Generalized SQP-Methods with "Parareal" Time-Domain Decomposition for Time-dependent PDE-constrained Optimization

Stefan Ulbrich[*]

1 Introduction

We consider a time-dependent PDE-constrained optimization problem of the form

$$\min_{y \in Y, u \in U} \quad J(y,u) \quad \text{subject to} \quad C(y,u) = 0. \tag{1}$$

Here, $u \in U$ is the control living in a Banach space U, $y \in Y \subset C([0,T];B)$ is a time-dependent state with Banach spaces B and Y, where $B \subset L^2(\Omega)$ with a domain $\Omega \subset \mathbb{R}^n$. The state equation $C(y,u) = 0$ is the appropriate formulation of a time-dependent PDE (or a system of time-dependent PDEs)

$$\begin{aligned} &y_t + A(t,x,y,u) = 0, \quad (t,x) \in (0,T) \times \Omega \\ &y(0,x) = y_0(x), \quad x \in \Omega \end{aligned} \tag{2}$$

with initial data $y_0 \in B$. For convenience we assume that boundary conditions are incorporated in the state space Y.

[*] Fachbereich Mathematik, AG10, Technische Universität Darmstadt, Schlossgartenstr. 7, 64289 Darmstadt, Germany, E-mail: ulbrich@mathematik.tu-darmstadt.de

In many applications there are already robust and validated discretization schemes for (2) available. Lions, Maday, and Turinici [14] have observed that the time discretization can efficiently be parallelized by a time-domain decomposition technique, which the authors call the *parareal scheme*. The parareal scheme combines a coarse grid predictor and a parallel fine grid corrector. In this paper we propose a generalized SQP-algorithm that allows the fast parallel solution of (1) by using the parareal technique for the state equation and the adjoint equation. The generalized SQP-algorithm is inspired by the nonmonotone SQP-method of Ulbrich and Ulbrich [18] that does–similar to filter methods [7, 8]–not use a penalty function as merit function. Our aim is to achieve high effeciency of the optimization method on parallel computers while requiring minimal changes of existing sequential state solvers by the user. Besides the parallelizm provided by the time-domain decomposition the proposed method admits the application of highly efficient solvers inside the time domains, e.g., multigrid-based PDE solvers. These features allow the efficient solution of very large optimal control problems for time-dependent PDEs on parallel computers, while only minimal changes to existing state and adjoint solvers are necessary. In addition, the proposed approach is not restricted to special types of PDEs. It is only required that certain differentiability properties are satisfied, see Assumption 1, and that the state and adjoint solver are convergent.

Recently, several approaches have been proposed for the efficient and parallel solution of optimal control problems for time-dependent PDEs. Time-domain decomposition techniques for the optimal control of the wave equation were considered by Lagnese and Leugering [12, 13] and for distributed linear quadratic optimal control problems by Heinkenschloss [9]. In these works the time-domain decomposition is carried out for the optimality system and in contrast to the parareal technique no coarse grid propagator is used. By using the parareal idea Maday and Turinici propose in [15] a preconditioned steepest descent method for the optimal control of time-dependent PDEs. Other recent approaches consider space-time multigrid methods for the optimal control of PDEs. In particular, Borzi proposes in [5] a space-time multigrid method for parabolic distributed optimal control problems. In contrast to the present paper the approach in [5] does not allow to apply user-provided solvers, since a specific multigrid algorithm is used.

The paper is organized as follows. In section 2 we recall the parareal time-domain decomposition technique and state known convergence results. In section 3 we peform a time-domain decomposition of the optimal control problem (1) by applying the parareal technique to the state equation. Moreover, we derive optimality conditions for the decomposed problem and show that the parareal technique can also be applied to the adjoint system. In section 4 we propose a generalized SQP-method that allows the solution of (1) for arbitrary user-provided state solver and adjoint solver, where we use parareal solvers in the present paper. We prove global convergence of the method. In section 5 we use the generalized SQP-method with the specific choice of parareal solvers for state and adjoint equation. We demonstrate the efficiency of the approach by numerical results for the optimal control of a semilinear parabolic equation in 2D.

2 Parareal time-domain decomposition

The parareal algorithm was proposed by Lions, Maday, and Turinici [14] to speed up the numerical solution of time dependent partial differential equations by using parallel computers with a sufficiently large number of processors.

2.1 Description of the parareal method

We consider a time-dependent partial differential equation (or a system) of the general form

$$\begin{aligned} &y_t + A(t,x,y) = 0, \quad (t,x) \in (0,T) \times \Omega, \\ &y(0,x) = v_0(x), \quad x \in \Omega, \end{aligned} \tag{3}$$

where $y : [0,T] \to B$ maps time to a Banach space $B \subset L^2(\Omega)$, $v_0 \in B$ are initial data and $A(t,x,y)$ is a possibly time dependent partial differential operator in the variables $x \in \Omega$. The parareal technique, which was originally proposed in [14] and slightly modified in [2, 3], uses a decomposition

$$0 = T_0 < T_1 < \ldots < T_N = T$$

in time, which is uniform in the sense that

$$\eta_0 \Delta T \le T_{n+1} - T_n \le \Delta T, \quad \text{where} \;\; \Delta T := \max_{0 \le n < N} T_{n+1} - T_n.$$

We assume that on each time domain $[T_n, T_{n+1}]$, $0 \le n < N$, there exists a unique solution propagator

$$g(T_n, \cdot) : \quad v \in B \mapsto g(T_n, v) := y(T_{n+1}) \in B,$$

where y is the solution of

$$\begin{aligned} &y_t + A(t,x,y) = 0, \quad (t,x) \in (T_n, T_{n+1}) \times \Omega, \\ &y(T_n, x) = v(x), \quad x \in \Omega. \end{aligned} \tag{4}$$

Moreover, we assume that we have a coarse grid approximation $g_\Delta(T_n, v)$ of the exact propagator $g(T_n, v)$ at our disposal.

Example 1. *We will later use a backward Euler step*

$$\frac{g_\Delta(T_n, v) - v}{T_{n+1} - T_n} + A(T_{n+1}, x, g_\Delta(T_n, v)) = 0$$

as coarse propagator. As we will see the dissipativity of the backward Euler discretization is useful to stabilize the parareal scheme.

The parareal technique is based on a multiple shooting reformulation of the initial value problem (3). It combines parallel fine grid propagators with a coarse propagator for the iterative solution of the multiple shooting reformulation of (3).

More precisely, the parareal algorithm is defined as follows.

Algorithm 2.1. "Parareal" Time Integration Method:

Compute approximations y_k^n of $y^n = y(T_n)$, $1 \le n \le N$, for the solution y of (3) as follows.

1. Initialize by coarse scheme: $y_1^0 = v_0$,
$$y_1^{n+1} = g_\Delta(T_n, y_1^n), \quad 0 \le n < N.$$

2. For $k = 1, \dots, K-1$: $y_{k+1}^0 = v_0$
$$y_{k+1}^{n+1} = \underbrace{g_\Delta(T_n, y_{k+1}^n)}_{\text{predictor}} + \underbrace{(g(T_n, y_k^n) - g_\Delta(T_n, y_k^n))}_{\substack{\text{corrector, computable}\\ \text{in parallel on time slabs}}}, \quad 0 \le n < N. \tag{5}$$

The sequential predictor step propagates the new approximation y_{k+1}^n by the coarse propagator. The error is corrected by using the approximation y_k^n of the previous iteration and can thus be computed in parallel.

Remark 1.

- In practice $g(T_n, y_k^n)$ is replaced by an approximation $g_\delta(T_n, y_k^n)$ obtained by an accurate fine grid scheme.
- The exact solution $y^n = y(T_n)$ satisfies the parareal scheme, since $y^{n+1} = g(T_n, y^n)$.
- After N iterations the exact solution is obtained.
- The parareal scheme can directly be applied to nonlinear equations and is then a nonlinear iteration. Thus, user-provided nonlinear fine grid solvers can be used directly. Besides the fact that nonlinear solvers can be more robust and efficient for some problems the nonlinear parareal algorithm has the advantage that the state in the time slabs has not to be stored in contrast to, e.g., a Newton iteration with inner parareal solver for the linearized equation.

2.2 Convergence properties of the parareal algorithm

In this subsection we collect recent convergence results for the parareal scheme. The first result yields a convergence order km after k parareal iterations provided the solution is smooth enough and the coarse propagator has order m and is Lipschitz.

Theorem 1. (G. Bal [2])
Let $B_k \subset B_{k-1} \subset \ldots \subset B_0 = B$ be a scale of Banach spaces. Assume that

- (3) *is stable in all spaces* B_j, $0 \leq j \leq k$, *i.e.*

$$\|y(t)\|_{B_j} \leq C\|v_0\|_{B_j} \quad \forall\, v_0 \in B_j, \quad t \in [0,T].$$

- *The coarse propagator* g_Δ *is Lipschitz in the sense that for* $0 \leq j < k$

$$\max_{0\leq n<N} \|g_\Delta(T_n, v) - g_\Delta(T_n, w)\|_{B_j} \leq (1 + C\Delta T)\|v - w\|_{B_j} \quad \forall\, u, v \in B_j.$$

- *The coarse propagator* g_Δ *has order* m *in the sense that for* $0 \leq j < k$ *with* $\delta g(T_n, v) := g(T_n, v) - g_\Delta(T_n, v)$ *the estimate holds*

$$\max_{0\leq n<N} \|\delta g(T_n, v) - \delta g(T_n, w)\|_{B_j} \leq C(\Delta T)^{m+1}\|v - w\|_{B_{j+1}} \quad \forall\, v, w \in B_{j+1}.$$

Then for initial data $v_0 \in B_k$ *the parareal scheme after* k *iterations has an accuracy of order* mk, *i.e.,*

$$\max_{1\leq n\leq N} \|y(T_n) - y_k^n\|_{B_0} \leq C(\Delta T)^{mk}\|v_0\|_{B_k}$$

with C *independent of* ΔT *and* v_0.

Proof. A proof can be found in [2, Thm. 1]. Our last assumption is a modification of assumption (H3) in [2], which is taylored directly to the proof of [2, Thm. 1]. □

This general result requires additional regularity of the solution in order to achieve order km. By exploiting dissipative effects of the coarse propagator, Bal [2] shows an improved result for certain linear partial differential operators by using Fourier analysis.

Theorem 2. (G. Bal, [2])
Consider the case of a linear M*-th order 1D spatial operator* A *with constant coefficients and symbol* $P(\xi) = \alpha_0 + \sum_{k=1}^{M-1} \alpha_k \xi^k + |\xi|^M$, *where* $\alpha_k \in \mathbb{C}$ *and* $\alpha_0 \geq 0$ *such that* $P(\xi) \geq 0$ *or* $\Re(P(\xi)) > 0$ *(e.g., heat equation, advection-diffusion equation). If for* $T_n = n\Delta T$, $\Delta T = T/N$, *the coarse propagator is given by the* θ*-scheme*

$$\frac{g_\Delta(T_n, v) - v}{\Delta T} + A(\theta g_\Delta(T_n, v) + (1 - \theta)v) = 0$$

then for $\theta \in (1/2, 1]$ *the parareal scheme is stable and*

$$\max_{1\leq n\leq N} \|y(T_n) - y_k^n\|_{H^s(\mathbb{R})} \leq C(\Delta T)^{mk}\|v_0\|_{H^s(\mathbb{R})}.$$

for all $s \in \mathbb{R}$. *Here,* $H^s(\mathbb{R})$ *denotes the usual Sobolev space of order* s.

Proof. As shown in [2], the result follows from Theorem 2 and 3 in [2]. □

2.3 Interpretation as preconditioned iteration

Rather than interpreting the parareal scheme as a scheme of higher order on the coarse grid, we prefer to view the parareal scheme as a preconditioned iteration. Consider for simplicity the case that (3) is linear. Then

$$g(T_n, y^n) = G_n y^n + c_n, \quad g_\Delta(T_n, y^n) = G_{\Delta,n} y^n + c_{\Delta,n}.$$

The relation

$$y^{n+1} = g(T_n, y^n) \quad 0 \le n < N, \quad y^0 = v_0$$

can be written as

$$\begin{pmatrix} I & & & & \\ -G_0 & I & & & \\ & -G_1 & I & & \\ & & \ddots & \ddots & \\ & & & -G_{N-1} & I \end{pmatrix} \begin{pmatrix} y^0 \\ y^1 \\ \vdots \\ y^N \end{pmatrix} = \begin{pmatrix} v_0 \\ c_0 \\ \vdots \\ c_{N-1} \end{pmatrix}$$

which we write short as

$$My = F.$$

Since the parareal update (5) can be written as

$$y_{k+1}^{n+1} - g_\Delta(T_n, y_{k+1}^n) = y_k^{n+1} - g_\Delta(T_n, y_k^n) + g(T_n, y_k^n) - y_k^{n+1}, \quad 0 \le n < N,$$

we see that with the coarse grid approximation M_Δ of M

$$M_\Delta = \begin{pmatrix} I & & & & \\ -G_{\Delta,0} & I & & & \\ & -G_{\Delta,1} & I & & \\ & & \ddots & \ddots & \\ & & & -G_{\Delta,N-1} & I \end{pmatrix}$$

the parareal scheme (5) can be written as

$$y_{k+1} = y_k + M_\Delta^{-1}(F - My_k). \tag{6}$$

Theorem 1 and 2 show that under appropriate assumptions M_Δ^{-1} is a good preconditioner for the operator M.

To cover also the nonlinear case we introduce the operators

$$\mathcal{G}(y) = \begin{pmatrix} y^0 - v_0 \\ y^1 - g(T_0, y^0) \\ \vdots \\ y^N - g(T_{N-1}, y^{N-1}) \end{pmatrix}, \quad \mathcal{G}_\Delta(y) = \begin{pmatrix} y^0 - v_0 \\ y^1 - g_\Delta(T_0, y^0) \\ \vdots \\ y^N - g_\Delta(T_{N-1}, y^{N-1}) \end{pmatrix}.$$

Then the parareal scheme can be written as the preconditioned iteration

$$\mathcal{G}_\Delta(y_{k+1}) = \mathcal{G}_\Delta(y_k) - \mathcal{G}(y_k)$$

which is (6) in the linear case.

3 Time-domain decomposition of the optimal control problem

We use a multiple shooting reformulation of the optimal control problem (1). The proposed optimization algorithm, however, is different from SQP type optimization algorithms typically applied to multiple shooting reformulations of optimal control problems. In particular, our SQP algorithms are matrix free and, in the context of (1), use the parareal technique for the solution of the state equation and the adjoint equation.

Let as above $0 = T_0 < T_1 < \ldots < T_N = T$. We consider a PDE-constrained optimization problem (1) with state equation (2), where the cost functional J depends for simplicity only on the end states $y(T_n)$ of the time domains, i.e.,

$$J(y,u) = J^{\Delta}(y(T_0),\ldots,y(T_N),u). \tag{7}$$

Assume that for all $v \in B$, $u \in U$

$$\begin{aligned} &y_t + A(t,x,y(t),u(t)) = 0, \quad (t,x) \in (T_n,T_{n+1}) \times \Omega \\ &y(T_n,x) = v(x), \quad x \in \Omega \end{aligned}$$

has a unique solution and define the corresponding propagator

$$g(T_n,v;u) := y(T_{n+1}) \in B.$$

Using $y_\Delta = (y^0,\cdots,y^N) \in B^{N+1} =: Y_\Delta$ as new state we can rewrite (1), (2), (7) as

$$\min_{y_\Delta=(y^0,\cdots,y^N)\in Y_\Delta, u\in U} \quad J^{\Delta}(y,u) \quad \text{subject to} \quad C^{\Delta}(y,u) = 0, \tag{8}$$

where $C^{\Delta} : Y_\Delta \times U \to Y_\Delta$,

$$C^{\Delta}(y_\Delta,u) = \begin{pmatrix} y^0 - v_0 \\ y^1 - g(T_0,y^0;u) \\ \vdots \\ y^N - g(T_{N-1},y^{N-1};u) \end{pmatrix}. \tag{9}$$

Obviously, the parareal algorithm can be used as an iterative algorithm to solve the decomposed state equation (9) in parallel. In the following we will develop an optimization algorithm for (8) that is motivated by the following basic concept.

- Use the parareal scheme as state solver in a generalized inexact SQP-algorithm.
- Use the parareal scheme also for the adjoint equation to compute inexact gradients of the cost functional.

To develop the generalized SQP-method we state first necessary optimality conditions for (8).

3.1 Optimality conditions

The Lagrangian function for (8) is given by

$$L(y_\Delta, u, \lambda) = J^\Delta(y_\Delta, u) + \langle \lambda, C^\Delta(y^\Delta, u)\rangle_{Y_\Delta^*, Y_\Delta}$$

By standard optimality theory we obtain the following result.

Proposition 3. *Assume that* $J^\Delta : Y_\Delta \times U \to \mathbb{R}$, $C^\Delta : Y_\Delta \times U \to Y_\Delta$ *are continuously differentiable. If* $(\bar{y}_\Delta, \bar{u})$ *is a local solution of* (8) *then there exists a Lagrange multiplier (adjoint state)* $\bar{\lambda} \in Y_\Delta^*$ *with*

$$\begin{aligned}
L_y(\bar{y}_\Delta, \bar{u}, \bar{\lambda}) &= J_y^\Delta(\bar{y}_\Delta, \bar{u}) + (C_y^\Delta)^*(\bar{y}_\Delta, \bar{u})\bar{\lambda} = 0 && \textit{(Adjoint equation)} && (10)\\
L_u(\bar{y}_\Delta, \bar{u}, \bar{\lambda}) &= J_u^\Delta(\bar{y}_\Delta, \bar{u}) + (C_u^\Delta)^*(\bar{y}_\Delta, \bar{u})\bar{\lambda} = 0 && \textit{(Stationarity)} && (11)\\
C^\Delta(\bar{y}_\Delta, \bar{u}) &= 0 && \textit{(State equation).} && (12)
\end{aligned}$$

Proof. The structure of C^Δ implies immediately that $C_y^\Delta(\bar{y}_\Delta, \bar{u})$ is surjective. Hence, $C_{(y,u)}^\Delta(\bar{y}_\Delta, \bar{u})$ is surjective, which is a regularity condition for $(\bar{y}_\Delta, \bar{u})$. Therefore it is well known that the Karush-Kuhn-Tucker conditions (10), (11), (12) hold at a local minimum $(\bar{y}_\Delta, \bar{u})$ of (8). □

3.2 Structure of the adjoint aquation

We have with $G_n(y^n, u) := g_{y^n}(T_n, y^n; u)$, which is the propagator for the linearized state equation,

$$C_y^\Delta(y, u) = \begin{pmatrix} I & & & & \\ -G_0 & I & & & \\ & -G_1 & I & & \\ & & \ddots & \ddots & \\ & & & -G_{N-1} & I \end{pmatrix}$$

Thus, we obtain the following structure of the adjoint equation.

$$(C_y^\Delta)^*(y_\Delta, u)\lambda + J_y^\Delta(y_\Delta, u) = \begin{pmatrix} I & -G_0^* & & & \\ & I & -G_1^* & & \\ & & \ddots & \ddots & \\ & & & I & -G_{N-1}^* \\ & & & & I \end{pmatrix} \begin{pmatrix} \lambda^0 \\ \vdots \\ \lambda^N \end{pmatrix} + J_y^\Delta(y_\Delta, u) = 0. \tag{13}$$

Here, G_n^* is the propagator for the adjoint equation on time domain n backwards in time. Therefore, the parareal scheme is directly applicable backwards in time to the adjoint equation.

In the rest of the paper we will assume that we have a parareal solver for the state equation (9) and a parareal solver for the adjoint equation (13) at our disposal.

4 A generalized SQP-method

We propose in this section a generalized SQP-algorithm for the solution of the problem

$$\min_{y\in Y, u\in U} \quad J(y,u) \quad \text{subject to} \quad C(y,u)=0. \tag{1}$$

Motivated by the time-dependent PDE-constrained problem (8) the design of the proposed optimization algorithm offers the following features.

- The algorithm solves state equation and optimization problem simultaneously.
- Arbitrary user-provided iterative solvers for state equation and adjoint equation can be used, in our case parallel parareal solvers.
- The optimization algorithm requires only very few iterations of the solvers in each step by controlling the inexactness.
- The algorithm ensures convergence to a stationary point.

For the considered time-dependent problem we have in addition the following properties.

- A nonlinear parareal solver can be used for nonlinear state equations.
- Only the state on the time-domain interfaces has to be stored, the states in the interior of the time domains can be handeled locally by the local solver of the time domain.

4.1 Basic assumptions

We will work under the following assumptions.

Assumption 1. *With $X := Y \times U$ the following holds for a given open convex set $X_0 = Y_0 \times U_0 \subset X$*

- *The mappings*

$$\begin{aligned}(y,u) &\in X \mapsto J(y,u)\\ (y,u) &\in X \mapsto C(y,u) \in Z\end{aligned}$$

  are continuously differentiable and the derivatives are uniformly bounded and Lipschitz on X_0.

- *For any $u \in U_0$ there exists a unique solution $y(u) \in Y_0$ of $C(y(u),u)=0$.*
- *The derivative $C_y(y,u) \in \mathcal{L}(Y,Z)$ has an inverse that is uniformly bounded for all $(y,u) \in X_0$.*

4.2 Review of classical trust-region SQP-methods

Classical SQP-method

It is well known that the local SQP-method is given by the iteration iteration

$$x_{k+1} = x_k + s_k,$$

where s_k solves the *SQP-Problem* at $(x_k, \lambda_k) \in X \times Z^*$

$$\begin{aligned} &\min_{s \in X} \quad q_k(s) := L(x_k, \lambda_k) + \langle L_x(x_k, \lambda_k), s \rangle_{X^*,X} + \frac{1}{2} \langle s, H_k s \rangle_{X,X^*} \\ &\text{subject to} \;\; C(x_k, \lambda_k) + C_x(x_k, \lambda_k) s = 0 \end{aligned}$$

with $x_k = (y_k, u_k)$ and $H_k = L_{xx}(x_k, \lambda_k)$ (or an approximation). In *reduced SQP-methods* the quadratic model $q_k(s_y, s_u)$ is reduced to the control component s_u by choosing λ_k according to the adjoint equation

$$L_y(x_k, \lambda_k) = 0 \tag{14}$$

and by replacing H_k by

$$B_k := \begin{pmatrix} 0 & 0 \\ 0 & \hat{B}_k \end{pmatrix},$$

where $\hat{B}_k$ is an approximation of the reduced Hessian

$$\hat{H}_k = W_k^* H_k W_k, \quad W_k = \begin{pmatrix} -(C_y^{-1} C_u)(x_k) \\ I_U \end{pmatrix}.$$

Here, I_U denotes the identity on U. This leads to the *reduced SQP-Problem* at (x_k, λ_k)

$$\begin{aligned} &\min_{s=(s_y,s_u) \in X} \quad \hat{q}_k(s_u) := L(x_k, \lambda_k) + \langle L_u(x_k, \lambda_k), s_u \rangle_{U^*,U} + \frac{1}{2} \langle s_u, \hat{B}_k s_u \rangle_{U,U^*} \\ &\text{subject to} \;\; C(x_k, \lambda_k) + C_y(x_k, \lambda_k) s_y + C_u(x_k, \lambda_k) s_u = 0. \end{aligned}$$

Trust-region globalization

Following the composite-step approach by Byrd and Omojokun [17], see also [6], a trust region globalization of SQP-methods can be obtained by a step decomposition

$$s_k = s_k^n + s_k^t$$

with *quasi-normal step* s_k^n and a *tangential step* s_k^t. Here, the quasi-normal step $s_k^n = (s_{y,k}^n, 0)$ updates only the state and is obtained as an approximate solution of the quasi-normal problem

$$\min_{s_y^n \in Y} \|C(x_k) + C_y(x_k) s_y^n\|_Z^2 \quad \text{subject to} \quad \|s_y^n\|_Y \leq \delta_k, \tag{15}$$

where $\delta_k > 0$ is the current trust-region-radius. The tangential step s_k^t is an approximate solution of the *tangential problem*

$$\min_{s^t=(s_y^t,s_u^t)\in X} q_k(s_k^n + s^t) \quad \text{subject to} \quad C_x(x_k)s^t = 0, \quad \|s_u^t\|_U \le \delta_k.$$

For reduced SQP-methods the problem simplifies to: Compute an approximate solution $s_{u,k}^t$

$$\min_{s_u^t\in U} \hat{q}_k(s_u^t) \quad \text{subject to} \quad \|s_u^t\|_U \le \delta_k, \tag{16}$$

then compute $s_{y,k}^t$ by solving

$$C_y(x_k)s_{y,k}^t = -C_u(x_k)s_{u,k}^t.$$

The evaluation of the step $s_k = s_k^n + s_k^t$ and the adaption of the trust-region radius can now be performed in various ways, e.g., by using a merit function [6] or by using techniques without merit function such as filter-techniques [7, 8] or nonmonotone trust-region techniques [18].

4.3 Development of the generalized SQP-algorithm

The classical SQP-method has several drawbacks that limit the flexibility of the method.

- The SQP-method uses a Newton iteration for the simultaneous solution of the state equation. But often the user provides already his own efficient and robust solver (sometimes nonlinear).
- For nonlinear state equations nonlinear solvers can be just as efficient as the solver for the linearized equation (e.g., nonlinear parareal solvers).
- The linearized state equation can be inappropriate as approximation of the state equation (for example for flows with shocks [19, 20]).

We circumvent these limitations by using nonmonotone trust-region techniques similar to [18] together with a generalized quasi-normal step. Again, we use a step decomposition

$$s_k = s_k^n + s_k^t.$$

The generalized tangential step $s_k^t = (s_{k,y}^t, s_{k,u}^t)$ is obtained as follows.

Generalized tangential step

Compute $s_{k,u}^t$ as an approximate solution of the reduced tangential problem

$$\min_{s_u^t\in U} \hat{q}_k(s_u^t) \quad \text{subject to} \quad \|s_u^t\|_U \le \delta_k \tag{16}$$

satisfying a fraction of Cauchy-decrease condition (FCD)

$$\hat{q}_k(0) - \hat{q}_k(s_{k,u}^t) \ge \kappa_q \|L_u(x_k,\lambda_k)\|_{U^*} \min\{\|L_u(x_k,\lambda_k)\|_{U^*}, \delta_k\}. \tag{17}$$

with a constant $\kappa_q > 0$. Then we set

$$s_k^t = (s_{k,y}^t, s_{k,u}^t), \tag{18}$$

where we require only that

$$\|s_{k,y}^t\|_Y \le \kappa_t \|s_{k,u}^t\|_U \tag{19}$$

with a fixed constant $\kappa_t > 0$. The choice $s_{k,y}^t = 0$ is allowed.

Remark 2. *It is well known that (FCD) holds if $\|\hat{B}_k\|_{U,U^*}$ is uniformly bounded and if s_k^t yields at least a fixed fraction of the Cauchy decrease, which is the decrease along the steepest descent direction $-\nabla \hat{q}_k(0)$ inside the trust region.*

Inexact adjoint state

Instead of choosing the multiplier λ_k as the exact solution of the adjoint equation (14) we admit inexact adjoint states λ_k satisfying

$$\|L_y(x_k, \lambda_k)\|_{Y^*} \le \kappa_\lambda \min\{\|L_u(x_k, \lambda_k)\|_{U^*}, \delta_k\} \tag{20}$$

with a constant $\kappa_\lambda > 0$. This requirement is similar to the inexactness framework of Heinkenschloss and Vicente [10].

Generalized quasi-normal step

Instead of the Newton-like quasi-normal step (15) we invoke a user-provided iterative solver, in our case the parareal state solver, to compute a generalized quasi-normal step

$$s_k^n = (s_{k,y}^n, 0) \tag{21}$$

such that the following non-monotone decrease condition holds

$$\max\left\{R_k, \|C(x_k)\|_Z^2\right\} - \|C(x_k + s_k)\|_Z^2 \ge \kappa_c \|C(x_k)\|_Z^2, \quad \kappa_c \in (0,1), \tag{22}$$

where we require that with constants $\delta_c > 0$, $\kappa_n > 0$ holds

$$\|s_{k,y}^n\|_Y \le \kappa_n(\|C(x_k)\|_Z + \delta_k) \quad \text{if } \|C(x_k)\|_Z + \delta_k \le \delta_c. \tag{23}$$

Here, $R_k \ge \|C(x_k)\|_Z^2$, $R_k > 0$, is a relaxation term which is chosen as $R_k = \|C(x_k)\|_Z^2$ unless feasibility is much better than stationarity, i.e., $\|C(x_k)\|_Z \ll \|\nabla_u L(x_k, \lambda_k)\|_{U^*}$.

Remark 3. *If Assumption 1 holds with a convex ε-neighborhood X_0 of all x_k, $x_k + s_k^t$ then (23) can always be ensured. In fact, under this assumption the Newton-Kantorovich theorem ensures the existence of some $\delta > 0$ such that the following holds: in the case $\|C(x_k + s_k^t)\|_Z \le \delta$ Newton's method for the equation*

$$C(y_k + s_{k,y}^t + z, u_k + s_{k,u}^t) = 0 \tag{24}$$

starting with $z_0 = 0$ generates iterates z_j that converge with linear rate $1/2$ to a point z^* with $C(y_k + s^t_{k,y} + z^*, u_k + s^t_{k,u}) = 0$ and the iteration stays in a ball of radius $\leq 2M\|C(x_k + s^t_k)\|_Z \leq 2M\delta$, where M is a bound for $\|C_y^{-1}\|_{Z,Y}$ on X_0 (Assumption 1 allows a uniform choice for δ and M). But since C is Lipschitz on X_0 with a constant L_C by Assumption 1 and since $\|s^t_k\|_X \leq (1 + \kappa_t)\delta_k$, we have

$$\|C(x_k + s^t_k)\|_Z \leq \|C(x_k)\|_Z + L_C(1 + \kappa_t)\delta_k \leq (1 + L_C)(1 + \kappa_t)(\|C(x_k)\|_Z + \delta_k)$$

and hence the condition $\|C(x_k + s^t_k)\|_Z \leq \delta$ is satisfied whenever $\|C(x_k)\|_Z + \delta_k \leq \delta_c$ with $\delta_c = \delta(1 + L_C)^{-1}(1 + \kappa_t)^{-1}$. Moreover, the iteration stays in a ball of radius $\leq \kappa_n(\|C(x_k)\|_Z + \delta_k)$ with $\kappa_n = (1 + L_C)(1 + \kappa_t)2M$. Since $R_k > 0$ and $R_k \geq \|C(x_k)\|^2_Z$, (22) will eventually hold with $x_k + s_k = (y_k + s^t_{k,y} + z_j, u_k + s^t_{k,u})$ and the resulting $s^n_{k,y} = z_j$ satisfies (23) for the above choices of δ_c and κ_n.

If $\|s^t_k\|_X \leq \kappa_s\|C(x_k)\|_Z$ with a constant $\kappa_s > 0$, then there exists $\delta > 0$ small enough, such that in the case $\|C(x_k + s^t_k)\|_Z \leq \delta$ the decrease condition (22) holds after a single Newton step by the local quadratic convergence.

The above considerations show, that also any solver for (24) which converges for the starting point $z_0 = 0$ to the closest point z^* with $C(y_k + s^t_{k,y} + z^*, u_k + s^t_{k,u}) = 0$ will eventually generate an iterate $s^n_{k,y} = z_j$ satisfying (22), (23).

The purpose of the relaxation term R_k is to allow a controlled increase of the constraint violation if the current point is almost feasible but the stationarity is unsatisfactory. This is essential to obtain an efficient All-at-Once method, since otherwise the algorithm would produce almost feasible iterates as soon as an almost feasible point was generated.

Similar as in [18] we use the following algorithm to adjust R_k.

Algorithm 4.1. Computation of the relaxation parameter

Input: k, j_k, $\|L_u(x_k, \lambda_k)\|_{U^*}$, $\|C(x_k)\|_Z$

Output: R_k, j_{k+1}

Given: Tolerance sequence $a_j \searrow 0$, $\dfrac{a_{j+1}}{a_j} \geq \alpha_0 > 0$, $\alpha, \beta \in]0, \frac{1}{2}[$.

If $\|C(x_k)\|_Z \geq \min\{\alpha a_{j_k}, \beta\|L_u(x_k, \lambda_k)\|_{U^*}\}$: set $R_k := \|C(x_k)\|^2_Z$, $j_{k+1} := j_k$.

Else: set $R_k := \min\left\{a^2_{j_k}, \|\nabla_u L_u(x_k, \lambda_k)\|^2_{U^*}\right\}$, $j_{k+1} := j_k + 1$.

Remark 4. *In [18] it is in addition assumed that the tolerance sequence decays slowly enough that*

$$\sum_{j=0}^{\infty} a^\eta_j < \infty$$

with a fixed constant $\eta > 4/3$. This is necessary to allow transition to fast local convergence when classical tangential step and quasi-normal step are used.

Evaluation of steps

For the evaluation of steps we adapt the merit-function-free strategy in [18] that provides a similar acceptance behavior as filter methods [7, 8] and is convenient to integrate in our generalized SQP-framework. The basic idea is to compare the model reduction

$$\Delta q_k(s_k^t) := \hat{q}_k(0) - \hat{q}_k(s_{k,u}^t)$$

obtained in the quasi-tangential step with the actual reduction

$$\Delta L_k(s_k) := L_k(x_k, \lambda_k) - L_k(x_k + s_k, \lambda_k)$$

of the Lagrangian function if the model reduction $\Delta q_k(s_k^t)$ is sufficiently large compared to the constraint violation. More precisely, if

$$\Delta q_k(s_k^t) \geq (\nu \|C(x_k)\|_Z^2)^\mu \tag{25}$$

with constants $\nu > 0$, $\mu \in (2/3, 1)$, then we accept the step if in addition

$$\Delta L_k(s_k) \geq \eta_1 \Delta q_k(s_k^t)$$

with a constant $\eta_1 \in (0, 1)$.

If (25) is violated then we always accept the step. In this case (22) ensures a non-monotone decrease of the constraint violation.

Remark 5. *In this paper we could allow $\mu \in (0, 1)$ in (25). The range $\mu \in (2/3, 1)$ allows transition to fast local convergence when classical tangential step and quasi-normal step are used, see [18].*

The generalized SQP-method

Combining all ingredients we obtain the following algorithm.

Algorithm 4.2. Generalized SQP-method without merit function

Let $\eta_1 \in (0, 1)$, $0 < \gamma_1 < 1 < \gamma_2$, $\nu > 0$, $\mu \in]\frac{2}{3}, 1[$.
Choose $x_0 \in X$, $\delta_0 \geq \delta_{min} > 0$, $\hat{B}_0 \in \mathcal{L}(U, U^*)$, and set $j_0 := 0$, $k := 0$.

1. Compute $C(x_k)$, $J(x_k)$.

2. Compute an approximate adjoint state λ_k satisfying (20).

3. If $\|C(x_k)\|_Z + \|L_x(x_k)\|_{X^*} = 0$: STOP $\quad$ (x_k is KKT-point)

4. Compute the generalized tangential step s_k^t acoording to (16), (18) satisfying (17), (19).

5. Determine the relaxation parameter R_k and j_{k+1} according to Algorithm 4.1.

6. Starting from $x_k + s_k^t$ compute a generalized quasi-normal step s_k^n satisfying (21), (22).

7. If $\Delta q_k(s_k^t) \geq (\nu \|C(x_k)\|_Z^2)^\mu$ goto 8 else goto 9.

8. If $\Delta L_k(s_k) < \eta_1 \, \Delta q_k(s_k^t)$ then set $x_{k+1} = x_k$, $j_{k+1} = j_k$. (step rejected)
   Set $\delta_{k+1} = \gamma_1 \delta_k$, $k := k+1$ and goto 2.

9. Set $x_{k+1} = x_k + s_k^t + s_k^n$, compute new Hessian update $\hat{B}_{k+1}$. (step successful)
   Choose $\delta_{k+1} \in [\max\{\delta_{min}, \delta_k\}, \max\{\delta_{min}, \gamma_2 \delta_k\}]$.
   Set $k := k+1$ and goto 1.

4.4 Convergence result

We will prove global convergence under the following assumption.

Assumption 2.

- *Assumption 1 holds for an open convex ε-neighborhood $X_0 = Y_0 \times U_0$ containing all trial points x_k, $x_k + s_k$, $x_k + s_k^t$.*
- *There is a constant $M_B > 0$ with*
$$\|\hat{B}_k\|_{U,U^*} \leq M_B \quad \textit{for all } k.$$
- *λ_k remains uniformly bounded.*
- *$J(x_k)$ is bounded from below.*

Well definedness

Lemma 4. *Let Assumption 2 hold. If Algorithm 4.2 does not terminate finitely then it generates an infinite sequence of successful steps.*

Proof. We have already mentioned in Remarks 2 and 3 that the requirements on s_k^t and s_k^n can be met.

Assume that there is $K > 0$ such that all steps s_k with $k \geq K$ are not successful. Since the algorithm does not terminate, we have with a constant $\varepsilon > 0$

$$\|C(x_k)\|_Z + \|L_x(x_k, \lambda_k)\|_{X^*} \geq 2\varepsilon.$$

We will derive a contradiction.

By the mechanism of the algorithm all iterations $k \geq K$ can only be unsuccessful if

$$\delta_{k+1} = \gamma_1 \delta_k \to 0 \quad \text{and} \quad \Delta q_k(s_k^t) \geq (\nu \|C(x_k)\|_Z^2)^\mu \quad \forall \, k \geq K. \tag{26}$$

Since

$$\Delta q_k(s_k^t) \leq \|L_u(x_k, \lambda_k)\|_{U^*} \delta_k + \frac{1}{2} \|\hat{B}_k\|_{U,U^*} \delta_k^2 \leq c_1(\delta_k + \delta_k^2), \tag{27}$$

we conclude with (26) that

$$0 \le (\nu \|C(x_k)\|_Z^2)^\mu \le \Delta q_k(s_k^t) \to 0 \quad \text{for } k \to \infty$$

and therefore

$$\|C(x_k)\|_Z \to 0, \quad \delta_k \to 0 \tag{28}$$

(we have even $\|C(x_k)\|_Z = \|C(x_K)\|_Z = 0$ for $k \ge K$ but for later reference it is useful to use only (28)). Hence, we find $K' \ge K > 0$ with

$$\|L_x(x_k, \lambda_k)\|_{X^*} \ge \varepsilon \quad \forall\, k \ge K'.$$

Then we have either $\|L_u(x_k, \lambda_k)\|_{U^*} \ge \varepsilon/2$ or $\|L_y(x_k, \lambda_k)\|_{X^*} \ge \varepsilon/2$ and thus by (20)

$$\|L_u(x_k, \lambda_k)\|_{U^*} \ge \frac{\varepsilon}{2 + 2\kappa_\lambda} =: \kappa_1 \varepsilon \quad \forall\, k \ge K'. \tag{29}$$

Hence, the quasi-tangential step leads by (17) and (26) to a model decrease with

$$\begin{aligned} \Delta q_k(s_k^t) &\ge \kappa_q \kappa_1 \varepsilon \min\{\kappa_1 \varepsilon, \delta_k\} \quad \forall\, k \ge K' \\ \Delta q_k(s_k^t) &\ge \nu^\mu \|C(x_k)\|_Z^{2\mu} \quad \forall\, k \ge K. \end{aligned} \tag{30}$$

We show that this eventually implies

$$\Delta L_k(s_k) \ge \eta_1 \Delta q_k(s_k^t)$$

and thus the step would be accepted in contradiction to our assumption. In fact, we find by (28) some $K'' \ge K'$ with

$$\|C(x_k)\|_Z + \delta_k \le \delta_c \quad \forall\, k \ge K''.$$

Thus (23) yields

$$\|s_k^n\|_X \le \kappa_n (\|C(x_k)\|_Z + \delta_k) \quad \forall\, k \ge K''$$

and we have with some $\xi \in [0, 1]$

$$\begin{aligned} |\Delta L_k(s_k) - \Delta q_k(s_k^t)| &\le \|L_x(x_k + \xi s_k, \lambda_k) - L_x(x_k, \lambda_k)\|_{X^*} \|s_k\|_X \\ &\quad + \|L_y(x_k, \lambda_k)\|_{Y^*} \|s_{k,y}\|_Y + \frac{1}{2} \|\hat{B}_k\|_{U,U^*} \|s_{k,u}^t\|_U^2 \\ &\le c_2 \|s_k\|_X^2 \le c_2 (1 + \kappa_t + \kappa_n)^2 (\|C(x_k)\|_Z + \delta_k)^2. \end{aligned}$$

Since $\mu \in (2/3, 1)$, a comparison with (30) shows that there is $0 < \delta \le \delta_c$ with

$$|\Delta L_k(s_k) - \Delta q_k(s_k^t)| \le (1 - \eta_1) \Delta q_k(s_k^t) \quad \text{for all } k \ge K' \text{ with } \|C(x_k)\|_Z + \delta_k \le \delta$$

and thus

$$\Delta L_k(s_k) \ge \eta_1 \Delta q_k(s_k^t) \quad \text{for all } k \ge K' \text{ with } \|C(x_k)\|_Z + \delta_k \le \delta.$$

Since $\|C(x_k)\|_Z + \delta_k \to 0$ by (28) this shows that the step would eventually be successful which contradicts our assumption. □

Convergence to feasible points

Lemma 5. *If $R_k = \|C(x_k)\|_Z^2$ for all successful iterations $k \geq K$ then*

$$\lim_{k\to\infty} \|C(x_k)\|_Z = 0.$$

Proof. In this case we have

$$\|C(x_k)\|_Z^2 - \|C(x_{k+1})\|_Z^2 \geq \kappa_c \|C(x_k)\|_Z^2 \quad \forall\, k \geq K \;\text{ with }\; x_{k+1} \neq x_k,$$

and thus $\|C(x_{k+1})\|_Z^2 \leq (1-\kappa_c)\|C(x_k)\|_Z^2$ for all successful iterations $k \geq K$. Since $C(x_{k+1}) = C(x_k)$ for rejected steps and since we have infinitely many successful steps by Lemma 4, we conclude that $\|C(x_k)\|_Z \to 0$ □

Otherwise, the following lemma applies.

Lemma 6. *If k is a successful iteration and $j_{k+1} = j_k + 1$ in Algorithm 4.1 then*

$$\|C(x_l)\|_Z \leq a_{j_k} \quad \forall\, l \geq k.$$

Moreover, we have

$$\|C(x_k)\|_Z \leq a_{j_k} \quad \forall\, k \quad \textit{with} \;\; j_k \geq 1.$$

Proof. Consider Algorithm 4.1 for a successful iteration k. If $j_{k+1} = j_k + 1$ then we must have

$$\|C(x_k)\|_Z < \min\{\alpha a_{j_k}, \beta \|L_u(x_k)\|_{U^*}\} \leq \alpha a_{j_k}.$$

We now show by induction

$$\|C(x_l)\|_Z \leq a_{j_k} \quad \forall\; l \geq k. \tag{31}$$

Since x_k and j_k change only for successful iterations it is sufficient to show (31) only for successful iterations l. For $l = k$ (31) is already shown. Now we have by using that $R_l \leq \max\left\{\|C(x_l)\|_Z^2, a_{j_l}^2\right\}$ together with (22)

$$\|C(x_{l+1})\|_Z^2 \leq \max\left\{R_l, \|C(x_l)\|_Z^2\right\} \leq \max\left\{a_{j_l}^2, \|C(x_l)\|_Z^2\right\} \leq a_{j_k}^2.$$

To show the second assertion we observe that $\|C(x_k)\|_Z \leq a_{j_k}$ is already proven for every successful iterations k with $j_{k+1} = j_k + 1$ (and this includes the first iteration with $j_{k+1} = 1$). Now it follows also for all $l > k$ with $j_l = j_k$ that

$$\|C(x_l)\|_Z \leq a_{j_k} = a_{j_l}.$$

□

Together we obtain the following result.

Theorem 7. *We have*

$$\lim_{k\to\infty} \|C(x_k)\|_Z = 0.$$

Proof. If $R_k = \|C(x_k)\|_Z^2$ for all successful iterations $k \geq K$ this follows from Lemma 5. Otherwise we have by Algorithm 4.1 that $j_{k+1} = j_k + 1$ for an infinite sequence of successful iterations. Thus, $j_k \to \infty$ and we find $K > 0$ with $j_k \geq 1$ for all $k \geq K$. The previous lemma now yields

$$\|C(x_k)\|_Z \leq a_{j_k} \quad \forall\, k \geq K$$

and thus $\|C(x_k)\|_Z \to 0$, since $a_{j_k} \to 0$. □

Global convergence result

We have the following convergence result.

Theorem 8. *Let Assumption 2 hold. If Algorithm 4.2 does not terminate finitely then it generates an infinite sequence of iterates with*

$$\lim_{k\to\infty} \|C(x_k)\|_Z = 0, \quad \liminf_{k\to\infty} \|\nabla_x L_k\|_{X^*} = 0.$$

We start with the following auxiliary result.

Lemma 9. *Let Assumption 2 hold and assume that*

$$\|\nabla_x L_k\|_{X^*} \geq \varepsilon > 0 \quad \forall\, k \geq K'.$$

Then there is $K \geq K'$ and a constant $\delta > 0$ such that

$$\delta_k \geq \gamma_1 \min\{\delta_{min}, \delta/2\} =: \delta' \quad \textit{for all successful iterations } k \geq K. \tag{32}$$

Proof. We know already by Theorem 7 that $\|C(x_k)\|_Z \to 0$. After each successful step the trust-region radius is $\geq \delta_{min}$. Unsuccessful steps and a reduction of δ_k occur only in the case $\Delta q_k(s_k^t) \geq (\nu\|C(x_k)\|_Z^2)^\mu$.

In the latter case we deduce (29) and (30) exactly as in the proof of Lemma 4. As in the proof of Lemma 4 we obtain a constant $0 < \delta \leq \delta_c$ with

$$\Delta L_k(s_k) \geq \eta_1 \Delta q_k(s_k^t) \quad \text{for all } k \geq K' \text{ with } \|C(x_k)\|_Z + \delta_k \leq \delta.$$

Now choose $K \geq K'$ so large that $\|C(x_k)\|_Z \leq \delta/2$ for all $k \geq K$. Then all steps $k \geq K$ are accepted as soon as $\delta_k \leq \delta/2$, which yields the lower bound in (32). □

Proof. (of Theorem 8)
We know by Theorem 7 that $\|C(x_k)\|_Z \to 0$. Assume that $\liminf_{k\to\infty} \|\nabla_x L_k\|_{X^*} > 0$. Then we find $K' \geq 0$ with

$$\|\nabla_x L_k\|_{X^*} \geq \varepsilon > 0 \quad \forall\, k \geq K'.$$

Now Lemma 9 yields $K \geq K'$ such that

$$\delta_k \geq \gamma_1 \min\{\delta_{min}, \delta/2\} =: \delta' \quad \text{for all successful iterations } k \geq K. \tag{32}$$

We show next that there is $K'' \geq K$ with

$$\Delta q_k(s_k^t) \geq (\nu \|C(x_k)\|_Z^2)^\mu \quad \text{for all successful iterations } k \geq K''. \tag{33}$$

As in (29) we obtain

$$\|L_u(x_k, \lambda_k)\|_{U^*} \geq \kappa_1 \varepsilon \quad \forall\, k \geq K'$$

and thus by the Cauchy decrease condition (17) and (32)

$$\Delta q_k(s_k^t) \geq \kappa_q \kappa_1 \varepsilon \min\{\kappa_1 \varepsilon, \delta_k\} \geq \kappa_q \kappa_1 \varepsilon \min\{\kappa_1 \varepsilon, \delta'\} =: \varepsilon'$$

for all successful iterations $k \geq K$. Since $\|C(x_k)\|_Z \to 0$ we deduce (33) with $K'' \geq K$ large enough.

Now (33) and the mechanism of the algorithm yield

$$\Delta L_k(s_k) \geq \eta_1 \Delta q_k(s_k^t) \geq \eta_1 \varepsilon' \quad \text{for all successful iterations } k \geq K''.$$

Finally, $\|C(x_k)\|_Z \to 0$ yields

$$L(x_k, \lambda_k) - L(x_{k+1}, \lambda_{k+1}) \geq \Delta L_k(s_k) - \|\lambda_{k+1} - \lambda_k\|_{Z^*} \|C(x_{k+1})\|_Z \geq \frac{\eta_1 \varepsilon'}{2}$$

for all successful iterations $k \geq K'''$ with some $K''' \geq K''$. We deduce that $L(x_k, \lambda_k) \to -\infty$. This is a contradiciton, since $J(x_k, \lambda_k)$ is bounded from below, $\|\lambda_k\|_{Z^*}$ and $\|C(x_k)\|_Z$ are uniformly bounded and thus $L(x_k, \lambda_k)$ is bounded from below. □

5 Application of the generalized SQP-method with parareal solvers

In this section we apply the generalized SQP-method in Algorithm 4.2 to the specific decomposed time-dependent problem (8) for a semilinear parabolic equation in 2D and use parareal solvers for the state equation (9) and the adjoint equation (13).

5.1 An optimal control problem for a semilinear parabolic equation

We consider the problem

$$\min_{y\in W(0,T),u\in L^2((0,T)\times\Omega)} \quad J(y,u) := \frac{1}{2}\int_\Omega (y(T)-y_d)^2\,dx + \frac{\alpha}{2}\int_0^T\int_\Omega u^2\,dx\,dt$$

subject to

$$\begin{aligned} &y_t - \Delta y + \beta y^3 = u \quad \text{on } (0,T\times\Omega,\\ &y|_{(0,T)\times\partial\Omega} = 0,\\ &y(0,\cdot) = v_0 \quad \text{on } \Omega, \end{aligned} \tag{34}$$

where $\beta \geq 0$ is a constant. As in [4] the state and control space are given by

$$\begin{aligned} &U = L^2((0,T)\times\Omega),\\ &Y = W(0,T) = \left\{y \,:\, y\in L^2(0,T;H_0^1(\Omega)),\quad y_t \in L^2(0,T;H^{-1}(\Omega))\right\}. \end{aligned}$$

Since $W(0,T)\hookrightarrow C([0,T];L^2(\Omega))$, it is natural to choose in the decomposed problem (8), (9) the state space $Y_\Delta = B^{N+1}$ with $B = L^2(\Omega)$. Thus

$$C^\Delta : (L^2(\Omega))^{N+1}\times L^2(\Omega\times(0,T)) \to (L^2(\Omega))^{N+1},$$

$$C^\Delta(y_\Delta,u) = \begin{pmatrix} y^0 - v_0\\ y^1 - g(T_0,y^0;u)\\ \vdots\\ y^N - g(T_{N-1},y^{N-1};u) \end{pmatrix}.$$

For the linear case $\beta = 0$ it is easy to check that Assumption 1 is satisfied for all convex open and bounded sets $X_0 = Y_0\times U_0 \subset Y\times U$. For the semilinear case $\beta > 0$ it follows from the theory of semilinear parabolic equations that (34) admits for all $u\in U$ and initial data $v_0 \in W_0^{1,p}$, $p>1$, a unique solution $y\in Y$, see [16, 4]. It is beyond the scope of this paper to analyze (34) in detail. For sufficiently smooth Lipschitz continuous monotone increasing nonlinearities $f(y)$ instead of βy^3 results can be found in [4]. Semilinear elliptic equations are considered in [1, 11].

5.2 Propagators in the parareal scheme

For the implementation of the parareal state solver we use the equidistant time decomposition

$$T_n = n\Delta T, \quad \Delta T = \frac{T}{N}.$$

Coarse propagator

For the coarse propagator $g_\Delta(T_n,v;u)$ we apply one backward Euler step in time and use a standard 5-point stencil for the Laplacian, i.e., $y^{n+1} = g_\Delta(T_n,v;u)$ is given by

$$\frac{y^{n+1}-v}{\Delta T} - Ay^{n+1} + v^3 = u^n.$$

Fine propagator

To approximate the exact propagator $g(T_n, v; u)$ we apply N_f steps of a Crank-Nicholson scheme with time step $\delta T = \Delta T / N_f$ and use a standard 5-point stencil for the Laplacian. Hence, $y_f^{n+1} = g_\delta(T_n, v; u)$ is given by

$$\begin{aligned} &v^0 = v, \\ &\frac{v^{j+1} - v^j}{\delta T} - A\frac{v^{j+1} + v^j}{2} + \frac{(v^{j+1})^3 + (v^j)^3}{2} = u_{n,j}, \quad j = 0, \dots, N_f - 1, \\ &y_f^{n+1} = v^{N_f}. \end{aligned}$$

The implicit equation in each time step is solved by $1 - 3$ Newton iterations in each time step.

5.3 Implementation of the generalized SQP-method

We used the following solvers and approximation of the reduced Hessian

- Nonlinear parareal solver for the state equation starting with the current state, parareal solver for adjoint equation.
- Limited memory BFGS-update for the reduced Hessian. $\hat{B}_0^{-1}$ is chosen as the inverse of the Hessian of the L^2-regularization. The update is skipped if the positive definiteness would be destroyed.
- As approximate solution of (16) we minimize along the Newton step in the trust region if it satisfies a sufficient decrease condition. Otherwise we minimize along the steepest descent direction in the trust region.

Remark 6. *For the solution of the time step equations in the propagators, standard multigrid solvers can directly be used. It would be possible to control the inexactness of the multigrid solvers by the generalized SQP-methods.*

In addition, a coarser space grid could be used for the coarse propagator.

5.4 Numerical results

We consider the following specific problem

$$\begin{aligned} &T = 1, \quad \alpha = 10^{-3}, \quad \beta = 1, \\ &y_d(x) = (2 + \sin(3T + 3x_1 - x_2^2))\, x_1 x_2 (1 - x_1)(1 - x_2), \\ &v_0(x) = (2 + \sin(3x_1 - x_2^2))\, x_1 x_2 (1 - x_1)(1 - x_2). \end{aligned}$$

For the discretization we use the following mesh sizes.

$$N = 20 \ \text{ time domains}, \quad \Delta T = \frac{1}{N}, \quad \delta T = \frac{\Delta T}{40}, \quad 40 \times 40 \text{ equidistant space grid}.$$

For the conditions (22) and (20) we used the constants

$$\kappa_c = 10^{-1}, \quad \kappa_\lambda = 10^{-2}.$$

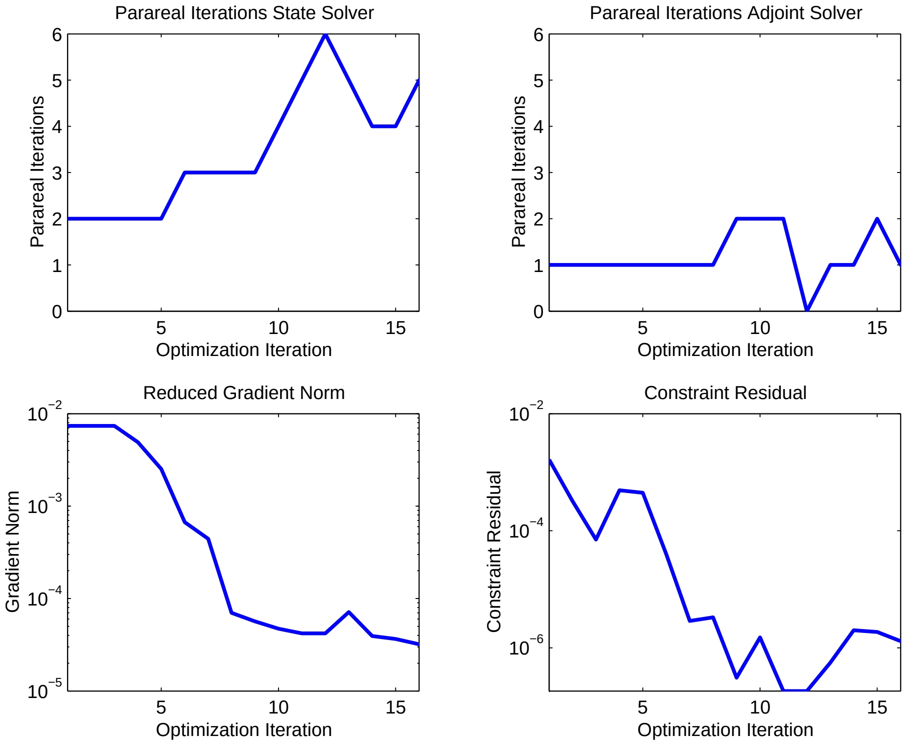

Figure 1. *Iteration history for the generalized SQP-parareal-method*

The iteration history is shown in Figure 1. The generalized SQP-parareal-method takes 16 optimization iterations. The upper plots show the number of parareal iterations for the state equation (upper left) and for the adjoint equation (upper right) versus the optimization iteration. We see that in the mean 3.5 parareal iterations for the state equation and 1.2 parareal iterations for the adjount equation are necessary in each optimization iteration. The lower figures plot the norm of the reduced gradient (lower left) and the constraint residual (lower right) versus the optimization iteration. Note that one state solve requires $5-6$ parareal iterations to achieve a comparable constraint residual of 10^{-6}. This yields the following cost ratio between optimization and one state solve

$$\frac{\text{time(solution of optimization problem)}}{\text{time(solution of state equation)}} = \frac{16 \times (3.5 + 1.2) \text{ parareal its.}}{5 \text{ parareal its.}} \approx 15.$$

Parallel efficiency

With $N = 20$ processors we have

$$\frac{\text{time(parallel parareal it.)}}{\text{time(serial state solve)}} = \frac{\text{20 coarse + 40 fine steps}}{\text{800 fine steps}} = 0.075.$$

Since the optimization with serial state and adjoint solver takes also about 16 iterations, this yields

$$\frac{\text{time(parallel optimization)}}{\text{time(serial optimization)}} \approx 0.075 \times \frac{3.5 + 1.2}{1 + 1} \approx \frac{1}{5.7}.$$

and thus the parallel efficiency

$$\frac{5.7}{20} = 28.5.\%$$

We believe that a comparable parallel efficiency can be obtained for very large problems with many time steps even if many processors are used. Moreover, the parallel optimization allows to treat much larger problems than a serial implementation on one processor, since it automatically distributes the problem to the resources of the available processors.

6 Conclusions

We have proposed a generalized SQP-method for the solution of PDE-constrained optimization problems that works with any user-provided iterative solvers for the state equation and the adjoint equation. The algorithm uses–similar to filter methods [7, 8]–an approach without merit function and is a true all-at-once method that achieves feasibilty and optimality simultaneously. The inaccuracy in the user-provided solvers is controlled by the SQP-method in such a way that global convergence is ensured. While the proposed method is applicable to general problems of the form (1), we have shown that the generalized SQP-method can efficiently be applied to time-dependent PDE-constrained optimization problems by using highly parallel parareal state and adjoint solvers. This approach provides a modular and flexible framework to obtain a parallel optimization solver for large scale transient PDE-constrained optimization problems. The user has only to provide fine and coarse propagators for the considered PDE and its adjoint. We have shown that the approach yields promising numerical results for the optimal control of a semilinear parabolic problem in 2D. The application to other problems, in particular to the optimal control of the Navier-Stokes equations is a topic for future research.

While it is quite straightforward to handle additional control constraints, we plan to extend the approach to problems with state constraints, which is important for applications. Moreover, the parareal technique could also be applied after a time-domain decomposition of the optimality system. We plan to investigate this approach. Finally, the proposed generalized SQP-method is in particular well suited for the optimal control of flows with shocks, since nonlinear solvers for the state equation can directly be used, which avoids the difficulty of linearizing the state equation at shocks. Results for the optimal control of discontinuous flows will be presented elsewhere.

Bibliography

[1] N. Arada, J.-P. Raymond, and F. Tröltzsch, *On an augmented Lagrangian SQP method for a class of optimal control problems in Banach spaces*, Comput. Optim. Appl., 22 (2002), pp. 369–398.

[2] G. Bal, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in Proceedings of DD15, Berlin, Lect. Notes Comput. Sci. Eng., Vol. 40, Springer, Berlin, 2004, pp. 425–432.

[3] G. Bal and Y. Maday, *A 'parareal" time discretization for non-linear PDE's with application to the pricing of an American put*, in Recent developments in domain decomposition methods (Zürich, 2001), Lect. Notes Comput. Sci. Eng., Vol. 23, Springer, Berlin, 2002, pp. 189–202.

[4] M. Bergounioux and F. Tröltzsch, *Optimality conditions and generalized bang-bang principle for a state-constrained semilinear parabolic problem*, Numer. Funct. Anal. Optim., 17 (1996), pp. 517–536.

[5] A. Borzì, *Multigrid methods for parabolic distributed optimal control problems*, J. Comput. Appl. Math., 157 (2003), pp. 365–382.

[6] J. E. Dennis, Jr., M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[7] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM J. Optim., 13 (2002), pp. 635–659.

[8] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[9] M. Heinkenschloss, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, J. Comput. Appl. Math., 173 (2005), 169–198.

[10] M. Heinkenschloss and L. N. Vicente, *Analysis of inexact trust-region SQP algorithms*, SIAM J. Optim., 12 (2001/02), pp. 283–302.

[11] M. Hintermüller and M. Ulbrich, *A mesh-independence result for semismooth newton methods*, Math. Program., 101 (2004), pp. 151–184.

[12] J. E. Lagnese and G. Leugering, *A posteriori error estimates in time-domain decomposition of final value optimal control of the acoustic wave equation*, Appl. Math. Optim., 46 (2002), pp. 263–290.

[13] ——, *Time-domain decomposition of optimal control problems for the wave equation*, Systems Control Lett., 48 (2003), pp. 229–242.

[14] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.

[15] Y. Maday and G. Turinici, *A parareal in time procedure for the control of partial differential equations*, C. R. Math. Acad. Sci. Paris, 335 (2002), pp. 387–392.

[16] P. Neittaanmäki and D. Tiba, *Optimal control of nonlinear parabolic systems*, vol. 179 of Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker Inc., New York, 1994. Theory, algorithms, and applications.

[17] E. O. Omojokun, *Trust region algorithms for optimization with nonlinear equality and inequality constraints*, PhD thesis, University of Colorado, Boulder, Colorado, USA, 1989.

[18] M. Ulbrich and S. Ulbrich, *Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function*, Math. Program., 95 (2003), pp. 103–135. ISMP 2000, Part 3 (Atlanta, GA).

[19] S. Ulbrich, *A sensitivity and adjoint calculus for discontinuous solutions of hyperbolic conservation laws with source terms*, SIAM J. Control Optim., 41 (2002), pp. 740–797.

[20] ——, *Adjoint-based derivative computations for the optimal control of discontinuous solutions of hyperbolic conservation laws*, Systems & Control Letters, 48 (2003), pp. 309–324.