization for the LISA-Newton method [17].

4.2 A-Posteriori κ -Estimators

Based on [17, Lemma 4.28], we observe that there is a representation of Δz_l^k for $l \geq 1$ in terms of a truncated Neumann series

$$\Delta z_l^k = - \left[\sum_{i=0}^{l-1} \left(\mathbb{I}_N - \hat{M}(z^k)J(z^k) \right)^i \right] \hat{M}(z^k)F(z^k) =: -M(z^k)F(z^k),$$

where $M(z^k)$ depends on l . On the basis of this M , we can observe the connection

$$\kappa_\varepsilon \in [(\kappa_{\text{lin}})^l, (\kappa_{\text{lin}})^l + \varepsilon] \quad \text{for all } \varepsilon > 0$$

between the linear convergence rate κ_{lin} of the LISA iteration (7) and the linear convergence rate κ_ε (which depends on an ε -dependent norm) of the nonlinear LISA-Newton iteration (6), if each Δz^k is computed from l LISA iterations [17, Theorem 4.29]. This implies that under the assumption $\kappa_{\text{lin}} < 1$, $l = 1$ is already enough for local convergence and, moreover, if we perform $l > 1$ LISA iterations

per outer step, then this extra effort is compensated for by fewer outer LISA-Newton iterations, at least asymptotically for $k \rightarrow \infty$.

Furthermore, this result is the basis for a-posteriori κ -estimators [17]. To explain the different estimators, we fix k and define $\zeta_l := \Delta z_{l+1}^k - \Delta z_l^k$. We immediately observe that $\zeta_{l+1} = A_k \zeta_l$. The Rayleigh κ -estimator is based on the Rayleigh quotient

$$\kappa_l^{\text{Rayleigh}} = \frac{\zeta_l^T A_k \zeta_l}{\zeta_l^T \zeta_l} = \frac{\zeta_l^T \zeta_{l+1}}{\zeta_l^T \zeta_l}.$$

The sequence $(\kappa_l^{\text{Rayleigh}})$ converges to κ_{lin} for $l \rightarrow \infty$ if A^k is diagonalizable, has a single eigenvalue of largest modulus, and ζ_1 has a nonzero component in the corresponding eigenspace. For non-diagonalizable A_k , the Root κ -estimator

$$\kappa_l^{\text{Root}} = (\|\zeta_{l+1}\| / \|\zeta_l\|)^{1/l}$$

can be used. It converges if $\kappa_{\text{lin}} > 0$ and ζ_1 has a nonzero component in the dominant invariant subspace of A_k . However, convergence to κ_{lin} can be quite slow. In contrast, the Ritz κ -estimator converges in a finite number of iterations. It is based on the largest Ritz value κ_l^{Ritz} of A_k on the order- l Krylov subspace generated by A_k and ζ_1 . The disadvantage is that an orthonormal basis of the Krylov subspace needs to be maintained, which can be prohibitive due to excessive memory consumption. In our numerical experience, this is not a problem because the Newton-Picard based preconditioners \hat{M} described in Sect. 4.3 require only few LISA iterations.

The κ -estimators can also be used to assess the error of the LISA iteration (7) in the following sense: For all $\varepsilon \in (0, 1 - \kappa_{\text{lin}})$, there exists a vector-norm $\|\cdot\|_{*,k}$ with corresponding matrix-norm $\|A_k\|_{*,k} \leq \kappa_{\text{lin}} + \varepsilon < 1$ such that (compare [17, Lemma 4.30])

$$\|\Delta z_l^k - \Delta z^k\|_{*,k} \leq \frac{(\kappa_{\text{lin}} + \varepsilon)^l}{1 - (\kappa_{\text{lin}} + \varepsilon)} \|\Delta z_1^k - \Delta z_0^k\|_{*,k}.$$

In our implementation, we have $l \geq 2$ and terminate the LISA iterations as soon as either the κ -estimate is lower than $\kappa_{\text{max}} = \sqrt{1/2}$ or a maximum number of LISA iterations, in our case $l_{\text{max}} = 7$, is reached. In the latter case, we ameliorate the preconditioner \hat{M} in order to reduce the contraction rate κ_{lin} as discussed in the following section. In most outer iterations only $l = 2$ LISA iterations are required. We used the Ritz κ -estimator for the computations in Sect. 6.

4.3 Two-Grid Newton-Picard Preconditioning

The requirement $\kappa_{\text{lin}} < 1$ for convergence of the LISA iteration (7) is a strong requirement, which calls for specially tailored preconditioners. In our case, Newton-

Picard preconditioning is the method of choice for computing \hat{M} , see [20]. Although grid-independent convergence has so far only been proven for a linear quadratic model problem with single shooting, our numerical experience on nonlinear problems with Multiple Shooting is completely satisfying.

The general paradigm of Newton-Picard preconditioners is the following: Under reasonable assumptions, \bar{u}^i has a compact Fréchet derivative with respect to q^i and u^i . Thus, even though the discretized Jacobian matrices $G_q^i = \partial \bar{u}^i / \partial q^i$ and $G_u^i = \partial \bar{u}^i / \partial u^i$ are in general large dense matrices, their eigenvalues and singular values cluster at 0 and they can thus be approximated well by low-rank matrices.

From a geometrical point of view, the compactness is a result of a smoothing property of \bar{u}^i , which can alternatively be exploited to form a low-rank approximation by a two-grid approach. To this end, we approximate the Jacobian matrices G_q^i and G_u^i on coarse grids with suitable projection and restriction matrices P and R . This is the preferred way because no large eigenvalue and singular value problems have to be solved. Furthermore, this approach can be extended in a straight-forward fashion to Multiple Shooting approaches.

4.4 The Inequality Constrained Case

We now consider the inequality constrained case when $\mathcal{I} \neq \{\}$. The Jacobian

$$J(z) = \begin{pmatrix} J_1(z) & -J_2(z)^T \\ J_2(z) & 0 \end{pmatrix} = \begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(z) & -\nabla g(x) \\ \nabla g(x)^T & 0 \end{pmatrix}$$

is a saddle-point matrix in unsymmetric form. It is singular in general, e.g., due to rank deficiency of $J_2(z)$ if lower and upper variable bounds are present. Thus, we need to recede to a pseudo-inverse approach. Let us first consider the case of exact derivatives in the SQP method. In this case, we can define a suitable pseudo-inverse

$$\Delta z^k = J^\oplus(z^k, -F(z^k)) \quad \text{instead of } \Delta z^k = -M(z^k)F(z^k),$$

by computing the primal-dual solution $\tilde{z} = (\tilde{x}, \tilde{y}) \in \mathbb{R}^{n+m}$ of the QP

$$\underset{\tilde{x} \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \tilde{x}^T J_1(z^k) \tilde{x} + (F_1(z^k) - J_1(z^k)x^k + J_2(z^k)^T y^k)^T \tilde{x} \tag{8a}$$

$$\text{s. t.} \quad (J_2(z^k)\tilde{x} + (F_2(z^k) - J_2(z^k)x^k))_i = 0, \quad i \in \mathcal{E}, \tag{8b}$$

$$(J_2(z^k)\tilde{x} + (F_2(z^k) - J_2(z^k)x^k))_i \geq 0, \quad i \in \mathcal{I}, \tag{8c}$$

and then the step $\Delta z^k = \tilde{z} - z^k$. One can show that if QP (8) has a unique solution at z^k and if some $\hat{z} = (\hat{x}, \hat{y}) \in \mathbb{R}^N$ satisfies $\hat{y}_i \geq -y_i$ for all $i \in \mathcal{I}$, then J^\oplus acts linearly

on the second argument like a pseudo-inverse in the sense that $J^\oplus(z^k, J(z^k)\hat{z}) = \hat{z}$ [17, Lemma 4.36].

Furthermore, if the active set of QP (8) does not change from iteration $k - 1$ to k and the second order sufficient condition and strong complementarity hold, then we can construct a matrix M^k such that for a neighborhood U of $F(z^k)$ it holds that

$$-M^k \hat{F} = J^\oplus(z^k, -\hat{F}) \quad \text{for all } \hat{F} \in U.$$

In words, the pseudo-inverse J^\oplus acts locally like a matrix [17, Theorem 4.37]. In addition, if the SQP method with J^\oplus converges to z^* , then z^* is a critical point of NLP (4). Moreover, the second order sufficiency condition and strong complementarity transfer from QP (8) at z^* to NLP (4) at z^* [17, Theorem 4.41].

In the case of inexact derivatives on the basis of the Newton-Picard approximation from Sect. 4.3 we use a structured approximation of the Jacobian

$$J(z^k) = \begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(z^k) & -\nabla g(x^k) \\ \nabla g(x^k)^T & 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{B}^k & -(\tilde{C}^k)^T \\ \tilde{C}^k & 0 \end{pmatrix} =: \hat{J}^k.$$

In analogy to the construction of $J^\oplus(z^k, \cdot)$ from $J(z^k)$, we can construct a QP preconditioner $\hat{M}(z^k)$ for a QP-LISA iteration. We first compute the primal-dual solution $\tilde{z} = (\tilde{x}, \tilde{y})$ of the QP

$$\underset{\tilde{x} \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \tilde{x}^T \tilde{B}^k \tilde{x} + (F_1(x^k) - \tilde{C}^k(x^k + \Delta x_{l-1}^k) + (\tilde{C}^k)^T(y^k + \Delta y_{l-1}^k))^T \tilde{x} \quad (9a)$$

$$\text{s. t.} \quad (\tilde{C}^k \tilde{x} + (F_2(x^k) - \tilde{C}^k(x^k + \Delta x_{l-1}^k)))_i = 0, \quad i \in \mathcal{E}, \quad (9b)$$

$$(\tilde{C}^k \tilde{x} + (F_2(x^k) - \tilde{C}^k(x^k + \Delta x_{l-1}^k)))_i \geq 0, \quad i \in \mathcal{I}, \quad (9c)$$

and then $\Delta z_l^k = \tilde{z} - z^k - \Delta z_{l-1}^k$.

In order to have valid κ -estimators, the working set of QP (9) should be fixed for $l > 1$, effectively deferring further changes of the active set to the next outer iteration.

5 Numerical Solution of the Large-Scale QPs

In order to solve the large-scale QP (9) efficiently, we need to exploit the specific block-sparse and low-rank structures generated in the Multiple Shooting and Newton-Picard approaches. To this end, we first regroup the n_1 PDE variables and n_2 non-PDE variables of NLP (3) in the order

$$(\mathbf{u}^0, \dots, \mathbf{u}^{n_{MS}} \mid \mathbf{v}^0, \dots, \mathbf{v}^{n_{MS}}, \mathbf{q}^0, \dots, \mathbf{q}^{n_{MS}}) = (x_1 \mid x_2) \in \mathbb{R}^{n_1 + n_2}.$$

Then we observe that we can also divide the constraints of NLP (3) to obtain the structured NLP form

$$\underset{(x_1, x_2) \in \mathbb{R}^{n_1+n_2}}{\text{minimize}} \quad f(x_1, x_2) \tag{10a}$$

$$\text{s. t.} \quad g_i(x_1, x_2) = 0, \quad i \in \mathcal{E}_1, \tag{10b}$$

$$g_i(x_1, x_2) = 0, \quad i \in \mathcal{E}_2, \tag{10c}$$

$$g_i(x_1, x_2) \geq 0, \quad i \in \mathcal{I}, \tag{10d}$$

where $|\mathcal{E}_1| = n_1$. The constraints (10b) comprise the PDE boundary constraints (3b) and PDE matching conditions (3c).

5.1 Multiple Shooting Structure Exploitation

The QP constraint matrix C has the form

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \\ C_{31} & C_{32} \end{pmatrix} = \left(\begin{array}{ccc|ccc} -\mathbb{I}_{n_u} & & R_{uu}^b & & & \\ G_u^1 & -\mathbb{I}_{n_u} & & G_v^1 & & R_{uv}^b & G_q^1 \\ & \ddots & & \ddots & & & \ddots \\ & & G_u^{n_{MS}} & -\mathbb{I}_{n_u} & & & G_q^{n_{MS}} \\ \hline H_u^1 & & R_{vu}^b & -\mathbb{I}_{n_v} & & R_{vv}^b & H_q^1 \\ & \ddots & & H_v^1 & -\mathbb{I}_{n_v} & & \ddots \\ & & H_u^{n_{MS}} & & H_v^{n_{MS}} & -\mathbb{I}_{n_v} & H_q^{n_{MS}} \\ \hline & & & & & \mathbb{I}_{n_q} & -\mathbb{I}_{n_q} \\ R_v^{i,1} & & & & & R_q^{i,1} & \\ & & & \ddots & & & \ddots \\ & & & & R_v^{i,n_{MS}} & & R_q^{i,n_{MS}} \\ & & & & & R_v^{e,n_{MS}} & \end{array} \right),$$

with the derivative abbreviations

$$\begin{aligned}
 R_{uu}^b &= \frac{\partial \mathbf{r}_u^b}{\partial \mathbf{u}^{n_{MS}}}, & R_{uv}^b &= \frac{\partial \mathbf{r}_u^b}{\partial \mathbf{v}^{n_{MS}}}, & R_{vu}^b &= \frac{\partial \mathbf{r}_v^b}{\partial \mathbf{u}^{n_{MS}}}, & R_{vv}^b &= \frac{\partial \mathbf{r}_v^b}{\partial \mathbf{v}^{n_{MS}}}, & R^e &= \frac{r^e}{\partial \mathbf{v}^{n_{MS}}}, \\
 G_q^i &= \frac{\partial \bar{\mathbf{u}}^i}{\partial \mathbf{q}^{i-1}}, & G_u^i &= \frac{\partial \bar{\mathbf{u}}^i}{\partial \mathbf{u}^{i-1}}, & G_v^i &= \frac{\partial \bar{\mathbf{u}}^i}{\partial \mathbf{v}^{i-1}}, & R_q^{i,i} &= \frac{\partial \mathbf{r}^i}{\partial \mathbf{q}^{i-1}}, \\
 H_q^i &= \frac{\partial \bar{\mathbf{v}}^i}{\partial \mathbf{q}^{i-1}}, & H_u^i &= \frac{\partial \bar{\mathbf{v}}^i}{\partial \mathbf{u}^{i-1}}, & H_v^i &= \frac{\partial \bar{\mathbf{v}}^i}{\partial \mathbf{v}^{i-1}}, & R_v^{i,i} &= \frac{\partial \mathbf{r}^i}{\partial \mathbf{v}^{i-1}},
 \end{aligned}$$

at the current iterate z^k . We want to stress that contrary to the appearance the block C_{11} is several orders of magnitude larger than the blocks C_{22} and C_{32} on fine spatial discretization levels.

Let $M_B = \mathbb{I}_{n_u} - (\prod_{i=1}^{n_{MS}} G_u^i) R_{uu}^b$. It is easy to see that if M_B is invertible, then C_{11} is also invertible and its inverse is given by $C_{11}^{-1} =$

$$\begin{pmatrix} -\mathbb{I} & & & -(\prod_{i=1}^0 G_u^i) R_{uu}^b \\ & \ddots & & \vdots \\ & & -\mathbb{I} & -(\prod_{i=1}^{n_{MS}-1} G_u^i) R_{uu}^b \\ & & & -\mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & & & \\ & \ddots & & \\ & & \mathbb{I} & \\ & & & M_B^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & & & \\ \prod_{i=1}^1 G_u^i & \mathbb{I} & & \\ \vdots & \ddots & \ddots & \\ \prod_{i=1}^{n_{MS}} G_u^i & \dots & \prod_{i=n_{MS}}^{n_{MS}} G_u^i & \mathbb{I} \end{pmatrix}.$$

This decomposition into block-triangular matrices yields a procedural recipe for evaluating matrix-vector products with C_{11}^{-1} that does not require to explicitly form either C_{11} or any part of C_{11}^{-1} , except for the M_B^{-1} block. In the following, we shall see that a two-grid Newton-Picard approximation of C_{11} also enables us to use a simple procedural form for matrix-vector products with the approximation of M_B^{-1} .

5.2 Two-Grid Newton-Picard Structure Exploitation

To this end, we need to establish grid transfer operators between a coarse and a fine grid. For two grid levels $\ell_c \leq \ell_f$, we denote the PDE-state degrees of freedom by $n_c = n_u^{\ell_c}$ and $n_f = n_u^{\ell_f}$. Typically, $n_c \approx 100$, while $n_f > 10,000$. We then assume that the prolongation operator $P \in \mathbb{R}^{n_f \times n_c}$ and the restriction operator $R \in \mathbb{R}^{n_c \times n_f}$ satisfy

$$RP = \mathbb{I}_{n_c}. \quad (11)$$

On nested grids, it is rather simple to construct P via interpolation. The restriction of $v_f \in \mathbb{R}^{n_f}$ can then be computed as the unique coarse grid vector $v_c \in \mathbb{R}^{n_c}$ whose prolongation minimizes the L_2 -distance to v_f on the fine grid. In case of a variational spatial discretization, we can use the coarse and fine grid mass matrices $M_c \in \mathbb{R}^{n_c \times n_c}$ and $M_f \in \mathbb{R}^{n_f \times n_f}$ in order to obtain the form

$$R = M_c^{-1} P^T M_f,$$

which satisfies (11) due to $P^T M_f P = M_c$. Again, the matrix R should not be formed explicitly if M_c^{-1} is a dense matrix.

We then approximate the blocks of C via a two-grid approach. To this end, we denote all matrices on the coarse grid with hats ($\hat{\cdot}$) and approximate C by a matrix \tilde{C} of the same block pattern, but consisting of the approximated blocks

$$\tilde{R}_{uu}^b = P\hat{R}_{uu}^b R, \quad \tilde{R}_{uv}^b = P\hat{R}_{uv}^b, \quad \tilde{R}_{vu}^b = \hat{R}_{vu}^b R, \quad \tilde{R}_{vv}^b = \hat{R}_{vv}^b, \quad (12a)$$

$$\tilde{G}_q^i = P\hat{G}_q^i, \quad \tilde{G}_u^i = P\hat{G}_u^i R, \quad \tilde{G}_v^i = P\hat{G}_v^i, \quad (12b)$$

$$\tilde{H}_q^i = \hat{H}_q^i, \quad \tilde{H}_u^i = \hat{H}_u^i R, \quad \tilde{H}_v^i = \hat{H}_v^i, \quad (12c)$$

$$\tilde{R}_q^{i,i} = \hat{R}_q^{i,i}, \quad \tilde{R}_v^{i,i} = \hat{R}_v^{i,i}, \quad \tilde{R}^e = \hat{R}^e. \quad (12d)$$

Along the same lines, we use the approximations

$$\begin{aligned} \hat{G}_B &:= \left(\prod_{i=1}^{n_{MS}} \hat{G}_u^i \right) \hat{R}_{uu}^b, & \hat{M}_B &:= \mathbb{I}_{n_c} - \hat{G}_B, \\ \tilde{G}_B &:= \left(\prod_{i=1}^{n_{MS}} \tilde{G}_u^i \right) \tilde{R}_{uu}^b, & \tilde{M}_B &:= \mathbb{I}_{n_f} - \tilde{G}_B. \end{aligned}$$

We can then observe that if \hat{M}_B is invertible, then \tilde{M}_B is also invertible and we obtain the procedural recipe [17, Lemma 7.2]

$$\tilde{M}_B^{-1} = \left(\mathbb{I}_{n_f} - P\hat{G}_B R \right)^{-1} = \mathbb{I}_{n_f} - P \left(\mathbb{I}_{n_c} - \hat{M}_B^{-1} \right) R.$$

Thus, we do not need to explicitly form any matrix but \hat{M}_B on the coarse grid in order to evaluate matrix-vector products with \tilde{M}_B and thus also with \tilde{C}_{11}^{-1} . Furthermore, this form of \tilde{M}_B allows for an even more concise representation of \tilde{C}_{11}^{-1} . To this end, we define the projectors

$$\Pi^{\text{slow}} = \mathbb{I}_{n_{MS}} \otimes (PR), \quad \Pi^{\text{fast}} = \mathbb{I}_{n_{MS}n_f} - \Pi^{\text{slow}},$$

where \otimes denotes the Kronecker product of matrices. Thus Π^{slow} is a block-diagonal matrix of n_{MS} blocks PR . Then, we can show [17, Theorem 7.3] that

$$\tilde{C}_{11}^{-1} \Pi^{\text{slow}} = (\mathbb{I}_{n_{MS}} \otimes P) \hat{C}_{11}^{-1} (\mathbb{I}_{n_{MS}} \otimes R), \quad \tilde{C}_{11}^{-1} \Pi^{\text{fast}} = -\Pi^{\text{fast}}.$$

This equation illustrates the name Newton-Picard: On the slowly converging modes, we approximate the inverse of C_{11} by its coarse grid counterpart and perform a Newton iteration on this subspace, while we only use a Picard (or fixed point) iteration on the anyway fast converging modes through approximating the Jacobian with a negative identity. Moreover, we can now easily derive [17, Corollary 7.4] that

if it exists, the Newton-Picard approximation of block C_{11} has the inverse

$$\tilde{C}_{11}^{-1} = (\mathbb{I}_{n_{MS}} \otimes P) \left(\hat{C}_{11}^{-1} + \mathbb{I}_{n_{MS}n_c} \right) (\mathbb{I}_{n_{MS}} \otimes R) - \mathbb{I}_{n_{MS}n_f}.$$

5.3 QP Condensing

We now consider again QPs with a structure inherited from NLP (10)

$$\underset{(x_1, x_2) \in \mathbb{R}^{n_1+n_2}}{\text{minimize}} \quad \frac{1}{2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (13a)$$

$$\text{s. t.} \quad C_{11}x_1 + C_{12}x_2 = c_1, \quad (13b)$$

$$C_{21}x_1 + C_{22}x_2 = c_2, \quad (13c)$$

$$C_{31}x_1 + C_{32}x_2 \geq c_3. \quad (13d)$$

We show that we can employ a partial null-space approach called *condensing* in order to efficiently solve the large-scale, structured QP (13b). Condensing is the key linear algebra ingredient for an efficient Direct Multiple Shooting method (see, e.g., [6]), but must be extended to also exploit the two-grid Newton-Picard structures. For this purpose, we prefer the following presentation of the concept of condensing.

Theorem 1 (Condensing) *Assume that C_{11} in QP (13) is invertible and define*

$$\begin{aligned} Z &= \begin{pmatrix} -C_{11}^{-1}C_{12} \\ \mathbb{I}_{n_2} \end{pmatrix}, & B' &= Z^T B Z, \\ c'_1 &= C_{11}^{-1}c_1, & b' &= B_{21}c'_1 + b_2 - C_{12}^T C_{11}^{-T} (B_{11}c'_1 + b_1), \\ c'_2 &= c_2 - C_{21}c'_1, & C'_2 &= C_{22} - C_{21}C_{11}^{-1}C_{12}, \\ c'_3 &= c_3 - C_{31}c'_1, & C'_3 &= C_{32} - C_{31}C_{11}^{-1}C_{12}. \end{aligned}$$

Let furthermore $(x_2^*, y_{\mathcal{E}_2}^*, y_{\mathcal{I}}^*) \in \mathbb{R}^{n_2+m_2+m_3}$ be a primal-dual solution of the QP

$$\underset{x_2 \in \mathbb{R}^{n_2}}{\text{minimize}} \quad \frac{1}{2} x_2^T B' x_2 + b'^T x_2 \quad \text{s. t.} \quad C'_2 x_2 = c'_2, \quad C'_3 x_2 \geq c'_3, \quad (14)$$

If we choose

$$x_1^* = C_{11}^{-1}(c_1 - C_{12}x_2^*), \quad (15a)$$

$$y_{\mathcal{E}_1}^* = C_{11}^{-T} \left((B_{12} - B_{11}C_{11}^{-1}C_{12})x_2^* + B_{11}c'_1 + b_1 - C_{21}^T y_{\mathcal{E}_2}^* - C_{31}^T y_{\mathcal{I}}^* \right) \quad (15b)$$

then $(x^*, y^*) := (x_1^*, x_2^*, y_{\mathcal{E}_1}^*, y_{\mathcal{E}_2}^*, y_{\mathcal{I}}^*)$ is a primal-dual solution of QP (13).

Proof See, e.g., [17, Theorem 7.6].

Theorem 1 shows how the solution of the structured, large-scale QP (13) can be boiled down to a medium-scale QP (14), which only contains non-PDE variables x_2 . The pre- and post-processing steps only require matrix-vector products with the large-scale blocks, e.g., n_2 matrix-vector products for the computation of $C_{11}^{-1}C_{12}$.

5.4 Two-Grid Hessian Approximation

If we now use the approximated version of QP (13), we see that based on Sects. 5.1 and 5.2, the partial null-space basis can be evaluated purely on the coarse grid due to

$$\tilde{Z} = \begin{pmatrix} -\tilde{C}_{11}^{-1}\tilde{C}_{12} \\ \mathbb{I} \end{pmatrix} = \begin{pmatrix} -(\mathbb{I} \otimes P)\hat{C}_{11}^{-1}(\mathbb{I} \otimes R)(\mathbb{I} \otimes P)\hat{C}_{12} \\ \mathbb{I} \end{pmatrix} = \begin{pmatrix} -(\mathbb{I} \otimes P)\hat{C}_{11}^{-1}\hat{C}_{12} \\ \mathbb{I} \end{pmatrix}.$$

This observation suggests a projected Newton-Picard approximation of the Hessian matrix via

$$\begin{aligned} \tilde{B}^{\text{fast}} &= \begin{pmatrix} (\Pi^{\text{fast}})^T B_{11} \Pi^{\text{fast}} & 0 \\ 0 & 0 \end{pmatrix}, \\ \tilde{B}^{\text{slow}} &= \begin{pmatrix} (\mathbb{I}_{n_{\text{MS}}} \otimes R)^T \hat{B}_{11} (\mathbb{I}_{n_{\text{MS}}} \otimes R) & (\mathbb{I}_{n_{\text{MS}}} \otimes R)^T \hat{B}_{12} \\ \hat{B}_{21} (\mathbb{I}_{n_{\text{MS}}} \otimes R) & \hat{B}_{22} \end{pmatrix}, \\ \tilde{B} &= \tilde{B}^{\text{fast}} + \tilde{B}^{\text{slow}}. \end{aligned}$$

Consequently, we have

$$\tilde{Z}^T \tilde{B}^{\text{fast}} \tilde{Z} = 0$$

and thus we can also compute the condensed two-grid Newton-Picard Hessian matrix purely on the coarse grid according to

$$\tilde{B}' = \tilde{Z}^T \tilde{B} \tilde{Z} = \hat{Z}^T \hat{B} \hat{Z} \quad \text{with } \hat{Z} = \begin{pmatrix} -\hat{C}_{11}^{-1}\hat{C}_{12} \\ \mathbb{I}_{n_c} \end{pmatrix}.$$

Thus, it is possible to set up QP (14) efficiently with only a few fine-grid operations.

6 Numerical Examples

We now report on the numerical results for Direct Multiple Shooting applied to tracking problems for an advection-diffusion equation in 2D and a bacterial chemotaxis example in 1D. Further examples can be found in [17], including a real-world separation process from chemical engineering with economic objective and constraint functions. All computational results were computed on a Linux workstation with four 2.67 GHz Intel Core i7 cores and 24 GB of RAM.

6.1 Advection-Diffusion Equation

We consider an advection-diffusion equation on $\Omega \subset (-1, 1)^2$, whose boundary $\partial\Omega$ is partitioned into disjoint sets $\Gamma_i, i = 0, 1, 2, 3$. Let ν denote the outwards pointing normal on $\partial\Omega$ and let $\mathbf{U} \in L^2(\Omega)^2$ be a divergence-free velocity field. For $\gamma, \varepsilon, T > 0$, inflow profiles $u_i^{\text{in}} \in L^2(\Gamma_i), i = 1, 2$, and a desired profile $\hat{u} \in L^2(\Omega)$, we apply the proposed method to the following periodic tracking-type boundary-control problem with control gradient constraints

$$\underset{\substack{u \in W(0,T) \\ q \in H^1(0,T)^2}}{\text{minimize}} \quad \frac{1}{2} \|u(T, \cdot) - \hat{u}\|_{L^2(\Omega)}^2 + \frac{\gamma}{2} (\|q_1\|_{H^1(0,T)}^2 + \|q_2\|_{H^1(0,T)}^2) \quad (16a)$$

$$\text{s. t.} \quad \partial_t u = \varepsilon \nabla \cdot \nabla u - \mathbf{U} \cdot \nabla u, \quad \text{in } \Omega, \quad (16b)$$

$$u(0, \cdot) = u(T, \cdot), \quad \text{in } \Omega, \quad (16c)$$

$$\nu \cdot (\varepsilon \nabla u - \mathbf{U}u) = 0, \quad \text{on } \Gamma_0, \quad (16d)$$

$$\nu \cdot (\varepsilon \nabla u - \mathbf{U}u) = u_i^{\text{in}} q_i - u, \quad \text{on } \Gamma_i, i = 1, 2, \quad (16e)$$

$$\nu \cdot (\varepsilon \nabla u - \mathbf{U}u) = -u, \quad \text{on } \Gamma_3, \quad (16f)$$

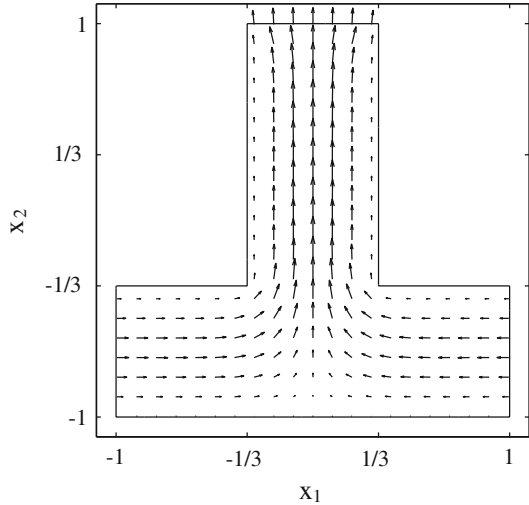
$$q_i(0) = q_i(T), \quad \text{for } i = 1, 2, \quad (16g)$$

$$q_i(t) \geq 0, \quad \text{for } t \in [0, T], i = 1, 2, \quad (16h)$$

$$\frac{dq_i}{dt}(t) \in [-20, 20], \quad \text{for a.a. } t \in [0, T], i = 1, 2. \quad (16i)$$

The setting $q_i \in H^1(0, T)$ can be reformulated with $L^2(0, T)$ controls by introduction of ODE states v_i , whose time-derivative we control, and adequate adaption of all occurrences of q_i in (16).

Fig. 1 Divergence-free velocity field obtained from the solution of a Stokes problem on an inverted-T-shaped domain with inflow from the left on Γ_1 , inflow from the right on Γ_2 and outflow on top on Γ_3



We obtain a suitable velocity field U as the solution of a Stokes problem with parabolic inflow velocities on $\Gamma_1 = \{-1\} \times (-1, -1/3)$ and $\Gamma_2 = \{1\} \times (-1, -1/3)$ and free outflow velocity on $\Gamma_3 = (-1/3, 1/3) \times \{1\}$ (compare Fig. 1)

$$\begin{aligned}
 -\nabla \cdot \nabla U + \nabla p &= 0, && \text{in } \Omega, \\
 \nabla \cdot U &= 0, && \text{in } \Omega, \\
 U(x) &= 0, && \text{for } x \in \Gamma_0, \\
 U(x) &= (9(x_2 + 1/3)(x_2 + 1), 0), && \text{for } x \in \Gamma_1, \\
 U(x) &= (-9(x_2 + 1/3)(x_2 + 1), 0), && \text{for } x \in \Gamma_2.
 \end{aligned}$$

We use Finite Differences on equidistant, staggered grids for the discretization of the Stokes problem. This discretization is compatible with an upwind-flux Finite Volume method for the advection-diffusion equation, because the velocity degrees of freedom align with the center of the volume interfaces.

We compute optimal controls for $T = 10$, $\hat{u}(x) = 1 + x_1$, $u_i^{\text{in}}(x) = \exp(-10(x_2 + 2/3)^2)$, $i = 1, 2$, and for varying values of $\varepsilon = 10^0, 10^{-1}, 10^{-2}, 5 \cdot 10^{-3}$ and $\gamma = 10^{-2}, 10^{-3}$ on $n_{\text{MS}} = 20$ shooting intervals. We use a 5-level hierarchy of grids with mesh sizes $h = 2/15, 1/15, 1/30, 1/60, 1/120$, giving rise to 125, 500, 2000, 8000, and 32,000 degrees of freedom on the respective level. As initial values, we chose $u^i = 0$, $v^i = (0.1, 0.1)^T$, $q^i = (0.01, 0.01)^T$ for $i = 0, \dots, 20$.

One purpose of treating an advection-diffusion problem here is to fathom the limits of the Newton-Picard approach, which exploits dispersion numerically. We can expect the problem to become more difficult to solve as we approach the hyperbolic limit $\varepsilon \rightarrow 0$. Because the upwind discretization achieves numerical stability by introduction of spurious numerical dispersion, the discretized problem

Table 1 The number of inexact SQP iterations and the computation times depend on the diffusion coefficient ε and the regularization parameter γ

$\varepsilon =$	$\gamma = 10^{-1}$				$\gamma = 10^{-3}$			
	10^0	10^{-1}	10^{-2}	$5 \cdot 10^{-3}$	10^0	10^{-1}	10^{-2}	$5 \cdot 10^{-3}$
Overall iterations	12	15	19	16	12	21	17	28
Iterations $l_f = 5$	5	7	7	7	5	6	6	8
Final coarse level	1	1	2	2	1	2	2	2
Time steps	657	760	884	903	655	741	859	935
Sim. time (min:s)	1:48	2:09	2:10	2:13	1:48	1:54	2:17	2:12
Time (h:min:s)	0:49:04	1:21:53	2:34:15	3:30:21	0:48:58	1:23:36	2:49:57	3:22:31

The overall computation time is given as the wall-clock time on four processors, while the simulation time is the CPU time on a single processor

exhibits more diffusion on coarser grid levels where the effect of numerical dispersion is more pronounced. Thus, the efficiency of the diffusion-exploiting Newton-Picard preconditioners is slightly better for coarser fine grids. We also want to remark that problem (16) is a linear-quadratic problem. We tackle it with a nonlinear solver and thus the reader should keep in mind that numerical linear algebra approaches (e.g., along the lines of Potschka et al. [20]) are more efficient, because they exploit linearity explicitly.

From the computational results in Table 1, we can observe the expected trend in the number of overall inexact SQP iterations that is growing for decreasing values of the diffusion coefficient ε . The increase is higher for lower regularization parameters γ . The number of inexact SQP iterations with the fine grid on the finest level grows only slightly. For high diffusion $\varepsilon = 10^0$, level $\ell_c = 1$ for the coarse grid is already enough for sufficient local contraction on all fine grid levels $\ell_f \leq 5$. For $\varepsilon = 10^{-1}$, $\ell_c = 1$ is still sufficiently fine for a high regularization parameter $\gamma = 10^{-1}$, but not for $\gamma = 10^{-3}$. For $\varepsilon = 10^{-2}, 5 \cdot 10^{-3}$, the κ -estimators trigger a refinement of the coarse level to $\ell_c = 2$ in order to obtain local contraction of the method on all fine grid levels $\ell_f \leq 5$. For $\varepsilon = 10^{-3}$, the coarse grid needs to be refined to $\ell_c = 3$. In this case, the memory consumption of DAESOL-II on the coarse and fine grids exceeds the available RAM space. This problem, however, is a momentary technical restriction of the software implementation and not a fundamental restriction of the method.

We can also observe in Table 1 that the number of adaptive time-steps for the initial value problem solver increases with decreasing diffusion coefficient γ . From the number of required time-steps we see that the time discretization has a higher resolution than the spatial discretization. The reason for this imbalance in accuracy are stability requirements for the solution of the initial value problems in (3c) and its first and second order derivatives as discussed in [17, Chap. 9, Sect. 3]. If we take into account the intermediate time steps, we obtain on the finest level between $2.1 \cdot 10^7$ and $3.0 \cdot 10^7$ state variables, although NLP (3) really only has $6.7 \cdot 10^5$ degrees of freedom in the PDE state variables $\mathbf{u}^i, i = 0, \dots, n_{MS}$, (independently of ε).

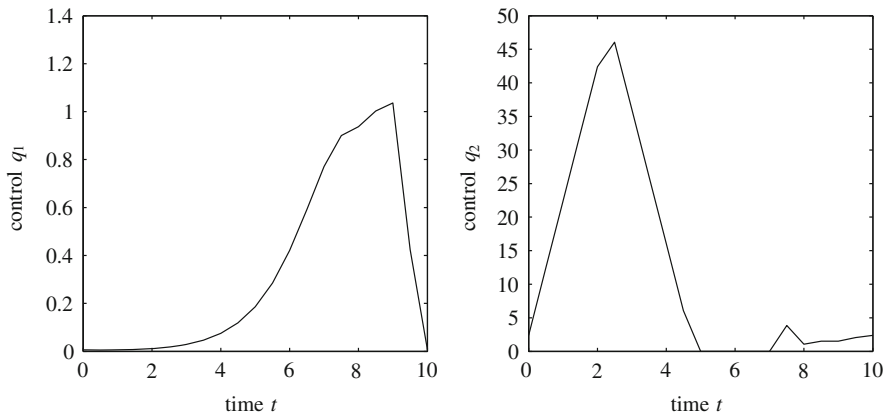


Fig. 2 The optimal controls for problem (16) with $\varepsilon = 10^{-2}, \gamma = 10^{-3}$ work with different magnitudes. The control gradient constraint is active for q_2 at the slopes of the initial peak

If we compare the serial simulation time for the numerical solution of the initial value problems for fixed initial values and fixed controls with the wall-clock time for the inexact SQP method parallelized on four cores, we obtain a ratio of 27–95. This is quite remarkable, because already for the forward problem of finding a time-periodic solution for fixed controls, several Newton-Picard iterations (in the sense of Lust et al. [13]) would be required.

For completeness, we depict the optimal control and the corresponding optimal final state for the case $\varepsilon = 10^{-2}$ and $\gamma = 10^{-3}$ in Figs. 2 and 3. We can observe that most of the control action happens at Γ_2 , where high values of u are to be tracked. The control gradient constraint (16i) is active at the slopes of the initial peak of q_2 . Furthermore, the optimal state at the end of the period matches the desired profile \hat{u} well except for the area around the inflow boundary Γ_2 .

6.2 Bacterial Chemotaxis

On the basis of a chemotaxis model by Tyson et al. [26, 27], we consider on $\Omega = (0, 1)$ with $\partial\Omega = \Gamma_1 \cup \Gamma_2 = \{0\} \cup \{1\}$ the nonlinear tracking-type boundary control problem (compare also [12])

$$\underset{\substack{z, c \in W(0,1) \\ q_1, q_2 \in L^2(0,1)}}{\text{minimize}} \quad \frac{1}{2} \int_{\Omega} (z(1, \cdot) - \hat{z})^2 + \frac{\gamma_c}{2} \int_{\Omega} (c(1, \cdot) - \hat{c})^2 + \frac{\gamma_q}{2} \int_0^1 (q_1^2 + q_2^2) \quad (17a)$$

$$\text{s. t.} \quad \partial_t z = \nabla \cdot \left(D_z \nabla z - \alpha \frac{z}{(1+c)^2} \nabla c \right), \quad \text{in } (0, 1) \times \Omega, \quad (17b)$$

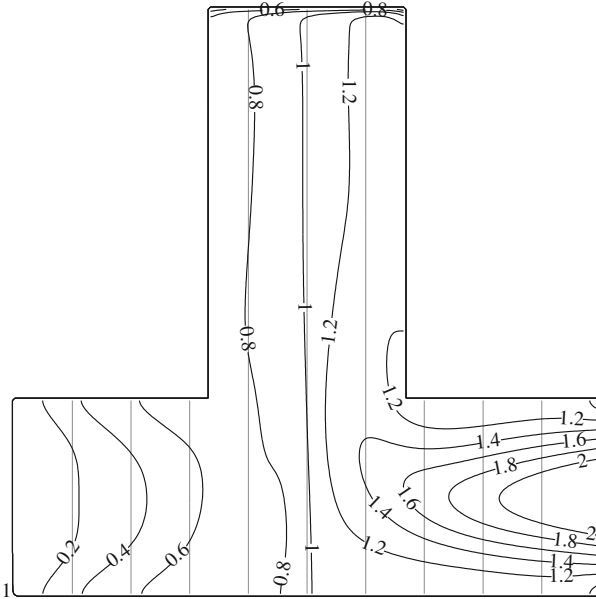


Fig. 3 For $\varepsilon = 10^{-2}$, $\gamma = 10^{-3}$, the *black isolines* of the optimal final state $u(T, \cdot)$ match the *gray isolines* of the target well except for the south-east part of the domain. A boundary layer is visible at the northern outflow Γ_3

$$\partial_t c = \nabla \cdot \nabla c + w \frac{z^2}{(\mu + z^2)} - \rho c \quad \text{in } (0, 1) \times \Omega, \quad (17c)$$

$$\partial_\nu z = \frac{\alpha \beta (q_i - c) z}{D_z (1 + c)^2}, \quad \partial_\nu c = \beta (q_i - c), \quad \text{in } (0, 1) \times \Gamma_i, i = 1, 2, \quad (17d)$$

$$z(0, \cdot) = z_0, \quad c(0, \cdot) = c_0, \quad (17e)$$

$$q_i(t) \in [0, 1], \quad \text{for a.a. } t \in (0, 1), \quad (17f)$$

where ∂_ν denotes the derivative in direction of the outwards pointing normal on Ω . The parameters considered here are $D_z = 0.3$, $\alpha = 4$, $\beta = 1$, $w = 0.1$, $\mu = 10$, $\rho = 0.1$, $\gamma_c = 10^{-2}$, $\gamma_q = 10^{-5}$. The targets to be tracked are $\hat{z}(x) = 2x$, $\hat{c}(x) = 0$ and the initial states are given as $z_0(x) = 1$, $c_0(x) = 0$.

On the basis of a Finite Difference discretization on a six-level hierarchy of nested grids with 322, 642, 1282, 2562, 5122, 10,242 degrees of freedom for the PDE states, we can use the proposed Direct Multiple Shooting method to obtain the optimal controls and states depicted in Fig. 4. Level $\ell_c = 1$ for the coarse level is sufficient to yield good local contraction for the inexact SQP method. In the solution, DAESOL-II uses 889 integration adaptive steps for the solution of the initial value

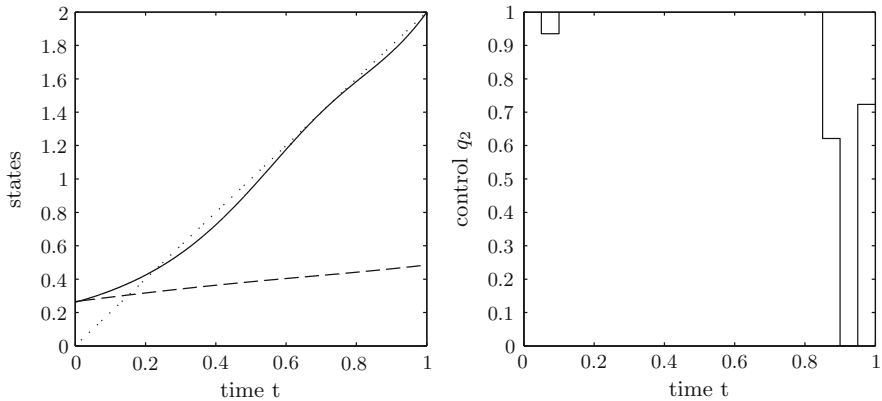


Fig. 4 *Left:* the desired cell distribution \hat{z} (dotted line) can be reached well with the optimal final cell distribution $z(1, \cdot)$ (solid line). The final chemoattractor concentration $c(1, \cdot)$ is almost affine-linear (dashed line). *Right:* the control bound constraints are active most of the time for q_2 . Not shown: the optimal control q_1 is always 0

Table 2 Cumulative CPU time in seconds for the chemotaxis example on each spatial discretization level for computations concerning the dynamic system, including system integration, forward and adjoint derivatives, and matrix-vector products with Hessian matrices

Level	Integration	Forward	Adjoint	Hessian
1	100.5	745.5	19.8	669.1
2	29.5	6.6	16.2	55.8
3	45.4	8.2	22.0	71.7
4	53.4	7.5	21.2	70.5
5	144.2	15.2	42.6	141.4
6	1314.9	73.5	190.3	597.9

problems on all 20 shooting intervals together. In this case, the spatial and temporal discretization errors are well balanced, because a high-order method is used in time.

In Table 2 we sum up the cumulative CPU time needed for the computation of values concerning the initial value problems on each spatial discretization level. This is the main part of the computational burden. The solution of all the condensed QPs (14) only takes 0.6 s, for instance. We observe that most of the effort is spent on levels 1 and 6. The effort on level 1 is due to the coarse grid computations in each iteration. Due to DAESOL-II’s memory requirements, we could not refine the fine grid further. Were we able to do so, then the numerical effort spent on the finest grid level $\ell = 7$ would dominate the overall effort even more.

7 Conclusions

We elaborated on the challenges and a numerical approach for a Direct Multiple Shooting method for optimal control problems with coupled ODE and parabolic PDE constraints. Even difficult boundary conditions like time periodicity of the states can be treated efficiently in this framework. The large-scale nature of the resulting NLPs can be tamed by a structure exploiting two-grid Newton-Picard inexact SQP method. Encouraging numerical results indicate that challenging PDE constrained optimal control problems can be solved efficiently with a Direct Multiple Shooting method. The proposed method does not require the often time-consuming derivation of adjoint equations and can be easily parallelized. It is in particular suited for practitioners who need an accurate resolution of the system dynamics and can live with lower resolution of the optimal control.

We would like to thank the unknown reviewer for the constructive comments on the initial manuscript. Support by the EU through S. Engell's and H.G. Bock's ERC Advanced Investigator Grant MOBOCON (291 458) is gratefully acknowledged.

References

1. Albersmeyer, J.: Adjoint based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg (2010)
2. Albersmeyer, J., Bock, H.G.: Sensitivity generation in an adaptive BDF-method. In: Bock, H.G., Kostina, E., Phu, X.H., Rannacher, R. (eds.) *Modeling, Simulation and Optimization of Complex Processes. Proceedings of the International Conference on High Performance Scientific Computing*, pp. 15–24, Hanoi, 6–10 March 2006. Springer, Berlin (2008)
3. Bellman, R.E.: *Dynamic Programming*, 6th edn. University Press, Princeton (1957)
4. Biegler, L.T.: Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput. Chem. Eng.* **8**, 243–248 (1984)
5. Bock, H.G.: Recent advances in parameter identification techniques for ODE. In: Deuffhard, P., Hairer, E. (eds.) *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pp. 95–121. Birkhäuser, Boston (1983)
6. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: *Proceedings of the 9th IFAC World Congress*, pp. 242–247, Budapest. Pergamon Press (1984)
7. Dautray, R., Lions, J.-L.: *Evolution problems I*. In: Craig, A. (ed.) *Mathematical Analysis and Numerical Methods for Science and Technology*, vol. 5. Springer, Berlin (1992)
8. Deuffhard, P.: *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, vol. 35. Springer, Berlin (2006)
9. Griewank, A., Walther, A.: *Evaluating Derivatives*, 2nd edn. SIAM, Philadelphia (2008)
10. Hesse, H.K.: Multiple shooting and mesh adaptation for PDE constrained optimization problems. Ph.D. thesis, University of Heidelberg (2008)
11. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: *Optimization with PDE Constraints*. Springer, New York (2009)
12. Lebedez, D., Brandt-Pollmann, U.: Manipulation of self-aggregation patterns and waves in a reaction-diffusion system by optimal boundary control strategies. *Phys. Rev. Lett.* **91**(20), 208301 (2003)

13. Lust, K., Roose, D., Spence, A., Champneys, A.R.: An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions. *SIAM J. Sci. Comput.* **19**(4), 1188–1209 (1998)
14. Nocedal, J., Wright, S.J.: *Numerical Optimization*, 2nd edn. Springer, Berlin (2006)
15. Osborne, M.R.: On shooting methods for boundary value problems. *J. Math. Anal. Appl.* **27**, 417–433 (1969)
16. Pontryagin, L.S., Boltyanski, V.G., Gamkrelidze, R.V., Miscenko, E.F.: *The Mathematical Theory of Optimal Processes*. Wiley, Chichester (1962)
17. Potschka, A.: A direct method for the numerical solution of optimization problems with time-periodic PDE constraints. Ph.D. thesis, Universität Heidelberg (2011)
18. Potschka, A.: *A Direct Method for Parabolic PDE Constrained Optimization Problems*. *Advances in Numerical Mathematics*. Springer, Berlin (2013)
19. Potschka, A., Bock, H.G., Schlöder, J.P.: A minima tracking variant of semi-infinite programming for the treatment of path constraints within direct solution of optimal control problems. *Optim. Methods Softw.* **24**(2), 237–252 (2009)
20. Potschka, A., Mommer, M.S., Schlöder, J.P., Bock, H.G.: Newton-Picard-based preconditioning for linear-quadratic optimization problems with time-periodic parabolic PDE constraints. *SIAM J. Sci. Comput.* **34**(2), 1214–1239 (2012)
21. Russell, R.D., Shampine, L.F.: A collocation method for boundary value problems. *Numer. Math.* **19**, 1–28 (1972)
22. Saad, Y.: *Iterative Methods for Sparse Linear Systems*, 2nd edn. SIAM, Philadelphia (2003)
23. Thomée, V.: *Galerkin Finite Element Methods for Parabolic Problems*. Springer Series in Computational Mathematics, vol. 25, 2nd edn. Springer, Berlin (2006)
24. Tröltzsch, F.: *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*, 2nd edn. Vieweg+Teubner Verlag, Wiesbaden (2009)
25. Tsang, T.H., Himmelblau, D.M., Edgar, T.F.: Optimal control via collocation and non-linear programming. *Int. J. Control.* **21**, 763–768 (1975)
26. Tyson, R., Lubkin, S.R., Murray, J.D.: Model and analysis of chemotactic bacterial patterns in a liquid medium. *J. Biol.* **38**, 359–375 (1999). doi:[10.1007/s002850050153](https://doi.org/10.1007/s002850050153)
27. Tyson, R., Lubkin, S.R., Murray, J.D.: A minimal mechanism for bacterial pattern formation. *Proc. R. Soc. B Biol. Sci.* **266**, 299–304 (1999)
28. Walther, A., Kowarz, A., Griewank, A.: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. Technical report, Institute of Scientific Computing, Technical University Dresden (2005)
29. Wloka, J.: *Partial Differential Equations*. Cambridge University Press, Cambridge (1987). Translated from the German by C.B. Thomas and M.J. Thomas