Research Center MATHEON
Mathematics for Key Technologies

An affine covariant composite step method for optimization with PDEs as equality constraints

Lars Lubkoll **Anton Schiela** **Martin Weiser**

Preprint 2015/04

MATHEON **preprint**
http://opus4.kobv.de/opus4-matheon

Preprint 2015/04 **April 2015**

An affine covariant composite step method for optimization with PDEs as equality constraints*

Lars Lubkoll†& Anton Schiela‡& Martin Weiser†

April 16, 2015

Abstract

AMS MSC 2000: 49M37, 90C55, 90C06

We propose a composite step method, designed for equality constrained optimization with partial differential equations. Focus is laid on the construction of a globalization scheme, which is based on cubic regularization of the objective and an affine covariant damped Newton method for feasibility. We show finite termination of the inner loop and fast local convergence of the algorithm. We discuss preconditioning strategies for the iterative solution of the arising linear systems with projected conjugate gradient. Numerical results are shown for optimal control problems subject to a nonlinear heat equation and subject to nonlinear elastic equations arising from an implant design problem in craniofacial surgery.
Keywords: composite step methods, cubic regularization, affine covariant, optimization with PDEs

1 Introduction

Subject of this work is the construction of an algorithm for nonlinear equality constrained optimization with a particular focus on the efficient solution of optimization problems with partial differential equations as constraints. These problems are originally posed in function space and become – after discretization – large scale problems with special structure, inherited from the infinite dimensional setting.

To fix the problem setting, consider a Hilbert space $(X, \langle\cdot,\cdot\rangle)$ and in addition a reflexive Banach space P. In this setting we consider the following optimization problem

$$\min_{x\in X} f(x) \text{ s.t. } c(x) = 0. \tag{1}$$

Here $f : X \to \mathbb{R}$ is a twice continuously Fréchet differentiable functional with suitable smoothness properties. The twice continuously Fréchet differentiable nonlinear operator $c : X \to P^*$ maps into the dual space of P so that it can model a differential equation in weak form:

$$c(x) = 0 \text{ in } P^* \quad \Leftrightarrow \quad c(x)v = 0 \text{ for all } v \in P.$$

*Supported by the DFG Research Center Matheon "Mathematics for key technologies"
†Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany
‡Universität Bayreuth, 94550 Bayreuth, Germany

Here we use that P is a reflexive space, expressed a little sloppily by the relation $P = P^{**}$. In the context of optimal control it is common to split the variable x into two parts $X = Y \times U$ and $x = (y, u)$, where y denotes the *state* and u the *control*. This splitting comes from the special structure of the equality constraints

$$c(x) = A(y) - Bu,$$

where $A : Y \to P^*$ is a nonlinear differential operator with continuous inverse, and $B : U \to P^*$ a linear, compact operator. Under these structural assumptions it is possible to show existence of minimizers and corresponding optimality conditions via the implicit function theorem and the closed range theorem. Often, invertibility of A is used to eliminate the state y from the system and consider the reduced problems $\min_{u \in U} f(y(u), u)$ where $y(u) := A^{-1}(Bu)$.

Our algorithmic approach is that of a *composite step method*. This class of methods is well established in nonlinear optimization and the basis for a couple of competitive optimization codes. Its way to cope with the double aim of feasibility and optimality is to split the full Lagrange-Newton step δx into a *normal step* δn and a *tangential step* δt, and to modify them separately for the purpose of globalization. More precisely, δn is a minimal norm Gauss-Newton step for the solution of the underdetermined problem $c(x) = 0$, and δt aims to minimize f on the current nullspace of the linearized constraints. A couple of variants have been proposed in the literature [13, Sec. 15.4]. Our approach resembles the Vardi approach [35] in the sense that normal steps are computed as damped Newton steps for the underdetermined equation $c(x) = 0$ and thus always satisfy $\nu c(x) + c'(x)\delta n = 0$ for some damping factor $\nu \in]0, 1]$. Compared to the approach of Byrd-Omojokun [28, 7, 6], where normal steps are computed as minimizers of $\|c(x)\|$ in a trust region, Vardi methods need in addition surjectivity of $c'(x)$ as a prerequisite for the computation of steps. This is widely considered as a weakness of this class of methods as a basis for a general purpose solver.

For our purpose, however, a Vardi type method is an appropriate choice for two reasons. First, due to the above described structure of optimal control problems one can usually exclude the case of non-surjective $c'(x)$, so the extra assumption imposed by Vardi type methods is fulfilled.

Second, we avoid the computation of norms of residuals $c(x) \in P^*$. This is important in our context, because the space P^* of residuals $c(x)$ is a dual space, which is often quite irregular and its norm is hard to compute. Rather, our globalization strategy for feasibility relies on the ideas of *affine covariant Newton methods* (which are invariant against affine transformations of the codomain space) for underdetermined problems, as described in [16, Sec. 4.4]. In fact, if $f = \text{const}$, our algorithm reduces to the one proposed there. In this context, a Vardi-like damping strategy is the natural result.

For the tangential step we employ a *cubic regularization method*, as used in [38, 8, 9, 31], and our algorithm reduces to this method in the absence of equality constraints. In total, we solve the following local problems, where $\nu \in]0, 1]$ is an adaptively computed damping factor, $[\omega_c]$ and $[\omega_f]$ are algorithmic parameters, adapted during the iteration, and $\Theta_{\text{aim}} \in]0, 1[$ is a

user provided desired contraction factor:

$$\begin{aligned}\min_{\delta x\in X} f(x)+f'(x)\delta x+\frac{1}{2}L_{xx}(x,p)(\delta x,\delta x)+\frac{[\omega_f]}{6}\|\delta x\|^3\\ \text{s.t. } \nu c(x)+c'(x)\delta x=0,\\ \frac{[\omega_c]}{2}\|\delta x\|\le\Theta_{\text{aim}}.\end{aligned}$$

This step is the basis of a globalization procedure which automatically results in the following algorithmic behavior: far away from a feasible point priority is given to come close to a feasible solution. In this phase the method behaves like a damped Newton method for underdetermined systems. Close to the feasible manifold optimality is stressed, with the restriction that the iterates remain in the Kantorovich neighborhood of contraction around the feasible set. For this we use parametrized models for the nonlinearity of the functional and the constraints. Since our model for the functional is *quadratic*, we use a *cubic* model for the error, while our *linear* model for the constraints is augmented by a *quadratic* model for the error.

The purpose of this paper is to develop a practical algorithm along these ideas and to establish some preliminary theoretical results, such as finite termination of the "inner loop" and fast local convergence. A proof of global convergence is not in the scope of this publication, and will certainly require some modifications of the algorithm. In particular, it is known that affine covariant Newton methods, although very successful in practice, lack a rigorous proof of global convergence, because due to affine covariance the evaluation of $\|c(x)\|$ and thus the usual globalization mechanisms are not available.

The functional analytic framework for our algorithms forces us to distinguish precisely between primal and dual quantities. In particular, we stress the distinction between the linear functional $f'(x)\in X^*$ and the gradient $\nabla f(x)\in X$. Both are connected by the Riesz isomorphism $M: X\to X^*$ which maps $v\in X$ to the linear functional $\langle v,\cdot\rangle\in X^*$. In our context, M is usually a non-trivial mapping. Similarly, we use the derivative $c'(x): X\to P^*$, instead of $\nabla c(x)$, which is widely seen in the literature, but not useful in a functional analytic setting. Concerning adjoint mappings, we use Banach space adjoints, throughout, i.e., $c'(x)^*: P\to X^*$ is defined by $(c'(x)^*p)(v)=pc'(x)v=p(c'(x)v)$.

Let us quickly comment on related, existing approaches of equality constrained optimization in the context of optimal control. Composite step trust-region methods of Byrd-Omojokun type have been considered in [23, 29], where focus was laid on inexact iterative solution techniques of the arising linear systems, cf. also the earlier work [24], while similar issues were considered in [5, 14] in a line-search context. In [12, 40] an algorithm is discussed that integrates adaptive mesh refinement into a composite step trust-region method. An alternative invariance concept has been used in [36] for local inexact SQP methods.

2 Lagrange multipliers and normal steps

Let us consider the generic equality constrained optimization problem (1), on the Hilbert space X. Under standard assumptions, we can derive its KKT conditions at a stationary point x_*. To be accurate, we require that f and c are continuously Fréchet differentiable at x_* and $c'(x_*): X\to P^*$ is surjective. Under these conditions there exists a Lagrange

multiplier $p \in P^{**} \cong P$ (recall that P is reflexive), such that

$$f'(x_*)v + pc'(x_*)v = 0 \qquad \forall v \in X \tag{2}$$

$$c(x_*) = 0. \tag{3}$$

Here (2) expresses the stationarity condition in $\ker c'(x_*)$:

$$(2) \;\Leftrightarrow\; f'(x_*) \in \operatorname{ran} c'(x_*)^* \;\Leftrightarrow\; f'(x_*)v = 0 \quad \forall v \in \ker c'(x_*). \tag{4}$$

The last equivalence is due to the *closed range theorem* (cf. e.g. [4, Thm. 2.19]). Thus, to show (2) it is sufficient to verify $f'(x_*)v = 0\, \forall v \in \ker c'(x_*)$. This can be done via *Ljusternik's theorem* (cf. e.g. [26, Sec. 0.2.4]), a variant of the implicit function theorem.

As X is a Hilbert space, equipped with scalar product $\langle \cdot, \cdot \rangle$, we can perform the splitting

$$X = \ker c'(x_*) \oplus \ker c'(x_*)^\perp$$

of X into $\ker c'(x_*)$ and its orthogonal complement $\ker c'(x_*)^\perp$. Application of this splitting to (2) then yields the equivalence

$$(f'(x_*) + pc'(x_*))v = 0 \quad \forall v \in X \quad \Leftrightarrow \quad \begin{cases} f'(x_*)v = 0 \quad \forall v \in \ker c'(x_*) \\ \\ (f'(x_*) + pc'(x_*))w = 0 \quad \forall w \in \ker c'(x_*)^\perp. \end{cases}$$

The first condition on the right hand side characterizes stationarity of x_* and neither depends on p, nor the scalar product. In contrast, the second condition

$$f'(x_*)w + pc'(x_*)w = 0 \quad \forall w \in \ker c'(x_*)^\perp, \tag{5}$$

depends on the scalar product $\langle \cdot, \cdot \rangle$ and involves p. We will see that the validity of (5) has nothing to do with the stationarity of x_*.

Rather, (5) holds for *arbitrary* $x \in X$ as long as $c'(x)$ is surjective, and the corresponding Lagrange multiplier p_x can be computed by solving a linear system, where the Riesz isomorphism $M : X \to X^*$ (characterized by $(Mv)(w) = \langle v, w \rangle$) enters:

$$\begin{pmatrix} M & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} v \\ p_x \end{pmatrix} + \begin{pmatrix} f'(x) \\ 0 \end{pmatrix} = 0. \tag{6}$$

Theorem 2.1. *For $x \in X$ assume that $c'(x) : X \to P^*$ is bounded and surjective. Then there is a unique element $p_x \in P$ that solves* (6) *and satisfies*

$$f'(x)w + p_x c'(x)w = 0 \quad \forall w \in \ker c'(x)^\perp. \tag{7}$$

Proof. It is well known that block operators of the form encountered in (6) are continuously invertible (in a Hilbert space context), as long as $c'(x)$ is bounded and surjective and the symmetric bilinear form $\langle v, w \rangle = (Mv)(w)$ is elliptic on $\ker c'(x)$ and continuous. This is the result of the famous Brezzi splitting theorem (cf. e.g. [3, Thm. 4.3]).

Now we test the first row of (6) with $w \in \ker c'(x)^\perp$.

$$(Mv)(w) + p_x c'(x)w + f'(x)w = 0.$$

Since $w \in \ker c'(x)^\perp$ and $v \in \ker c'(x)$ by the second row of (6) we conclude $(Mv)(w) = 0$ and thus (7). □

Definition 2.2. We call the element p_x in Theorem 2.1 the Lagrange multiplier of problem (1) at x (w.r.t. the scalar product $\langle \cdot, \cdot \rangle$).

We will see in the following section that our special Lagrange multiplier enjoys a couple of very favorable properties, also far away from an optimal solution.

Remark 2.3. In the literature a Lagrangian multiplier that is computed via (6) is known as a "least-squares estimate for p", in the context of the standard scalar product of $\mathbb{R}^n$: $\langle v, w \rangle_2 := v^T w$. The motivation is that p minimizes $\|f'(x)^T + c'(x)^T p\|_2$. It seems, however, that (7), which turns out to be very helpful in the context of our algorithm, is not widely known.

Lemma 2.4. *Let $x_0 \in X$ and assume that f' and c' depend continuously on x. Further, assume that $c'(x_0) : X \to P^*$ is a bounded, surjective linear operator. Then the Lagrange multiplier p_x at x is given as a continuous implicit function of x in some neighborhood around x_0.*

Proof. We apply the implicit function theorem to (6), which is of the form $F(x, p) = K(x)p + r(x) = 0$. In this context, x is the parameter, and $p_x = p(x)$ is the desired implicit function. We observe that F is linear and thus differentiable in p and that $\partial / \partial p \, F(x, p) = K(x)$ is continuously invertible at x_0 and depends continuously on x by our assumptions. Thus, we can apply the implicit function theorem (cf. e.g. [39, Thm 4.B]) to get the desired result. □

Lagrangian function. Let us discuss our result in terms of the Lagrangian function

$$L(x, p) := f(x) + pc(x),$$

where $p = p_x$ is chosen as in Theorem 2.1. In this context our result implies that normal steps $\delta n \in \ker c'(x)^\perp$ do not change the Lagrangian function up to first order:

$$L_x(x, p_x)\delta n = f'(x)\delta n + p_x c'(x)\delta n = 0 \quad \forall \delta n \in \ker c'(x)^\perp.$$

Thus, our p_x makes $L(\cdot, p_x)$ stationary in $\ker c'(x)^\perp$. In contrast, for tangential steps δt, which are contained in $\ker c'(x)$, the relevant relation is:

$$L_x(x, p_x)\delta t = f'(x)\delta t + p_x c'(x)\delta t = f'(x)\delta t \quad \forall \delta t \in \ker c'(x).$$

Thus, their contribution is, up to first order, independent of the choice of p. Taken together, this yields for the composite step $\delta x = \delta n + \delta t$:

$$L_x(x, p_x)\delta x = L_x(x, p_x)(\delta n + \delta t) = f'(x)\delta t$$

If we look at a second order approximation of L along δx we obtain

$$L(x + \delta x, p_x) = L(x, p_x) + f'(x)\delta t + \frac{1}{2} L_{xx}(x, p_x)(\delta x)^2 + o(\|\delta x\|^2).$$

Hence, p_x only enters in the second order approximation of L. In Section 3 below we will construct a similar second order model for f, which avoids the well known Maratos effect.

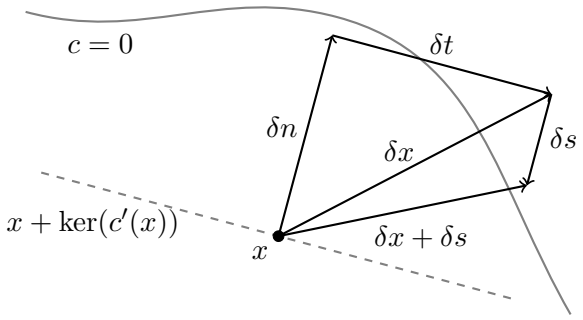


Figure 1: Sketch of a composite step.

3 Composite steps and their consistency

In this section we will discuss the properties of composite steps and in particular their order of consistency, i.e. the asymptotic behavior of the difference between quadratic models and the actual problem. Classically, composite steps are composed from a normal step δn and a tangential step δt. In our framework we add an additional *simplified normal step* δs that also plays the role of a *second order correction*.

For this purpose we introduce the following notation, which refers to a single step of our algorithm. Consider a fixed iterate x with associated Lagrange multiplier p_x computed as in Theorem 2.1. We denote the (damped) normal step by $\delta n \in \ker c'(x)^\perp$ and the tangential step by $\delta t \in \ker c'(x)$. The undamped normal step is denoted by Δn, so that $\delta n = \nu \Delta n$, where $\nu \in]0, 1]$ is the damping factor. A similar notation is conceivable for tangential steps, even though the computation of their direction and length is usually performed in one step.

Finally we call the simplified normal step, to be defined below, $\delta s \in \ker c'(x)^\perp$. Then the ordinary composite step is given by

$$\delta x := \delta n + \delta t, \tag{8}$$

but we will also consider the *extended* composite step defined as $\delta x + \delta s$.

The above steps have to fulfill the following equations (but are, of course not fully determined by them, since in general $\ker c'(x)$ is non-trivial):

$$c(x) + c'(x)\Delta n = 0 \quad \text{undamped normal step} \tag{9}$$

$$c'(x)\delta t = 0 \quad \text{tangential step} \tag{10}$$

$$(c(x + \delta x) - c(x) - c'(x)\delta x) + c'(x)\delta s = 0 \quad \text{simplified normal step.} \tag{11}$$

To fully determine Δn and δs, we use the scalar product $\langle \cdot, \cdot \rangle$ on the Hilbert space X and require

$$\Delta n, \delta s \in \ker c'(x)^\perp.$$

The tangential step will be determined by approximately minimizing a quadratic model of L on $\ker c'(x)$, which corresponds to a quadratic model of f on the feasible set $c(x) = 0$.

3.1 Computation of steps via saddle point problems

In this subsection we specify the conditions that determine the normal steps Δn, the Lagrange multiplier p_x, tangential steps δt, and the simplified normal step δs. All quantities are computed as solutions of certain saddle point problems.

3.1.1 Normal step

Since Δn and δs are both supposed to lie in $\ker c'(x)^\perp$ we start with some general discussion. First we note that the minimal norm problem

$$\min_{w \in X} \ \frac{1}{2}\langle w, w\rangle \text{ s.t. } c'(x)w + g = 0, \tag{12}$$

is equivalent to finding $w \in \ker c'(x)^\perp$ such that $c'(x)w + g = 0$. The optimality conditions for (12) motivate the following Lemma:

Lemma 3.1. *Suppose that $w \in X$ and $q \in P$ satisfy*

$$\begin{pmatrix} M & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} w \\ q \end{pmatrix} + \begin{pmatrix} 0 \\ g \end{pmatrix} = 0 \tag{13}$$

for some $g \in P^*$. Then $w \in \ker c'(x)^\perp$.

Proof. This follows from the first row of (13) and $(Mw)(\xi) = \langle w, \xi\rangle$:

$$(Mw)(\xi) + q(c'(x)\xi) = 0 \ \forall \xi \in X \Rightarrow (Mw)(\xi) = 0 \ \forall \xi \in \ker c'(x) \Leftrightarrow w \in \ker c'(x)^\perp$$

□

We stress again at this point that the choice of the Hilbert space scalar product $\langle \cdot, \cdot \rangle$ is crucial and depends on the function space context of the problem. Consequently, M, the Riesz-isomorphism of X, is usually a non-trivial linear operator. Further, we note that the normal step does not depend on the Lagrange multiplier p_x.

We denote the solution of (13) as

$$w = -c'(x)^- g. \tag{14}$$

With this notation, we can define the normal step via:

$$\Delta n := -c'(x)^- c(x)$$

as the solution of (13) with $g = c(x)$.

3.1.2 Lagrange multiplier

We have already discussed the role of p_x and that it can be computed via (6) in Section 2. However, instead of computing p_x via (6), we obtain it via a correction δp to the previous multiplier p_-, i.e. $p_x = p_- + \delta p$. Recalling that $L_x(x, p_-) = f'(x) + c'(x)^* p_-$ this is achieved by

$$\begin{pmatrix} M & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} w \\ \delta p \end{pmatrix} + \begin{pmatrix} L_x(x, p_-) \\ 0 \end{pmatrix} = 0.$$

This formulation has the advantage that its right hand side tends to 0 when x tends to a local minimizer, which in turn improves numerical stability with respect to rounding errors or truncated iterations in the system solution. In exact arithmetic both alternatives yield, of course, the same result p_x, which therefore only depends on x, but not on previous Lagrange multiplier estimates.

3.1.3 Tangential step

Once we have computed the normal step Δn, a damping factor ν, so $\delta n = \nu \Delta n$, and an adjoint state p_x, we want to compute the tangential step $\delta t \in \ker c'(x)$.

Ignoring for the moment issues of globalization, which are discussed in Section 4.3, this is done such that $\delta x := \delta n + \delta t$ is an approximation of the minimizer of the quadratic model

$$q(\delta x) := f(x) + f'(x)(\delta x) + \frac{1}{2} L_{xx}(x, p_x)(\delta x)^2, \tag{15}$$

of L on $\ker c'(x)$, provided such a minimizer exists. In this case, we call this exact minimizer Δt. Otherwise, δt should at least be a direction of descent. Later in our globalization scheme we will add some modifications to this functional (cf. (42) below).

Thus, the quadratic problem we have to solve is

$$\min_{\delta t} \; q(\delta n + \delta t) \quad \text{subject to} \quad c'(x)\delta t = 0. \tag{16}$$

Omitting terms that are independent of δt and adding the term $p_x c'(x)\delta t = 0$ to the functional, this is equivalent to

$$\min_{\delta t} \; \Big(L_x(x, p_x) + L_{xx}(x, p_x)\delta n \Big)\delta t + \frac{1}{2} L_{xx}(x, p_x)(\delta t)^2 \tag{17a}$$

subject to

$$c'(x)\delta t = 0. \tag{17b}$$

This formulation, which only depends on the Lagrange function and its derivatives, reduces the influence of rounding errors close to the optimal solution, since $L_x(x, p) \to 0$ for $(x, p_x) \to (x_*, p_{x_*})$.

The definition of the tangential step in this way is closely related to the Lagrange-Newton step. In the vicinity of a solution satisfying the sufficient second order conditions, i.e. when $\nu = 1$ and L_{xx} is positive definite (elliptic) on $\ker c'(x)$, then the exact minimizer Δt of problem (17) exists, and the corresponding first order optimality conditions are

$$\begin{pmatrix} L_{xx}(x, p_x) & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta t \\ \Delta p \end{pmatrix} + \begin{pmatrix} L_x(x, p_x) + L_{xx}(x, p_x)\delta n \\ 0 \end{pmatrix} = 0. \tag{18}$$

We observe that $(\Delta x, \Delta p) = (\Delta n + \Delta t, \Delta p)$ is a full Lagrange-Newton step:

$$\begin{pmatrix} L_{xx}(x, p_x) & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta p \end{pmatrix} + \begin{pmatrix} L_x(x, p_x) \\ c(x) \end{pmatrix} = 0. \tag{19}$$

In this case Δp would be a Newton update for the Lagrange multiplier or, as a different interpretation, the Lagrange multiplier at x with respect to the scalar product induced by $L_{xx}(x, p_x)$.

3.1.4 Simplified normal step

For a given δx we can compute the simplified normal step via a saddle point problem of the form (13), such that

$$\delta s := -c'(x)^- \left(c(x+\delta x) - c(x) - c'(x)\delta x\right). \tag{20}$$

It follows from Lemma 3.1 that $\delta s \in \ker c'(x)^\perp$, and thus

$$L_x(x,p_x)\delta s = \left(f'(x) + p_x c'(x)\right)\delta s = 0. \tag{21}$$

If $\delta x = \delta n + \delta t$ is computed as in (9) and (10) and thus satisfies $c'(x)\delta x + \nu c(x) = 0$ we can derive an alternative representation of the simplified normal step

$$\delta s := -c'(x)^- \left(c(x+\delta x) - (1-\nu)c(x)\right).$$

In the undamped case $\nu = 1$ this relation reduces to $\delta s = -c'(x)^- c(x+\delta x)$, which is the second step of a simplified Newton method for the equation $c(x) = 0$, starting at x, and explains our naming of δs. We will see in Lemma 3.2 below that δs also plays the role of a second order correction.

3.2 Order of consistency of composite steps

A basic principle of equality constrained SQP is to minimize a quadratic model of the functional subject to a linear model of the constraints. In this section we will study the order of consistency of these models, i.e., the order in which our local models approximate the true problem, close to an iterate. This will be the theoretical basis for the construction of our algorithm. Recalling that f and c are assumed to be twice Fréchet differentiable at x, the following quadratic model is used for the functional:

$$\begin{aligned} q(\delta x) &:= f(x) + f'(x)\delta x + \frac{1}{2}L_{xx}(x,p_x)(\delta x)^2 \\ &= f(x) + f'(x)\delta x + \frac{1}{2}\left(f''(x) + p_x c''(x)\right)(\delta x)^2. \end{aligned} \tag{22}$$

The last term, involving $c''(x)$ takes into account second order information of the equality constraints, which is necessary to achieve fast local convergence of the undamped Lagrange-Newton method. We will show that $q(\delta x)$ is second order consistent with $f(x+\delta x+\delta s)$, but only first order consistent with $f(x+\delta x)$. The latter is the reason for the well known Maratos effect, while the first result yields a possible remedy. Therefore we refer to the simplified normal step also as a *second order correction*.

For the results in this section, δx need not necessarily be defined as a "composite step", but can be an arbitrary (small) perturbation of our iterate x. However, δs is defined by (20).

Lemma 3.2. *Denote by δx an arbitrary perturbation of x and by δs the corresponding simplified normal step, determined through* (20)*. Then we have the following consistency results:*

$$\|\delta s\| = o(\|\delta x\|), \tag{23}$$

$$f(x+\delta x) = q(\delta x) + O(\|\delta x\|^2), \tag{24}$$

$$f(x+\delta x+\delta s) = q(\delta x) + o(\|\delta x\|^2). \tag{25}$$

Proof. Estimate (23) follows directly from the definition (20) of δs, using differentiability of c and continuous invertibility of $c'(x)$ on $\ker c'(x)^\perp$. Next, (24) directly follows from comparing the Taylor expansion for f at x with $q(\delta x)$:

$$\begin{aligned} q(\delta x) - f(x+\delta x) &= q(\delta x) - \Big(f(x) + f'(x)\delta x + \frac{1}{2} f''(x)(\delta x)^2 + o(\|\delta x\|^2) \Big) \\ &= \frac{1}{2} p_x c''(x)(\delta x)^2 + o(\|\delta x\|^2) = O(\|\delta x\|^2). \end{aligned}$$

Testing the defining equation (11) for δs with p_x, and by Taylor expansion of c at x in direction δx we compute

$$\begin{aligned} 0 &= p_x \Big([c(x+\delta x) - c(x) - c'(x)\delta x] + c'(x)\delta s \Big) \\ &= p_x \Big([c(x) + c'(x)\delta x + \frac{1}{2} c''(x)(\delta x)^2 + o(\|\delta x\|^2) - c(x) - c'(x)\delta x] + c'(x)\delta s \Big) \\ &= p_x \Big(\frac{1}{2} c''(x)(\delta x)^2 + c'(x)\delta s \Big) + o(\|\delta x\|^2). \end{aligned}$$

Using this and (21) we obtain

$$f'(x)\delta s = -p_x c'(x)\delta s = \frac{1}{2} p_x c''(x)(\delta x)^2 + o(\|\delta x\|^2)$$

and from (22)

$$q(\delta x) = f(x) + f'(x)\delta x + \frac{1}{2} f''(x)(\delta x)^2 + f'(x)\delta s + o(\|\delta x\|^2). \tag{26}$$

Subtracting (26) from the Taylor expansion for f at x in direction $\delta x + \delta s$ we compute

$$\begin{aligned} & f(x+\delta x+\delta s) - q(\delta x) \\ &= f(x) + f'(x)(\delta x + \delta s) + \frac{1}{2} f''(x)(\delta x + \delta s)^2 + o(\|\delta x + \delta s\|^2) - q(\delta x) \\ &= \frac{1}{2} f''(x)(\delta s, 2\delta x + \delta s) + o(\|\delta x + \delta s\|^2) + o(\|\delta x\|^2). \end{aligned}$$

Now (23) implies $f''(x)(\delta s, 2\delta x + \delta s) = o(\|\delta x\|^2)$ and in turn the desired result (25). □

In our interpretation, q is not a quadratic model of f on the linearization $c'(x)\delta t = 0$ of the feasible set. Rather it takes into account a better, quadratic, approximation of the true feasible set. Thus, to compare q and f, we should not evaluate f at $x + \delta x$, but at a point that is closer to the true feasible set, e.g. at the second order corrected point $x + \delta x + \delta s$. This issue, which is of course well known, manifests in the Maratos-effect and its elimination via second order corrections.

Quantitative estimates. After these qualitative considerations we discuss conditions under which the above qualitative estimates can be quantified more explicitly. Our considerations are based mainly on affine covariant Lipschitz conditions on L_{xx}, f' and c'. These estimates provide the main motivation for a couple of algorithmic choices in the following section, and they will be the basis for finite termination and local fast convergence results for our algorithm, provided below.

Recall that $v = c'(x)^- r$ denotes the least norm solution of the problem $c'(x)v = r$.

Lemma 3.3. *For any given x and δx, and corresponding simplified normal step δs the following identities hold:*

$$\begin{aligned} f(x+\delta x+\delta s)-q(\delta x) &= T_1+T_2 \quad \textit{where} \\ T_1 &:= L(x+\delta x,p_x)-L(x,p_x)-L_x(x,p_x)\delta x-\frac{1}{2}L_{xx}(x,p_x)(\delta x,\delta x) \\ &= \int_0^1 \left(L_x(x+\sigma\delta x,p_x)-L_x(x,p_x)-L_{xx}(x,p_x)\sigma\delta x\right)\delta x\, d\sigma \\ &= \int_0^1\int_0^1 \left(L_{xx}(x+\tau\sigma\delta x,p_x)-L_{xx}(x,p_x)\right)(\sigma\delta x,\delta x)\, d\tau\, d\sigma \\ T_2 &:= f(x+\delta x+\delta s)-f(x+\delta x)-f'(x)\delta s \\ &= \int_0^1 \left(f'(x+\delta x+\sigma\delta s)-f'(x)\right)\delta s\, d\sigma. \end{aligned}$$

Furthermore we have

$$\delta s=\int_0^1 c'(x)^-(c'(x+\sigma\delta x)-c'(x))\delta x\, d\sigma.$$

Proof. The identities for T_1 and T_2 follow from the fundamental theorem of calculus. So it remains to show

$$f(x+\delta x+\delta s)-q(\delta x)=T_1+T_2$$

Indeed, using the identities $-c'(x)\delta s=c(x+\delta x)-c(x)-c'(x)\delta x$, and $(f'(x)+p_xc'(x))\delta s=0$ we compute

$$\begin{aligned} T_1+q(\delta x) &= L(x+\delta x,p_x)-L(x,p_x)-L_x(x,p_x)\delta x-\frac{1}{2}L_{xx}(x,p_x)(\delta x,\delta x)+q(\delta x) \\ &= f(x+\delta x)+(p_xc(x+\delta x)-p_xc(x)-p_xc'(x)\delta x)=f(x+\delta x)-p_xc'(x)\delta s \\ &= f(x+\delta x)+f'(x)\delta s=f(x+\delta x+\delta s)-T_2. \end{aligned}$$

The result on δs follows similarly from the fundamental theorem of calculus. □

Theorem 3.4. *Assume that there are constants ω_c, $\omega_{f'}$, and ω_L, such that*

$$\|c'(x)^-(c'(x+v)-c'(x))v\|\le\omega_c\|v\|^2, \tag{27}$$

$$|(L_{xx}(x+v,p_x)-L_{xx}(x,p_x))(v,v)|\le\omega_L\|v\|^3, \tag{28}$$

$$|(f'(x+v)-f'(x))w|\le\omega_{f'}\|v\|\|w\|, \tag{29}$$

where (x,p_x) are taken among the iterates, and v,w arbitrary. Then for arbitrary δx and corresponding simplified normal steps δs we have the estimates:

$$\|\delta s\|\le\frac{\omega_c}{2}\|\delta x\|^2, \tag{30}$$

$$\begin{aligned} |f(x+\delta x+\delta s)-q(\delta x)| &\le \frac{\omega_L}{6}\|\delta x\|^3+\omega_{f'}\|\delta s\|\left(\|\delta x\|+\frac{1}{2}\|\delta s\|\right) \\ &\le \left(\frac{\omega_L}{6}+\frac{\omega_{f'}\omega_c}{2}\left(1+\frac{\omega_c}{4}\|\delta x\|\right)\right)\|\delta x\|^3. \end{aligned} \tag{31}$$

Proof. First note that (setting $v = \sigma\delta x$) we see

$$\|\delta s\| \leq \int_0^1 \frac{1}{\sigma}\|c'(x)^-(c'(x+\sigma\delta x) - c'(x))\sigma\delta x\|\, d\sigma \leq \omega_c\|\delta x\|^2 \int_0^1 \sigma\, d\sigma \leq \frac{\omega_c}{2}\|\delta x\|^2.$$

With respect to the Lipschitz constant for L_{xx} we get with Lemma 3.3

$$|f(x+\delta x+\delta s) - q(\delta x)| \leq |T_1| + |T_2|.$$

Then with the assumed affine covariant Lipschitz conditions (setting $v = \tau\sigma\delta x$) we get

$$\begin{aligned}|T_1| &\leq \int_0^1 \int_0^1 \frac{1}{\tau^2\sigma}\left|(L_{xx}(x+\tau\sigma\delta x, p_x) - L_{xx}(x,p_x))\,(\tau\sigma\delta x, \tau\sigma\delta x)\right| d\tau\, d\sigma \\ &\leq \omega_L\|\delta x\|^3 \int_0^1 \int_0^1 \tau\sigma^2\, d\tau\, d\sigma = \frac{\omega_L}{6}\|\delta x\|^3\end{aligned}$$

and (setting $v = \delta x + \sigma\delta s$, $w = \delta s$):

$$\begin{aligned}|T_2| &\leq \int_0^1 \left|\left(f'(x+\delta x+\sigma\delta s) - f'(x)\right)\delta s\right| d\sigma \leq \omega_{f'}\|\delta s\| \int_0^1 \|\delta x + \sigma\delta s\|\, d\sigma \\ &\leq \omega_{f'}\|\delta s\|\left(\|\delta x\| + \int_0^1 \sigma\, d\sigma\|\delta s\|\right) = \omega_{f'}\|\delta s\|\left(\|\delta x\| + \frac{1}{2}\|\delta s\|\right)\end{aligned}$$

Adding both estimates yields the first part of (31), inserting (30) the second part. □

4 The globalization scheme

The globalization mechanism is a central part of any algorithm for nonlinear problems. The particular difficulty in equality constrained optimization is the simultaneous achievement of the potentially conflicting aims of feasibility and optimality. As the determination of the feasible region is the prerequisite for finding an optimal solution, priority is attributed to feasibility. However, an algorithm that stresses feasibility too much is likely to be inefficient in finding an optimal point or may even converge to a non-stationary feasible point. Thus, the main difficulty is to weigh both aims properly. Roughly speaking an ideal algorithm should work as follows: *far away* from the feasible region, focus on getting close to it, *close to* the feasible region, focus on optimality without neglecting feasibility. However, to render this vague idea useful we first have to quantify, what *close* should mean.

A popular approach to do this is to say: "close to the feasible set means that $\|c(x)\|$ is small". Two popular globalization techniques arise from that statement, namely merit functions and filter methods [18]. Both combine monotonicity requirements on $f(x)$ and $\|c(x)\|$ to achieve $\|c(x)\| \to 0$ while minimizing f. However, this approach is in conflict with our algorithmic paradigm that residual norms must not enter the algorithm.

Thus we resort to a different idea, which originates from *affine covariant Newton methods* [16]. In the context of Newton's method (or simplified Newton) for nonlinear equations one can argue that *close* to the solution means *safely within the region of local convergence*, so that we can find a feasible point easily within a few steps of Newton's method. Carrying over this idea to equality constraints in nonlinear optimization we can say that a point $x \in X$ is considered close to the feasible set, if a sequence of pure *normal steps* started at x would converge quickly to a feasible point.

To transform this idea into an algorithm, we have to *quantify* this region, at least by a heuristic estimate of the relevant Lipschitz constant. Ways to construct such estimates are among the central topics in [16]. Here we focus on equality constrained optimization problems and refer to [16, Section 3.1] for an in depth treatment for the case of other nonlinear systems of equations.

In broad terms, this leads to a predictor-corrector algorithm outlined in Alg. 1. Based on the values of two algorithmic parameters $[\omega_c]$ and $[\omega_f]$, which can be interpreted as a-posteriori estimates of the corresponding Lipschitz constants ω_c and ω_L from Theorem 3.4, the normal step δn, the tangential step δt, and the simplified normal step δs are computed. If the results agree with the theoretical expectations, the step is accepted. Otherwise, $[\omega_c]$ and $[\omega_f]$ are adjusted according to the newly acquired information.

Algorithm 1 Outer and inner loop, inner loop strongly simplified

Require: initial iterate x, $[\omega_c], [\omega_f]$
  repeat// *NLP loop*
    repeat// *step computation loop*
      compute new trial correction δx, via (42)
      compute simplified correction δs, via (20)
      compute new Lipschitz constants $[\omega_c], [\omega_f]$
    until trial correction δx accepted
    $x \leftarrow x + \delta x + \delta s$
  until converged

In the remainder of this section, we will fill out the details of the algorithm: how to compute δx, how to update $[\omega_c]$ and $[\omega_f]$, and when to accept a trial step.

4.1 Models for non-linearities

As described, our algorithm, an SQP-method, applies linear models for the equality constraints and quadratic models for the functional. We describe the deviation of these linear and quadratic models from the true problem by parametrized quadratic and cubic error models, respectively. This approach is motivated qualitatively by Lemma 3.2 and quantitatively by Theorem 3.4. Adjusting the parameters of these error models appropriately yields a globalization scheme for our SQP iteration.

Newton contraction. Let us first recall the principal ideas of the affine covariant damping strategy for nonlinear systems [16], and then describe our modification for composite step methods. The situation is depicted in Figure 2.

Our main tool is the use of simplified Newton steps that we have encountered already in the last section, namely δs. In fact, if $\nu = 1$, i.e. $\delta n = \Delta n$, then

$$c(x) + c'(x)\delta x = 0,$$

and δs satisfies the equation

$$c(x + \delta x) + c'(x)\delta s = 0.$$

Thus, δx and δs can be interpreted as the first two steps of a simplified Newton method for the problem: find ξ such that $c(\xi) = 0$, starting at x. Thus, if $\|\delta s\| \ll \|\delta x\|$ holds, we expect

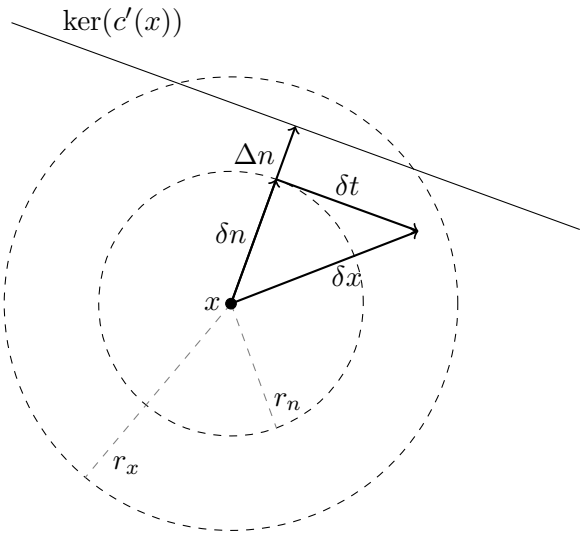


Figure 2: Sketch of a composite step.

fast local convergence to a feasible point. So if we denote the contraction factor

$$\Theta(\delta x) := \frac{\|\delta s\|}{\|\delta x\|},$$

then, $\Theta(\delta x) \ll 1$ is a good indication that Newton contraction takes place, so that δx is an acceptable correction.

In general, if $\nu \leq 1$, then δx and δs satisfy:

$$\begin{aligned} c(x) + c'(x)\delta x &= (1-\nu)c(x), \\ c(x+\delta x) + c'(x)\delta s &= (1-\nu)c(x), \end{aligned}$$

and thus, they form two steps of a simplified Newton method for the relaxed problem:

$$\text{find } \xi_\nu, \text{ such that } c(\xi_\nu) = (1-\nu)c(x). \tag{32}$$

As before, $\|\delta s\| \ll \|\delta x\|$ indicates fast local convergence of Newton's method towards the solution ξ_ν of the relaxed problem. If $\nu \ll 1$, we expect that (32) is much easier to solve than the problem $c(\xi) = 0$. In fact, by the implicit function theorem the solutions ξ_ν of these relaxed problems locally define a path, the so called Newton path [16, Sec. 3.1.4], or – in the context of underdetermined equations – the geodetic Gauss-Newton path [16, Sec. 4.4.2].

These considerations lead to the following concretization of our rough algorithmic idea, described above. We compute the step δx, such that $\Theta(\delta x) < \Theta_{\text{aim}}$ is to be expected (section 4.3), and accept δx, if after computation of δs, $\Theta(\delta x) < \Theta_{\text{acc}}$ is actually observed (section 4.2), where $0 < \Theta_{\text{aim}} < \Theta_{\text{acc}} < 1$ are user defined parameters.

Under the conditions of Theorem 3.4 we conclude the estimate

$$\Theta(\delta x) \leq \frac{\omega_c}{2}\|\delta x\|,$$

which is the basis for our model (39), introduced below, in which the theoretical quantity ω_c is replaced by a computational estimate $[\omega_c]$.

Cubic regularization. Similarly, Theorem 3.4 yields (at least for bounded $\|\delta x\|$) the cubic bound (31) for the difference $f(x+\delta x+\delta s)-q(\delta x)$. This motivates the introduction of a cubic regularization of the quadratic model (15), equipped with an algorithmic quantity $[\omega_f]$

$$\begin{aligned} m_{[\omega_f]}(\delta x) &= q(\delta x) + \frac{[\omega_f]}{6}\|\delta x\|^3 \\ &= f(x) + f'(x)\delta x + \frac{1}{2}L_{xx}(x,p)(\delta x)^2 + \frac{[\omega_f]}{6}\|\delta x\|^3. \end{aligned} \tag{33}$$

The quantity $[\omega_f]$ can be interpreted as an affine covariant estimate of the prefactor on the right hand side of (31).

4.2 Acceptance criteria

Next we describe how it is decided, whether a computed step is accepted or rejected. Our criteria are Newton contraction (for feasibility) and functional decrease (for optimality).

Acceptable feasibility contraction. The above considerations motivate the following choice of acceptance criterion for a trial iterate δx:

$$\Theta(\delta x) := \frac{\|\delta s\|}{\|\delta x\|} \leq \Theta_{\text{acc}} < 1. \tag{34}$$

It indicates, whether the above simplified Newton iteration for the relaxed feasibility problem (32) is likely to converge. Lemma 3.2 asserts that an acceptable iterate is found for sufficiently small ν. Practical choices for Θ_{acc} are in the range $\Theta_{\text{acc}} \in [0.25, 0.75]$.

Acceptable decrease. While normal steps aim at feasibility and thus a criterion measuring the deviation from the constraint has been introduced, tangential steps are responsible for decrease in the cost functional. Therefore we need a criterion similar to the case of unconstrained optimization that ensures decrease of the cost functional.

However, constraints introduce additional difficulties. First, we have to take into account the fact that the normal step may yield increase in the cost functional. In general, finding a feasible point may require an increase of the objective, relative to the current *infeasible* iterate. Thus, we cannot require decrease in the total step and decrease should only be measured for the tangential step. Thus, at first sight, a decrease condition of the form

$$f(x+\delta n+\delta t) < f(x+\delta n) \tag{35}$$

would seem useful.

This leads us to the second difficulty, which arises most likely, if acceptable normal steps are large, relative to the nonlinearity of the functional. Recall that tangential steps are computed with the help of a quadratic model based at the current iterate x, but they are added to the normal step δn after its computation. For δt getting smaller and smaller during a globalization loop, (35) can only be guaranteed, if

$$f'(x+\delta n)\delta t < 0. \tag{36}$$

However, $f'(x+\delta n)$ does not enter the computation of δt, only $f'(x)$, so there is no reason for (36) to hold, if δn is large. In this case we might be forced to completely reject tangential steps until the iterates are close enough to the constraint. For details see section 5.

Due to these two effects the design of a decrease based acceptance criterion is a delicate matter.

To measure the quality of tangential steps, taking the impact of the normal step into account, we estimate the ratio between actual and predicted decrease via

$$\eta := \frac{f(x + \delta x + \delta s) - m_{[\omega_f]}(\delta n)}{m_{[\omega_f]}(\delta x) - m_{[\omega_f]}(\delta n)}, \tag{37}$$

where $m_{[\omega_f]}$ was defined in (33). In this way we take the possible increase due to the normal step into account while avoiding any additional function evaluations. Moreover, the denominator is guaranteed to be negative for $\|\delta t\| > 0$. Then the natural criterion for acceptance of the tangential step is

$$\eta \geq \underline{\eta} \tag{38}$$

for a user-defined lower bound $\underline{\eta} \in]0, 1[$. For $\delta n = 0$ this reduces to the well-known standard decrease condition, which is used widely in trust-region methods [13], and also adapted in [8, 31] to a cubic regularization method in unconstrained optimization.

4.3 Definition of globalized steps

The computation of δx should be done in such a way as to satisfy the acceptance criteria (34) and (38). Motivated by (30), we introduce a parametrized model for Θ, equipped with an algorithmic parameter $[\omega_c] > 0$:

$$[\Theta](\xi) := \frac{[\omega_c]}{2}\|\xi\|. \tag{39}$$

The parameter $[\omega_c]$ is an estimate from below for the *affine covariant Lipschitz constant* ω_c, defined in (27). The step computation is then done in such a way as to guarantee

$$[\Theta](\delta x) = \frac{[\omega_c]}{2}\|\delta x\| \leq \Theta_{\text{aim}}, \tag{40}$$

where Θ_{aim} is a user provided desired contraction rate for δx.

Observe that (40) is a trust-region like constraint, which we could write alternatively as (cf. Figure 2):

$$\|\delta x\| \leq r_x := \frac{2\Theta_{\text{aim}}}{[\omega_c]}.$$

The algorithmic parameter Θ_{aim} is chosen in the interval $\Theta_{\text{aim}} \in]0, \Theta_{\text{acc}}[$. The condition $\Theta_{\text{acc}} > \Theta_{\text{aim}}$ is a prerequisite for finite termination of the inner loop.

Damping of normal step. Recall that the full normal step Δn is computed via (12) as the minimal norm correction satisfying $c(x)+c'(x)\Delta n = 0$. In view of (40) a damped normal step $\delta n = \nu\Delta n$ might then be computed as large as possible under the restrictions $[\Theta](\delta n) \leq \Theta_{\text{aim}}$ and $\nu \leq 1$.

However, if $[\Theta](\delta n) = \Theta_{\text{aim}}$, which holds at least for $\nu < 1$, the requirement $[\Theta](\delta x) \leq \Theta_{\text{aim}} = [\Theta](\delta n)$ then implies $\delta t = 0$. In order to leave some "elbow-space" for δt also in the case $\nu < 1$, we introduce an elbow-space factor $\rho_{\text{elbow}} \in]0, 1[$, and choose

$$\nu := \min\left\{1, \frac{2\rho_{\text{elbow}}\Theta_{\text{aim}}}{[\omega_c]\|\Delta n\|}\right\}. \tag{41}$$

This restriction can also be interpreted as a trust-region for δn (cf. Figure 2):

$$\|\delta n\| \leq r_n := \frac{2\rho_{\text{elbow}}\Theta_{\text{aim}}}{[\omega_c]}.$$

Computation of the total step. After δn has been computed, we have to compute the tangential step δt responsible for minimizing $f(x + \delta n + \delta t + \delta s)$. As the latter quantity is computationally inaccessible, we resort to minimizing its regularized model $m_{[\omega_f]}$, defined in (33):

$$\begin{aligned} \min_{\delta t \in X} m_{[\omega_f]}(\delta n + \delta t) \quad \text{s.t.} \qquad & c'(x)\delta t = 0 \\ & \frac{[\omega_c]}{2}\|\delta n + \delta t\| \leq \Theta_{\text{aim}} \end{aligned} \tag{42}$$

Compared to (15) we have add a cubic regularization term and a trust-region type constraint. However the considerations in Section 3.1.3 mainly carry over to this setting.

Now, tangential steps are computed as minimizers or at least directional minimizers along descent directions of (33). By orthogonality of the tangential and normal step, the trust region constraint on δx in (42) is equivalent to the trust region constraint

$$\|\delta t\| \leq \sqrt{\left(\frac{2\Theta_{\text{aim}}}{[\omega_c]}\right)^2 - \|\delta n\|^2} \tag{43}$$

on δt.

4.4 Update of Lipschitz estimates

If δx fails to satisfy (34), a new trial correction $\delta x_+ \neq \delta x$ has to be computed such that $\Theta(\delta x_+) \leq \Theta_{\text{acc}}$. As the computation of δx and δs is completely determined by the current iterate x and the Lipschitz estimates $[\omega_c]$ and $[\omega_f]$, those values have to be updated for computing δx_+.

Update of $[\omega_c]$. After a candidate correction δx and corresponding δs have been computed, we can compute (or update) the parameter $[\omega_c]$ by requiring the interpolation condition $[\Theta](\delta x) = \Theta(\delta x)$, i.e.,

$$[\omega_c] := \frac{2\Theta(\delta x)}{\|\delta x\|} = \frac{2\|\delta s\|}{\|\delta x\|^2}. \tag{44}$$

It follows immediately from (30) that $[\omega_c] \leq \omega_c$ and thus remains bounded, if ω_c exists.

Update of $[\omega_f]$. Here we use an adaption of the strategy proposed for unconstrained optimization in [38] to the equality constrained case. Therefore recall the definitions of the quadratic and cubic models

$$q(\delta x) = f(x) + f'(x)\delta x + \frac{1}{2}L_{xx}(x, p_x)(\delta x)^2$$

and

$$m_{[\omega_f]}(\delta x) := q(\delta x) + \frac{[\omega_f]}{6}\|\delta x\|^3.$$

By Lemma 3.2 $q(\delta x)$ is a second order consistent quadratic model for $f(x+\delta x+\delta s)$, so that we update $[\omega_f]$ as

$$[\omega_f] := \frac{6}{\|\delta x\|^3}\left(f(x+\delta x+\delta s) - q(\delta x)\right), \tag{45}$$

taking into account the restrictions

$$[\omega_f]^{\text{new}} \geq \rho_0[\omega_f]^{\text{old}} \quad \text{and} \quad [\omega_f]^{\text{new}} \leq \rho_1[\omega_f]^{\text{old}}$$

for $0 < \rho_0 < 1$ and $1 < \rho_1$. The first restriction guarantees $[\omega_f] > 0$, a necessary requirement for being able to determine finite tangential directions in the presence of non-convexities. The second dampens strong increases in the Lipschitz constant. This avoids the occurrence of oscillations of $[\omega_f]$. These restrictions can also be relaxed along the lines of [31, Sec. 3.4].

Successive updates of $[\omega_c]$ and $[\omega_f]$ yield a *predictor-corrector loop*, sketched in Algorithm 1, that terminates, if (34) is satisfied. In Section 5 we will show that this loop terminates finitely, as long as $\Theta_{\text{aim}} < \Theta_{\text{acc}}$.

In the context of the outer NLP iteration, the inner step computation loop is started with the estimate $[\omega_c]$ (and $[\omega_f]$, see below) from the previous iteration. The whole algorithm is started with an initial estimate for $[\omega_c]$ provided by the user.

4.5 Practical details

Increase of Lipschitz estimates. In the unconstrained case, failure of (38) yields an increase in $[\omega_f]$ at least by a factor of $1 + \frac{1+\eta}{2}$ [31]. Thus in the unconstrained case (i.e., in the absence of normal steps), repeated failure of the acceptance test yields a quick increase of $[\omega_f]$.

For constrained problems the expected minimal increase depends on the relative contributions of damped normal, resp. tangential, step to the composite step, i.e. on the quantity

$$\theta := \frac{\|\delta t\|}{\|\delta x\|},$$

only guaranteeing an increase of $[\omega_f]$ by a factor $g(\theta) \in [1, 1 + \frac{1+\eta}{2}]$ (see section 5) with $\lim_{\theta\to 0} g(\theta) = 1$. Thus if the iterates are not sufficiently close to the constraint stagnating updates of the Lipschitz constant may occur. In this case we should allow our algorithm to first improve feasibility before continuing optimization, i.e. we should discard the tangential step and accept the step $\delta x = \delta n$. Therefore we monitor the increase in the Lipschitz constant after failure of (38) and if

$$[\omega_f]^{\text{new}} < \Big(1 + \rho_s \frac{1+\eta}{2}\Big)[\omega_f]^{\text{old}}, \tag{46}$$

for some algorithmic parameter $0 < \rho_s < 1$, then we discard δt (which then is small relative to δx anyway), and accept the step $\delta x = \delta n$.

Combined update mechanism. The proposed acceptance test and update rules for the algorithmic parameters $[\omega_c]$ and $[\omega_f]$ are now combined in a single inner loop. Since both parameters may be increased or decreased in each step of this loop, depending, whether their corresponding acceptance criterion is fulfilled or not, a cyclic behavior of the inner loop may result if the following cases occur repeatedly:

i) A step is not acceptable in terms of (34), so $[\omega_c]$ is increased, but $[\omega_f]$ is decreased.

ii) A step is not acceptable in terms of (38), so $[\omega_f]$ is increased, but $[\omega_c]$ is decreased.

In order to guarantee that this case cannot occur we additionally have to ensure monotonicity of the Lipschitz estimates after first failure of the corresponding acceptance test. Thus we slightly modify our update rules, i. e. in each inner loop whenever

i) (34) has failed at least once, we do not allow decrease in $[\omega_c]$ after failure of (38),

ii) (38) has failed at least once, we do not allow decrease in $[\omega_f]$ after failure of (34).

In this way, if both (34) and (38) fail, we rule out cycling by strict monotonicity of the Lipschitz constants (see Section 5).

5 Finite termination of inner loops

Throughout this section we restrict the discussion to one inner loop. In order to show that it terminates after a finite number of rejected steps we first consider each Lipschitz constant and its corresponding acceptance test independently. We begin with the updates of $[\omega_c]$.

Lemma 5.1. *If a trial correction is rejected due to failure of the feasibility contraction test* (34)*, then* $[\omega_c]$ *is increased at least by the fixed factor* $\frac{\Theta_{\text{acc}}}{\Theta_{aim}}$*. Thus, as long as the tangential decrease test* (38) *does not fail, the inner loop terminates after a finite number of iterations.*

Proof. Using (44), failure of (34), and (40), the newly computed Lipschitz estimate satisfies

$$[\omega_c]^{\text{new}} \overset{(44)}{=} \frac{2\|\delta s\|}{\|\delta x\|^2} \overset{(34)}{>} \frac{2\Theta_{\text{acc}}}{\|\delta x\|} \overset{(40)}{\geq} \frac{\Theta_{\text{acc}}}{\Theta_{\text{aim}}}[\omega_c]^{\text{old}}$$

if (34) fails. □

Similarly, we obtain for the decrease criterion:

Lemma 5.2. *If a trial correction is rejected due to failure of* (38)*, then either* $[\omega_f]$ *is increased at least by the fixed factor* $1+\rho_s\frac{1-\eta}{2} > 1$*, or the trial correction is accepted, possibly discarding the tangential step. Thus as long as* (34) *does not fail the inner loop terminates after a finite number of iterations.*

Proof. By our mechanism, either $[\omega_f]$ is increased at least by a fixed factor

$$\min\{\rho_1, 1 + \rho_s\frac{1-\eta}{2}\},$$

or the tangential step is discarded, rendering $\delta x = \delta n$ an acceptable step. By Theorem 3.4 we can conclude that $[\omega_f]$ remains bounded within each inner loop (because δx is bounded), and so an infinite number of increases of $[\omega_f]$ by the above fixed factor is impossible. □

The lemmata 5.1 and 5.2 only considered the case that only one of the two acceptance tests fails. If we allow both tests to fail, cycling might occur. In this case the modification proposed in subsection 4.5 admits to transfer the above results to the general case.

Algorithm 2 Inner loop (case where Δt is not recomputed)

Require: Lipschitz constants $[\omega_c], [\omega_f]$, search directions $\Delta n, \Delta t$
  ContractionFailedOnce $\leftarrow$ false
  DecreaseFailedOnce $\leftarrow$ false
  DiscardTangentialStep $\leftarrow$ false
  repeat
    $\nu \leftarrow \frac{\rho_{\text{elbow}}\Theta_{\text{aim}}}{[\omega_c]\|\Delta n\|}$
    $\tau \leftarrow \min_{\tau \geq 0} m_{[\omega_f]}(\tau \Delta t)$
    if DiscardTangentialStep **then**
      $\delta x \leftarrow \nu \Delta n$
    else
      $\delta x \leftarrow \nu \Delta n + \tau \Delta t$
    $\delta s \leftarrow$ via (20)
    compute new Lipschitz constants $[\omega_c]^{\text{new}}, [\omega_f]^{\text{new}}$ via (44) and (45)
    if ContractionFailedOnce **then**
      $[\omega_c] \leftarrow \max([\omega_c], [\omega_c]^{\text{new}})$
    else
      $[\omega_c] \leftarrow [\omega_c]^{\text{new}}$
    if DecreaseFailedOnce **then**
      $[\omega_f] \leftarrow \max([\omega_f], [\omega_f]^{\text{new}})$
    else
      $[\omega_f] \leftarrow [\omega_f]^{\text{new}}$
    Accepted $\leftarrow$ true
    if (34) fails **then**
      Accepted $\leftarrow$ false
      ContractionFailedOnce $\leftarrow$ true
    else
      if (38) fails **then**
        Accepted $\leftarrow$ false
        DecreaseFailedOnce $\leftarrow$ true
        if (46) fails **then**
          DiscardTangentialStep $\leftarrow$ true
  until Accepted

Theorem 5.3. *Assume that the affine covariant Lipschitz conditions* (27)-(29) *hold. Then the inner loop, as described in Alg. 2, terminates after a finite number of iterations.*

Proof. We assume that the inner loop does not terminate finitely and show that this implies either $[\omega_c] \to \infty$ or $[\omega_f] \to \infty$. which is not consistent with Thm. 3.4.

If only one of the acceptance criteria (34) and (38) fails we get this behavior from Lemma 5.1 and Lemma 5.2, respectively.

Thus we only have to consider the case that both criteria fail. Let k be the first iteration where both criteria have failed before. Due to the modification introduced in subsection 4.5 none of the estimates for the Lipschitz constants is allowed to decrease during the following iterations in this inner loop. Then if the inner loop does not terminate finitely, at least one of the two acceptance criteria is violated infinitely often after the k-th iteration and either $[\omega_c] \to \infty$ or $[\omega_f] \to \infty$ holds. □

Discarding tangential steps. Let us discuss the case, where the tangential step is discarded, i.e., where $[\omega_f] < (1 + \rho_s \frac{1-\eta}{2})[\omega_f]^{\text{old}}$. Our aim is to justify that this algorithmic measure is necessary and useful. In particular, we show that the tangential step is only discarded, if $\|\delta t\| \ll \|\delta x\|$, (as long as $\rho_s \ll 1$ is chosen).

Before starting, we prove a basic property of the minimizers of the cubic model $m_{[\omega_f]}$.

Lemma 5.4. *Any directional minimizer δt of $m_{[\omega_f]}$ satisfies*

$$m_{[\omega_f]}(\delta x) - m_{[\omega_f]}(\delta n) \leq \frac{[\omega_f]}{12}\big(2\|\delta x\|^3 - 2\|\delta n\|^3 - 3\|\delta x\|\|\delta t\|^2\big). \tag{47}$$

Proof. From the symmetry of $\frac{1}{2}L_{xx}(x,p)(\delta t)^2 + \frac{[\omega_f]}{6}\|\delta n + \delta t\|^3$ in δt and the orthogonality $\langle \delta n, \delta t\rangle = 0$ it follows that

$$0 \geq m_{[\omega_f]}(\delta t) - m_{[\omega_f]}(-\delta t) = 2(f'(x) + L_{xx}(x,p)\delta n)\delta t.$$

Inserting this into the first order necessary optimality condition yields

$$0 = m'_{[\omega_f]}(\delta x)\delta t = \big(f'(x) + L_{xx}(x,p)\delta n\big)\delta t + L_{xx}(x,p)(\delta t)^2 + \frac{[\omega_f]}{2}\|\delta x\|\langle \delta x, \delta t\rangle \tag{48}$$

$$\leq L_{xx}(x,p)(\delta t)^2 + \frac{[\omega_f]}{2}\|\delta x\|\|\delta t\|^2 \tag{49}$$

Applying first (48) and then (49) to (33) we obtain

$$\begin{aligned} m_{[\omega_f]}(\delta x) - m_{[\omega_f]}(\delta n) &= (f'(x) + L_{xx}(x,p)\delta n)\delta t + \frac{1}{2}L_{xx}(x,p)(\delta t)^2 + \frac{[\omega_f]}{6}(\|\delta x\|^3 - \|\delta n\|^3) \\ &= -\frac{1}{2}L_{xx}(x,p)(\delta t)^2 - \frac{[\omega_f]}{2}\|\delta x\|\|\delta t\|^2 + \frac{[\omega_f]}{6}(\|\delta x\|^3 - \|\delta n\|^3) \\ &\leq \frac{[\omega_f]}{12}\big(2\|\delta x\|^3 - 3\|\delta x\|\|\delta t\|^2 - 2\|\delta n\|^3\big) \end{aligned}$$

and hence the claim. □

Since $[\omega_f]$ is defined by (45) we can compute for the update:

$$
\begin{aligned}
[\omega_f] &= \frac{6}{\|\delta x\|^3}\big(f(x+\delta x+\delta s) - q(\delta x)\big) \\
&= \frac{6}{\|\delta x\|^3}\big((f(x+\delta x+\delta s) - m_{[\omega_f]^{\text{old}}}(\delta n)) + m_{[\omega_f]^{\text{old}}}(\delta n) - q(\delta x)\big) \\
&= \frac{6}{\|\delta x\|^3}\big(\eta(m_{[\omega_f]^{\text{old}}}(\delta n) - m_{[\omega_f]^{\text{old}}}(\delta n)) + m_{[\omega_f]^{\text{old}}}(\delta n) - m_{[\omega_f]^{\text{old}}}(\delta x) + \frac{[\omega_f]^{\text{old}}}{6}\|\delta x\|^3\big) \\
&= \frac{6}{\|\delta x\|^3}(\eta-1)(m_{[\omega_f]^{\text{old}}}(\delta x) - m_{[\omega_f]^{\text{old}}}(\delta n)) + [\omega_f]^{\text{old}}.
\end{aligned}
$$

Since the step has been rejected, i.e., $\eta < \underline{\eta}$ we can continue, setting $\theta := \|\delta t\|/\|\delta x\|$:

$$
\begin{aligned}
[\omega_f] &> \frac{6}{\|\delta x\|^3}(1-\underline{\eta})(m_{[\omega_f]^{\text{old}}}(\delta n) - m_{[\omega_f]^{\text{old}}}(\delta x)) + [\omega_f]^{\text{old}} \\
&\overset{(47)}{\geq} \frac{6}{\|\delta x\|^3}(1-\underline{\eta})\frac{[\omega_f]^{\text{old}}}{12}(3\|\delta x\|\|\delta t\|^2 + 2\|\delta n\|^3 - 2\|\delta x\|^3) + [\omega_f]^{\text{old}} \\
&= (1-\underline{\eta})\frac{[\omega_f]^{\text{old}}}{2}(3\theta^2 + 2\frac{\|\delta n\|^3}{\|\delta x\|^3} - 2) + [\omega_f]^{\text{old}} \\
&= [\omega_f]^{\text{old}}\left(1 + \frac{1-\underline{\eta}}{2}\left(3\theta^2 + 2\sqrt{1-\theta^2}^3 - 2\right)\right)
\end{aligned}
$$

Thus, we obtain,

$$
\frac{[\omega_f]}{[\omega_f]^{\text{old}}} \geq g(\theta) := 1 + \frac{1-\underline{\eta}}{2}(3\theta^2 + 2(1-\theta^2)^{3/2} - 2), \quad \theta \in [0,1].
$$

The function g is monotonically increasing on $[0,1]$ and bounded by its local extrema

$$
1 = g(0) \leq g(\theta) \leq g(1) = 1 + \frac{1-\underline{\eta}}{2},
$$

where the case $\theta = 1$ corresponds to the case of unconstrained optimization, i.e. $\delta n = 0$ (cf. [31]). The other extreme $\theta = 0$ describes the case of a vanishing tangential step. Thus, if ρ_s is chosen small and

$$
g(\theta) \leq 1 + \rho_s \frac{1-\underline{\eta}}{2}.
$$

we conclude that $\theta = \|\delta t\|/\|\delta x\|$ is small as well, and thus tangential steps are only discarded, if their contribution to the total step is small anyway.

6 Transition to fast local convergence

In this section we discuss the transition of our method to fast local convergence. Of particular interest is to show that the Maratos effect does not occur. As usual for local convergence results, we will assume sufficient smoothness and second order sufficient optimality conditions (SSC) at the local minimizer.

To keep the discussion concise we do not aim for the most general results, but remain in a rather simple setting. In particular, we only consider the case that normal and tangential steps

can be computed exactly along Newton directions. This is in contrast to practical solvers, where at least the tangential steps are computed only inexactly up to a certain accuracy. To retain fast local convergence in that setting an appropriate accuracy matching strategy has to be developed and analyzed. This is subject to current work.

In the following we consider full Lagrange-Newton steps at an iterate (x, p):

$$(\Delta x, \Delta p) := L''(x,p)^{-1} L'(x,p), \tag{50}$$

which, in more detail satisfy the equation:

$$\begin{pmatrix} L_{xx}(x,p) & c'(x)^* \\ c'(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta p \end{pmatrix} + \begin{pmatrix} L_x(x,p) \\ c(x) \end{pmatrix} = 0.$$

As it is well known, at an SSC point x_* the Newton-Matrix $L''(x_*, p_{x_*})$ is continuously invertible, and a standard perturbation argument yields that the same holds true in a neighborhood of (x_*, p_{x_*}).

Consider first the classical, undamped Lagrange-Newton method:

$$(x_{k+1}, p_{k+1}) = (x_k, p_k) - L''(x_k, p_k)^{-1} L'(x_k, p_k). \tag{51}$$

Close to an SSC point that the undamped Lagrange-Newton method with iterates (x_k, p_k) is well defined, and converges locally superlinearly towards (x_*, p_{x_*}), if, e.g., Lipschitz conditions on L'' hold.

We will prove local superlinear convergence follows for the variant with the adjoint p_x defined by (6) (where w is a dummy dual variable which is discarded):

$$(x_{k+1}, w) = (x_k, p_{x_k}) - L''(x_k, p_{x_k})^{-1} L'(x_k, p_{x_k}) \tag{52}$$

that we are using (note the difference between (51) and (52): $p_k \neq p_{x_k}$). To this end, we will first show that this undamped iteration converges locally superlinearly, then we show that the globalized variant shows the same behavior.

As a preparatory step we show that small perturbations in p yield perturbations in the steps that are small *relative to the step length.*

Lemma 6.1. *Assume that $L_{xx}(x, p_{x_*})$ is positive definite and $c''(x) : X \times X \to P^*$ is bounded. Let p be a sufficiently small perturbation of p_{x_*} Denote by Δx_* the primal component of of* (50) *with argument $(x, p) = (x, p_{x_*})$ and by Δx the primal component of* (50) *with (x, p). Then eventually,*

$$\frac{\|\Delta x - \Delta x_*\|}{\|\Delta x\|} \leq c \|p - p_{x_*}\|.$$

Proof. By assumption $L_{xx}(x, p_{x_*})$ is positive definite on $\ker c'(x)$, i.e., there is $\alpha > 0$, such that

$$\alpha \|v\|^2 \leq L_{xx}(x, p_{x_*})(v, v).$$

Hence, for a close-by Lagrange multiplier p we know that $L_{xx}(x, p)$ is still positive definite on $\ker c'(x)$. Let Δx_* be the solution of (50) with (x, p_{x_*}), and Δx be the solution of (50) with p_{x_*} replaced by p. This implies that (using $f'(x) = L_x(x, p)$ on $\ker c'(x)$):

$$\begin{aligned} 0 &= f'(x)v + L_{xx}(x, p_{x_*})(\Delta x_*, v) && \forall v \in \ker c'(x) \\ 0 &= f'(x)v + (L_{xx}(x, p_{x_*}) + (p - p_{x_*})c''(x))(\Delta x, v) && \forall v \in \ker c'(x) \end{aligned}$$

Subtracting both equations yields:

$$0 = L_{xx}(x, p_{x_*})(\Delta x - \Delta x_*, v) + (p - p_{x_*})c''(x))(\Delta x, v).$$

Inserting $v = \Delta x - \Delta x_* \in \ker c'(x)$ (the normal components of the two steps do not differ) and using positive definiteness, we get:

$$\alpha\|\Delta x - \Delta x_*\|^2 \leq L_{xx}(x, p_{x_*})(\Delta x - \Delta x_*, \Delta x - \Delta x_*) = -(p - p_{x_*})c''(x)(\Delta x, \Delta x - \Delta x_*).$$

Taking norms, we obtain:

$$\alpha\|\Delta x - \Delta x_*\|^2 \leq \|p - p_{x_*}\|\|c''(x)\|\|\Delta x\|\|\Delta x - \Delta x_*\|$$

and thus the result:

$$\frac{\|\Delta x - \Delta x_*\|}{\|\Delta x\|} \leq c\|p - p_{x_*}\|. \tag{53}$$

□

Proposition 6.2. *Under the smoothness and SSC assumptions, described above, the iteration* (52) *converges locally superlinearly.*

Proof. For a pair $z = (x, p)$ let us introduce the notation $x := z_x = (x, p)_x$. For given (x, p), denote the next Newton iterate by (x_+, p_+). Since our update for p is not p_+, but p_{x_+}, computed via (6), we would like to estimate $\|x_+ - x_*\|$ in terms of $\|x - x_*\|$, namely $\|x_+ - x_*\| = o(\|x - x_*\|)$. Using the Newton step, we compute

$$\begin{aligned} x_+ - x_* &= (x_+ - x) + (x - x_*) = \Delta x - (x - x_*) \\ &= -(L''(x, p)^{-1}L'(x, p))_x + (x - x_*, 0)_x && (54) \\ &= -[(L''(x, p)^{-1}L'(x, p))_x - (L''(x, p_*)^{-1}L'(x, p_*))_x] && (55) \\ &\quad - (L''(x, p_*)^{-1}(L'(x, p_*) - L'(x_*, p_*) + L''(x, p_*)(x - x_*, 0)))_x && (56) \end{aligned}$$

By Lemma 6.1 we can estimate (55), while (56) can be estimated via the fundamental theorem of calculus and a continuity assumption on L'' with respect to x. This yields:

$$\|x_+ - x_*\| \leq c\|\Delta x\|\|p - p_*\| + \omega(\|x - x_*\|)\|x - x_*\|$$

Here $\omega(t)$ denote a generic function that tends to zero, if its argument tends to 0. Moreover, by Lemma 2.4 we infer $\|p - p_*\| = \omega(\|x - x_*\|)$. Next, we split $\|\Delta x\| = \|x_+ - x\| \leq \|x_+ - x_*\| + \|x - x_*\|$ and compute:

$$\|x_+ - x_*\| \leq \omega(\|x - x_*\|)(\|x_+ - x_*\| + \|x - x_*\|).$$

If $\omega(\|x - x_*\|) \leq \varepsilon < 1$ this yields

$$\|x_+ - x_*\|(1 - \varepsilon) = \omega(\|x - x_*\|)\|x - x_*\|,$$

hence $\|x_+ - x_*\| = o(\|x - x_*\|)$, i.e. local superlinear convergence. □

Let us now study the influence of our globalization scheme, close to an SSC point. For simplicity we assume that close to the minimizer, where L_{xx} is positive definite on $\ker c'(x)$ tangential steps are computed in direction of the minimizer Δt of (17), i.e. we have $\delta t = \tau \Delta t$, where $\tau \in]0,1]$ is a damping factor, computed via solving (33) in the affine subspace $\delta n + \operatorname{span}\{\Delta t\}$. Thus, we have the following relation for our optimization step δx and the full Lagrange-Newton step Δx:

$$\delta x = \delta n + \delta t = \nu \Delta n + \tau \Delta t, \quad \Delta x = \Delta n + \Delta t.$$

By orthogonality of Δn and Δt, as well as $\nu, \tau \in]0,1]$, this implies $\|\delta x\| \leq \|\Delta x\|$.

Theorem 6.3. *Assume that x_k converges to the SSC point x_* in the setting described above. Assume further that the Lipschitz conditions* (27),(28)*, and* (29) *hold in a neighborhood of x_*. Then its convergence is superlinear.*

Proof. First, we show that as $x_k \to x_*$ the corresponding damping factors ν_k and τ_k tend to 1.

By our assumptions, the algorithmic parameters $[\omega_c]$ and $[\omega_f]$ remain bounded along x_k, while $\delta x_k \to 0$ and $\Delta x_k \to 0$. Thus, it follows from (41) that $\nu_k = 1$ eventually.

Next, we show that $\tau_k \to 1$. Using the minimizing property of δx_k along the direction Δt_k and inserting this direction (in place of δt_k) into (48) we obtain:

$$\begin{aligned} 0 &= m'_{[\omega_f]}(\delta x_k)\Delta t_k \\ &= (f'(x_k) + L_{xx}(x_k, p_k)\delta n_k)\Delta t_k + L_{xx}(x_k, p_k)(\delta t_k, \Delta t_k) + \frac{[\omega_f]}{2}\|\delta x_k\|\langle \delta x_k, \Delta t_k\rangle \\ &= (f'(x_k) + L_{xx}(x_k, p_k)\delta n_k)\Delta t_k + \tau_k(L_{xx}(x_k, p_k)(\Delta t_k, \Delta t_k) + \frac{[\omega_f]}{2}\|\delta x_k\|\langle \Delta t_k, \Delta t_k\rangle). \end{aligned}$$

A similar equation holds for the full tangential step Δt_k, which minimizes $m_{[\omega_f]}$ for $[\omega_f] = 0$ (i.e. vanishing regularization term):

$$\begin{aligned} 0 = m'_0(\delta x_k)\Delta t_k &= (f'(x_k) + L_{xx}(x_k, p_k)\delta n_k)\Delta t_k + L_{xx}(x_k, p_k)(\delta t_k, \Delta t_k) \\ &= (f'(x_k) + L_{xx}(x_k, p_k)\delta n_k)\Delta t_k + L_{xx}(x_k, p_k)(\Delta t_k, \Delta t_k). \end{aligned}$$

Subtracting these two equations and solving for τ_k yields:

$$\tau_k = \frac{L_{xx}(x_k, p_k)(\Delta t_k, \Delta t_k)}{L_{xx}(x_k, p_k)(\Delta t_k, \Delta t_k) + \frac{[\omega_f]}{2}\|\delta x_k\|\langle \Delta t_k, \Delta t_k\rangle}.$$

Since L_{xx} is positive definite, uniformly around x_* and $[\omega_f]\|\delta x_k\| \to 0$ (by boundedness of $[\omega_f]$), this expression tends to 1, as $x_k \to x_*$.

It follows that

$$\frac{\|\Delta x_k - \delta x_k\|}{\|\Delta x_k\|} \to 0.$$

Moreover, the corresponding simplified normal steps δs_k satisfy as well:

$$\frac{\|\delta s_k\|}{\|\Delta x_k\|} \leq \frac{\|\delta s_k\|}{\|\delta x_k\|} \to 0,$$

which implies that $\|\Delta x_k - (x_{k+1} - x_k)\| = \|\Delta x_k - (\delta x_k + \delta s_k)\| = o(\|\Delta x_k\|)$, i.e. our computed steps approach the full Lagrange-Newton steps asymptotically and thus our iteration inherits local superlinear convergence from the Lagrange-Newton method. □

7 Computation of steps in the case of optimal control

Up to now we described our composite step method from the perspective of nonlinear optimization. Now we turn to some issues which arise in the context of optimal control problems, namely the practical computation of steps.

In this section we assume that by some Galerkin-type discretization our infinite dimensional problem has been reduced to a finite dimensional one. Then, after choosing bases for the now finite dimensional spaces X and P, which induces dual bases for X^* and P^*, the linear operators are represented by typically sparse matrices and their adjoints by transpose matrices. Moreover, the application of a linear functional $l \in X^*$ to an element $x \in X$ can be written in terms of their coefficient vectors as $l^T x$.

To capture the structure of optimal control problems we split the primal variable into state and control, $x = (y, u)^T$, and consider a problem of the form

$$\min_{x=(y,u)} f(x) \quad \text{s.t. } A(y) - Bu = c(y,u) = 0.$$

For brevity, we denote in the following $A'(y) = A$ and $c'(x) = C = (A \ -B)$, and we assume that A is continuously invertible.

In the context of optimal control, the saddle point matrices appearing in the computation of normal and tangential step via (13) and (18), respectively, read

$$H_n = \begin{pmatrix} M & C^T \\ C & \end{pmatrix} = \begin{pmatrix} M_y & & A^T \\ & M_u & -B^T \\ A & -B & \end{pmatrix}, \; H_t = \begin{pmatrix} L_{xx} & C^T \\ C & \end{pmatrix} = \begin{pmatrix} L_{yy} & L_{yu} & A^T \\ L_{uy} & L_{uu} & -B^T \\ A & -B & \end{pmatrix} \tag{57}$$

In the following, we only consider right hand sides of the form $(r_y, r_u, 0)^T$. This holds for the tangential step, but not for normal steps and simplified normal steps $z = (z_y, z_u, z_p)^T$ satisfying, for some right hand side $r = (0, 0, r_p)^T$, the system

$$H_n z = r.$$

In that case we can compute $z = z_0 + \tilde{z}$, with $z_0 = (A^{-1} r_p, 0, 0)^T$ and $\tilde{z}$ determined by

$$H_n \tilde{z} = r - H_n z_0 = \begin{pmatrix} -M_y A^{-1} r_p \\ 0 \\ 0 \end{pmatrix}.$$

Restricting the discussion to homogeneous constraints $c'(x) = 0$ we can exploit the fact that the restriction of the search space to $\ker c'(x)$ yields a convex unconstrained optimization problem for problems involving H_n. In conjugate gradient methods this restriction of the search space can be realized implicitly via *constraint preconditioners*. We discuss different strategies for the computation of the normal step, adjoint state and second order correction. We will restrict the discussion to the computation of $\tilde{z}$, as the same strategy will be applied for the computation of the adjoint state and second order correction, the latter in a similar affine space as in the computation of the normal step.

Regarding the tangential steps we also will incorporate the restriction to $\ker c'(x)$ with the help of constraint preconditioners, given either by a constraint preconditioner or a linear solver for the normal step. However, L_{xx} is in general not positive definite on $\ker c'(x)$. Therefore, before turning to the computation of tangential directions in Section 7.3, we discuss conjugate gradient methods for non-convex problems in Section 7.2.

7.1 Computing (simplified) normal steps and adjoint updates

In the following, we will consider the solution of the linear system

$$H_n z = r. \tag{58}$$

As M is positive definite and C has full rank, H_n is invertible. Next to the computation of the normal step Δn this system has to be solved in the computation of the adjoint state p_x, and the simplified normal step δs.

Depending on the size and structure of the problem, different solution methods are appropriate.

Problems of moderate size. If the problem size is moderate, the solution of (58) can be computed using a direct factorization of the saddle point matrix. The possibly high computational costs for the computation of the factorization are at least partially amortized by the multiple possibilities for its reuse.

This works fine for moderately sized, stationary optimal control problems, usually in two spatial dimensions. However, for larger problems, i.e., time dependent optimal control or three dimensional problems, sparse direct factorizations become prohibitively expensive, both in time and memory consumption. In our numerical experiments we use UMFPACK [15] to compute a LU-factorization of (58). An alternative that exploits symmetry of H_n is a sparse indefinite factorization $H_n = L^T B L$ [30].

Low dimensional control space. Let us consider the case that the space of controls is of low dimension (say, a couple of tens) and that A can be factorized by a sparse direct solver. In this case it is possible to use a Schur-complement approach in order to solve (58) by factorization of A and a couple of back-solves. This can be interpreted as a direct solution of the system (58) with a special pivoting strategy, often not recognized by standard sparse solvers.

Block Gauss elimination via A and A^T as pivots yields the dense but small Schur complement system

$$(M_u + S^T M_y S) z_u = r_u + S^T r_y - S^T S_y A^{-1} r_p,$$

where $S = A^{-1}B$ is a discretization of the linearized control-to-state mapping, well known in optimal control. If U is of dimension n_u, B has n_u columns, so n_u solves with A are required for computing the columns of S (which can be done in parallel), and a few solves with A and A^T are needed for computing right hand sides and performing back-substitution.

This approach is applicable and efficient, as long as sparse direct solvers applied to A are efficient. With the excellent solvers available today this strategy can be applied to fairly well resolved scalar elliptic problems in two and to some extent even three spatial dimensions. For a successful application of this strategy, we refer to [17], where an optimization problem from hyperthermia treatment was solved. The control consisted of 23 input parameters for the microwave antennas built into the hyperthermia applicator.

It is also possible to treat time dependent problems (with low-dimensional and time-independent control) in this setting. Then the solution of the system $Ay = b$ (and the corresponding adjoint equation) can be done by a time-stepping scheme.

High dimensional control space. If neither direct factorization nor the Schur complement reduction are applicable, iterative solvers have to be used. For a survey of saddle point solvers we refer to [2] and the references summarized in [32]. As the reduced problem on $\ker C$ is positive definite, we focus on conjugate gradient methods.

A popular method is the *projected preconditioned conjugate gradient method* (PPCG) that restricts the primal iterates x to the nullspace $\ker C$ by applying a projection in every iteration. The projection can be realized by *constraint preconditioners* of the form

$$P_{\text{sc}} = \begin{pmatrix} \tilde{M} & C^T \\ C & 0 \end{pmatrix}, \tag{59}$$

where $\tilde{M}$ is a symmetric positive definite preconditioner for M. One problem with this purely primal iteration is, that the residual vector does not approach zero in the course of the iteration, since a component in $(\ker C)^\perp$ remains. Rounding errors lead to a growing pollution of the reduced residual component in $\ker C$, which should converge to zero, and impede the convergence. To reduce this effect, both iterative refinement and a residual update strategy have been proposed in [21], which employs a least squares multiplier update to eliminate the residual part in $(\ker C)^\perp$.

An essentially equivalent, but computationally and conceptually simpler variant is to apply the PCG method to the full system (58) using the constraint preconditioner (59). This is justified by $P_{\text{sc}}^{-1} H_n$ being symmetric positive definite with $2 \dim Y$ eigenvalues 1 and $\dim U$ eigenvalues λ defined by the generalized eigenvalue problem $S^T M S x_z = \lambda S^T \tilde{M} S x_z$, where $S = A^{-1}B$ again the linearized, discretized control-to-state mapping.

The choice of $\tilde{M}$ in P_{sc} affects both the convergence rate and the computational effort for applying the preconditioner. A reasonable choice turns out to be the block triangular system

$$P_{\text{sc}} = \begin{pmatrix} 0 & 0 & A^T \\ 0 & M_u & -B^T \\ A & -B & \end{pmatrix} \qquad \text{i.e.} \qquad \tilde{M} = \begin{pmatrix} 0 & 0 \\ 0 & M_u \end{pmatrix}. \tag{60}$$

Note that here $\tilde{M}$ is spectrally equivalent to M on $\ker C$, as long as $S = A^{-1}B$ is continuous:

$$\langle u, u\rangle_U \le \langle x, x\rangle_{Y\times U} = \langle y, y\rangle_Y + \langle u, u\rangle_U = \langle Su, Su\rangle_Y + \langle u, u\rangle_U \le (1 + \|S\|^2_{U\to Y})\langle u, u\rangle_U.$$

Often, M_u is a scaled mass matrix and A an elliptic operator. Then we get an efficient but inexact constraint preconditioner by replacing A^{-1} by a fixed number of multigrid cycles. Since the constraint preconditioner has to project onto $\ker c'(x)$, and in the absence of further analysis, it is necessary to solve the arising systems $Ay = b$ to high accuracy. Relaxing this condition on P_{sc} is subject of current work. In contrast, M_u^{-1} can be replaced a fixed number of Chebyshev semi-iterations [19, 22, 37], which is straightforward and need not be overly accurate.

7.2 Computing tangential steps

The standard PPCG method admits the solution of saddle point problems of the form (58) since M is positive definite on $\ker C$. Now we discuss the solution of

$$H_t z = r,$$

where L_{xx} is in general not positive definite on $\ker C$. This requires additional strategies regarding the application of conjugate gradient methods. We will continue using the previously introduced notation, but mention that this section applies not only to constrained problems, but also to unconstrained ones.

Truncated conjugate gradient method. The most popular approach in this context is the truncated conjugate gradient method (TCG), which terminates as soon as a direction of non-positive curvature is found, see Alg. 3. The used search directions span a subspace on which H is positive definite and no further modification of standard CG implementations are required. Working as long as possible on the original problem, this approach seems to be quite effective in finding its way out of nonconvexities, see table 1. But we also observe that occasionally the TCG method does *not* lead us back into regions where the problem is convex, at least not in a reasonable number of iterations, see Tab. 1 for the parameters $c = 10^4$ and $d = 10^{-2}$ for our academic test problem (Section 8). Here the problem is that the algorithm runs into a nonconvexity which leads to termination of the TCG method after only 3–4 inner iterations. Thus, only a very small subspace of $\ker C$ is covered and the computed direction may be rather useless.

Algorithm 3 Truncated conjugate gradient method

Require: x, $r = Hx - b$, $Pg = r$, $\sigma = r^T g$, $d = -g$.
  while convergence test failed **do**
    if $d^T H d \leq 0$ **then**
      terminate
    $\alpha = \frac{\sigma}{d^T H d}$
    $x \leftarrow x + \alpha d$
    $r \leftarrow r + \alpha H d$
    $g \leftarrow P^{-1} r$
    $\beta \leftarrow \frac{r^T g}{\sigma}$
    $\sigma = r^T g$
    $d \leftarrow -g + d$

Regularized conjugate gradient method. An alternative strategy is to modify H by adding multiples of the preconditioner P. For some regularization parameter $\theta \geq 0$, the operator $H + \theta P$ is treated by a CG-method. We call this a regularized conjugate gradient method (RCG), see Alg. 4.

Remark 7.1. This seems to be of particular interest if $H + \theta P$ can be related to a physical model similar to H. As an example we mention problems from nonlinear elasticity where a simplified material model can be used for preconditioning. Then we may interpret $H + \theta P$ as the linearization of a model that corresponds to a more rigid material than the original one. Solutions of this problem enjoy more regularity properties than the result of a truncated CG method. Even if such an interpretation is not admissible the RCG method seems, in the presence of reasonable preconditioners, be more robust than the TCG method.

Since a preconditioner is rarely given explicitly, but as an algorithm for the application of P^{-1} to a vector, it is usually not possible to directly compute the application of P to a vector.

Still it is possible to implement the application of the operator $H+\theta P$ to the search direction d and thus construct a CG method for $H+\theta P$. The idea is to introduce an additional vector q, for which $q = Pd$ holds. Then, $(H+\theta P)d$ can be evaluated as $Hd + \theta q$.

The vector q is defined as follows. Starting with

$$q := -r = -Pg = Pd$$

we can update this vector in each CG iteration via

$$q \leftarrow -r + \beta q.$$

The following lemma shows that the claimed property $q = Pd$ indeed holds:

Lemma 7.2. *Consider the sequence q_k, computed as above, and the sequence d_k of search directions of the CG-method. Then*

$$q_k = Pd_k$$

Proof. For the initial iterate we have by definition $q_0 = -r_0 = Pd_0$. Let us assume for induction that $q_k = Pd_k$. Then by our update rules we have:

$$d_{k+1} = -g_{k+1} + \beta d_k$$

and

$$q_{k+1} = -r_{k+1} + \beta q_k = -Pg_{k+1} + \beta Pd_k = P(-g_{k+1} + \beta d_k) = Pd_{k+1},$$

which shows the desired result. □

Our regularization only requires few additional arithmetic operations. Moreover, the additional quantity $q = Pd$ is required anyway for termination criteria based on the P-norm [25, 34]. We will not employ such a norm here, but, when considering the inexact solution of normal steps, this norm is favorable in the computation of the tangential step as it allows a proper matching of inaccuracies.

The regularization parameter θ, which, as usual, should be as small as possible and as big as necessary, is chosen by a simple heuristic. Starting the computation with $\theta = 0$ we discard the computed iterates as soon as we encounter a direction d of non-positive curvature $d^T Hd < 0$ and update for some constant $c_d > 0$ the regularization parameter θ according to

$$\theta^{\text{new}} = \min\{\max\{\theta + \delta\theta, \underline{c}_\theta\theta\}, \bar{c}_\theta\theta\} \quad \text{with} \quad \delta\theta = \frac{c_d + |d^T(H+\theta P)d|}{d^T Pd}$$

with $1 < \underline{c}_\theta < \bar{c}_\theta$ such that the generated sequence of regularization parameters is strictly increasing each time a direction of non-positive curvature is encountered. The restriction $\theta^{\text{new}} \le \bar{c}_\theta\theta$ prevents the update from becoming too large, too quickly. Note, that for very large parameters θ the computed search direction approaches the steepest descent direction for the scalar product $\langle\cdot,\cdot\rangle_P$ induced by the preconditioner, and the condition number of $H + \theta P$ approaches 1.

After the update of θ we have to restart the CG iteration. Thus, it is to be expected that one application of RCG is more expensive than one application of TCG, but for difficult problems this additional cost is outweighed by a reduced number of outer iterations.

We refer to Tab. 1 for a comparison of outer iteration numbers in Example 8.1 below for different model parameters c and d. We observe that the RCG method behaves more robustly. However, each outer iteration tends to require more cg-steps, compared to TCG, because in case of non-convexity TCG terminates, while RCG restarts with a larger parameter.

Remark 7.3. Note the analogy of this to the well known Hessian modification methods. We stress, however, that we do not add multiplies of the identity matrix to the Hessian, but rather add implicitly multiples of the preconditioner. Thus we capture more problem structure in our modification.

Remark 7.4. There is a simple way to couple the choice of θ with our algorithmic parameter $[\omega_f]$. After the first step of the cg-method d_0 (the gradient step) has been computed, we can choose θ, such that the minimizer of $q+\theta/2\langle\cdot,\cdot\rangle_P$ along d_0 is also the directional minimizer of $m_{[\omega_f]}$. This will lead to gradient like steps for large values on $[\omega_f]$ and to Newton like steps, if $[\omega_f]$ is small.

Algorithm 4 Regularized conjugate gradient method

Require: x, $\theta \geq 0$, $r = Hx - b$, $Pg = r$, $\sigma = r^T g$, $d = -g$, $q = Pd = -r$.
  while convergence test failed **do**
    $z = d^T H d + \theta d^T q$
    if $z \leq 0$ **then**
      increase θ and restart
    $\alpha = \sigma / z$
    $x \leftarrow x + \alpha p$
    $r \leftarrow r + \alpha(Hd + \theta q)$
    $g \leftarrow P^{-1} r$
    $\beta = r^T g / \sigma$
    $\sigma = r^T g$
    $d \leftarrow -g + \beta d$
    $q \leftarrow -r + \beta q$

Hybrid conjugate gradient method. In order to benefit from both the small number of iterations which are often observed for the TCG method with the increased robustness of the RCG method we use a hybrid of both methods (Alg. 5). Motivated by the observation that the TCG method performs quite well except in the cases that it runs into nonconvexities early, we try to regularize only in these cases. The simplest approach would be to regularize only in case a nonconvexity is encountered in the first few CG iterations. Here we use another approach which implicitly contains a restriction on the minimum number of CG iterations. We choose to truncate the iteration in the case that a prescribed minimal decrease in the quantity underlying the termination criterion has been achieved. As termination criteria for the energy norm or the norm induced by preconditioner require some look-ahead parameter n, see [25, 34], we can not estimate the error before the $(n+1)^{st}$ iteration. Thus, in this case we will always regularize our problem as we can not decide if truncation makes sense. For Tab. 1 we used a termination criterion based on an estimate for the relative energy error from [34], see (61), which is admissible as we only work on the subspaces where $H + \theta P$ is positive definite.

Termination criterion. It is well known that the widely used termination criteria for the dual norm of the preconditioned residual only yields a useful termination criterion in the case that $\kappa(P^{-1}H) \approx 1$, which we can not expect here. Based on the observation that for strictly

Algorithm 5 Hybrid conjugate gradient method

Require: x, $\theta \geq 0$, $r = Hx - b$, $Pg = r$, $\sigma = r^T g$, $d = -g$, $q = Pd = -r$.
  while convergence test failed **do**
    $z = d^T H d + \theta d^T q$
    if $z < 0$ **then**
      if minimal decrease achieved **then**
        terminate
      else
        increase θ and restart
    $\alpha = \sigma / z$
    $x \leftarrow x + \alpha d$
    $r \leftarrow r + \alpha(Hd + \theta q)$
    $g \leftarrow P^{-1} r$
    $\beta = r^T g / \sigma$
    $\sigma = r^T g$
    $d \leftarrow -g + \beta d$
    $q \leftarrow -r + \beta q$

	TCG			RCG			HCG		
d \ c	10^2	10^3	10^4	10^2	10^3	10^4	10^2	10^3	10^4
10^{-5}	†	27	34	177	24	17	12	35	16
10^{-4}	24	34	29	21	36	17	24	22	14
10^{-3}	28	17	14	19	17	15	12	25	14
10^{-2}	10	19	16	18	14	18	13	18	17
10^{-1}	8	17	19	8	24	21	8	20	18
1	7	11	14	8	12	20	8	12	17

Table 1: Number of outer iterations on Example 8.1 for different model parameters c and d on a fixed uniform grid with $h_{\max} = 2^{-7}$, $\alpha = 10^{-6}$ (†: not convergent after 500 iterations).

convex problems the conjugate gradient method guarantees strict decrease in the energy norm, the representation formulae given in the original paper of Hestenes and Stiefel [25] have been used in [1, 34] to construct estimators for the absolute energy error $\|x - x_k\|_H$ and the relative energy error $\frac{\|x-x_k\|_H}{\|x\|_H}$. As all of the above presented conjugate gradient methods only work on subspaces where the, possibly regularized, problem is convex we can use the same termination criteria for nonconvex problems. Thus we employ the estimate for the relative energy error proposed in [34]. Exploiting only local H-orthogonality, the proposed estimate

$$\rho_{j,n} = \frac{\tilde{\rho}_{j,n}}{\xi_{j+n}} \tag{61}$$

with $\xi_{j+n} = \tilde{\rho}_{0,j+n} + b^T x + r_0^T x$ and

$$\tilde{\rho}_{j,n} = \sum_{i=j}^{j+n-1} \alpha_i r_i^T g_i$$

is numerically stable. All quantities are available during computation, the only drawback lying in the fact that we need to perform $j + d$ iterations, for some look-ahead parameter d, in order to estimate the relative energy error in the j-th step. As the conjugate gradient method guarantees descent in the energy norm in each iteration,

$$\|x - x_{j+d}\|_H < \|x - x_j\|_H,$$

we accept the last iterate x_{j+d} if the estimate for $\frac{\|x-x_j\|_H}{\|x\|_H}$ is accepted.

Far from the solution it does not make much sense to spend significant effort in the computation of highly accurate tangential directions. Therefore, following [16, Ch. 2.3.3], we choose a minimal accuracy of $\delta_0 = 0.25$. This guarantees that at least the leading two binary digits of the computed direction are correct. When getting close to the solution we should increase the prescribed accuracy in order to profit from the fast local convergence of the Newton-Lagrange scheme. For constrained optimization problems this is not at all a trivial issue and under current investigation. Here we only employ a heuristic argument. Therefore we decide being close to the solution if in the last, say $(k-1)^{st}$, step

i) no damping occurred,

ii) no direction of non-positive curvature was encountered in the computation of the tangential direction,

iii) and the estimate of the Kantorovich quantity satisfies $[\omega_c]\|\delta x_{k-1}\| < 1$.

In this case we adjust the desired accuracy to

$$\delta_k = \min\{\delta_0, [\omega_f]\|\delta x_{k-1}\|\},$$

cf. [38]. As desired relative accuracy for the solution we use $\delta_\infty = 10^{-6}$.

Note that the above choice $\delta_0 = 0.25$ implies that our algorithm will often overlook the presence of directions of negative curvature. Therefore, to illustrate the differences between the different conjugate gradient methods in dealing with nonconvexities, we employed $\delta_0 = 10^{-3}$ in the computations for Tab. 1. As maximal attainable accuracy we heuristically chose $\epsilon_{\max} = 10^{-11}$ in all our computations.

7.3 Application within composite step method

Let us finally summarize the step computations within the different settings. For the computations of δn, p_x, and δs we can always assume positive definiteness of M on $\ker C$, and thus unique solvability of the corresponding system. For moderately sized problems, or a low dimensional control space the solution can be found by direct elimination methods. Otherwise, a PPCG method can be used with a constraint preconditioner, as described above, and by positive definiteness of M we can expect that PPCG can compute solutions up to any accuracy.

The situation is different for tangential steps δt. As L_{xx} is in general indefinite we use one of the modifications from Section 7.2 of PPCG for nonconvex problems to compute descent directions for the cost functional. Again, the restriction of the tangential directions to $\ker c'(x)$ is incorporated with the help of a constraint preconditioner. For problems of moderate size or low dimensional control space we can reuse the direct factorization computed for the determination of the normal step as preconditioner. If this approach is not admissible we will use the same preconditioner as in the PPCG method for the computation of the normal step, given in (60).

There are many ways conceivable to introduce an influence of $[\omega_C]$ and $[\omega_f]$ into the computation of δt. For example, the value of $m_{[\omega_f]}$ at the cg-iterates could be monitored, or it could be monitored, if the cg-iterates leave the feasible region, used in (42). In our implementation we simply use the result of the cg-iteration, as described above, and compute δt as rescaling, such that $\delta x = \delta n + \delta t$ is a feasible directional minimizer of (42), i.e., δx minimizes $m_{[\omega_f]}$ on $\delta n + \operatorname{span}(\delta t)$.

8 Numerical examples

We provide two examples to illustrate the performance of the proposed composite step method on optimal control problems. First an academic two-dimensional example is presented. There we can easily control the nonlinearity by one of the model parameters. Secondly we give a somewhat more realistic problem arising in implant shape design, both on a simplified and a real patient geometry. In all examples we will use the Tikhonov-regularized tracking type cost functional

$$J(y,u) = \frac{1}{2}\|y - y_{\text{ref}}\|^2_{L^2(D_y)} + \frac{\alpha}{2}\|u\|^2_{L^2(D_u)}, \tag{62}$$

where $\alpha > 0$ is the Tikhonov regularization parameter and y_{ref} the prescribed solution. The sets $D_y \subseteq \Omega$ and $D_u \subseteq \Omega$ characterize the observation region as well as the region where the control acts. All examples were implemented with the finite element library Kaskade7 [20] using linear Lagrange elements. For the computation of tangential directions the HCG method was employed.

8.1 An academic example

In our first example we consider an optimal control problem in two dimensions with distributed control and observation. The constraints are given by a simple nonlinear model of heat transfer, which we consider in its weak formulation

$$c(y,u) = 0, \tag{63}$$

where, for some test function v,

$$c(y,u)v := \int_\Omega \nabla v^T \kappa(y) \nabla y \,\mathrm{dx} - \langle u, v\rangle_{L^2(\Omega)} \tag{64}$$

with isotropic heat conduction tensor $\kappa(y)(x) = (c|y(x)|^2 + d)I$ and $\Omega =]0,1[^2$ is the unit square. With the parameters $c, d > 0$ we can modify the influence of the nonlinear part and the distance to a singular problem. The optimal control of such a problem was analyzed in [10], where it was shown in particular that $y \in C(\overline{\Omega})$ for all $u \in L_2(\Omega)$ implying boundedness of $\kappa(y)$.

As desired displacement we set, see Fig. 3a,

$$y_{\text{ref}}(x_1, x_2) = 12(1 - x_2)x_2(1 - x_1)x_1.$$

In order to not severely underestimate the length of the normal steps, which might degrade the convergence speed of the composite step method, we use local scalar products at each iterate y_k of the form

$$\langle (y, u), (z, v) \rangle = \langle y, z \rangle_{M_y(y_k)} + \langle u, v \rangle_{M_u}$$

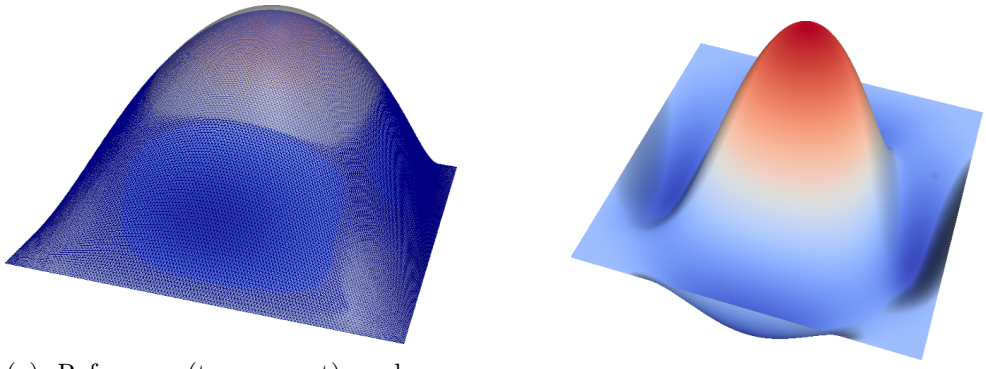
with

$$\langle y, z \rangle_{M_y(y_k)} = \int_\Omega \nabla y^T \kappa(y) \nabla z \, \mathrm{dx} + \langle y, z \rangle_{L^2(\Omega)}$$

and

$$\langle u, v \rangle_{M_u} = \alpha \langle u, v \rangle_{L^2(\Omega)}.$$

Then, as $L_{uu}(y, u, p)(u, v) = \langle u, v \rangle_{M_u}$, the preconditioner renders the PPCG-method independent of the Tikhonov regularization parameter.



(a) Reference (transparent) and computed solution.

(b) Corresponding control.

Figure 3: Computed solution and control for $c = 10$, $d = 0.1$, $\alpha = 10^{-6}$.

Computed control and solution are exemplarily given for $c = 10$, $d = 0.1$ and $\alpha = 10^{-6}$ in Fig. 3. In Tab. 2, iteration numbers for various choices of the model parameters are given for a fixed choice of the regularization parameter $\alpha = 10^{-6}$. Note that the iteration numbers slightly deviate from the ones reported in Tab. 1. This is due to differences in the required accuracy far from the solution. In practice we typically choose $\delta_0 = 0.25$. In contrast in Tab. 1 we required $\delta_0 = 10^{-3}$, which leads to a larger number of iterations where we have to deal with search directions for which the problem exhibits non-positive curvature.

8.2 Pressure-type control for rubbery hyperelastic materials

Now we consider a simplified example from implant shape design with control and observation on disjoint parts of the boundary [27]. Thus we work with the same function spaces for Y and P and replace the control space by $U = L^2(\Gamma_c)$, where Γ_c denotes the control boundary. In the following the state variable y describes the deformation of an elastic material and the control u can be interpreted as pressure on the control boundary. We consider a material that can be described by a compressible Mooney-Rivlin material law

$$W(y) = c_0 \iota_1(C) + c_1 \iota_2(C) + \Gamma(\det(\nabla y)),$$

d \ c	1	10^1	10^2	10^3	10^4	10^5
10^{-5}	5	6	17	20	14	16
10^{-4}	5	6	13	28	22	12
10^{-3}	4	6	17	23	17	16
10^{-2}	4	6	13	15	17	19
10^{-1}	4	6	10	19	21	19
1	5	6	9	14	23	18

Table 2: Number of outer iterations for Example 8.1 with different model parameters c and d on a fixed uniform grid with $h_{\max} = 2^{-7}$, $\alpha = 10^{-6}$

with strain tensor $C = \nabla y^* \nabla y$, first and second invariant

$$\iota_1(C) = \mathrm{tr}(C) \qquad \text{and} \qquad \iota_2(C) = \frac{1}{2}\left(\mathrm{tr}(C)^2 - \mathrm{tr}(C^2)\right),$$

and volumetric penalty

$$\Gamma(s) = c_3 s^2 - c_4 \log(s).$$

In the following two examples the material parameters are chosen according to [11] such that near the reference configuration the constitutive relation fades into the descriptions of linearized elasticity for material parameters $E = 1$ and $\nu = 0.45$. The Poisson ratio ν describes the compressibility of the material, which here is assumed to be only slightly compressible. Young's modulus E describes the rigidity of the material. Appearing as a spatially constant factor in the material parameters and indirectly, via the variational equality, in u we can set $E = 1$ w.l.o.g. For $E \neq 1$, the corresponding magnitude of pressure then is $\frac{1}{E}u$. Thus, in our computations we employ the corresponding material constants

$$c_1 = c_2 \approx 0.086206, \quad c_3 \approx 0.689655, \quad c_4 \approx -1.896552.$$

The corresponding stress tensor is

$$\sigma(\nabla y(x)) = \frac{\partial W(y(x))}{\partial y(x)}.$$

On the control boundary $\Gamma_c = \{x \in \Omega : \ x_2 = 0\}$ we impose boundary conditions of the form

$$\sigma(\nabla y(x))n = g(x)\mathrm{cof}(\nabla y(x))n,$$

where n is the surface normal, u the magnitude of pressure and cof the cofactor matrix. This boundary condition corresponds to a pressure type boundary condition

$$\hat{\sigma}(\hat{x})\hat{n} = u(\hat{x})\hat{n}, \quad \hat{x} \in y(\Gamma_c)$$

on the deformed domain and results from the fact that static equilibria for elastic materials must be formulated on the deformed domain and then transformed back to the undeformed reference configuration. The observation boundary is denoted by Γ_o. On the remaining part

of the boundary $\Gamma_d = \partial\Omega \setminus \{\Gamma_c \cup \Gamma_o\}$ we impose homogeneous Dirichlet boundary conditions. The equation that describes the corresponding equilibrium of forces then is given through

$$0 = c(y,u)v := \int_\Omega \sigma(\nabla y) : \nabla v \,\mathrm{d}\mu - \int_{\Gamma_c} u\mathrm{cof}(\nabla y) n v \,\mathrm{ds}.$$

For M_y we use the symmetric part of the corresponding description of the constitutively linearized theory, the St.Venant-Kirchhoff law. Then we have

$$M_y h_1 h_2 = \lambda \mathrm{tr}(C'\nabla h_1)\mathrm{tr}(C'\nabla h_2) + 2\mu\langle C'\nabla h_1, C'\nabla h_2\rangle,$$

with Lamé constants

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \approx 3.10 \qquad \text{and} \qquad \mu = \frac{E}{2(1+\nu)} \approx 0.34.$$

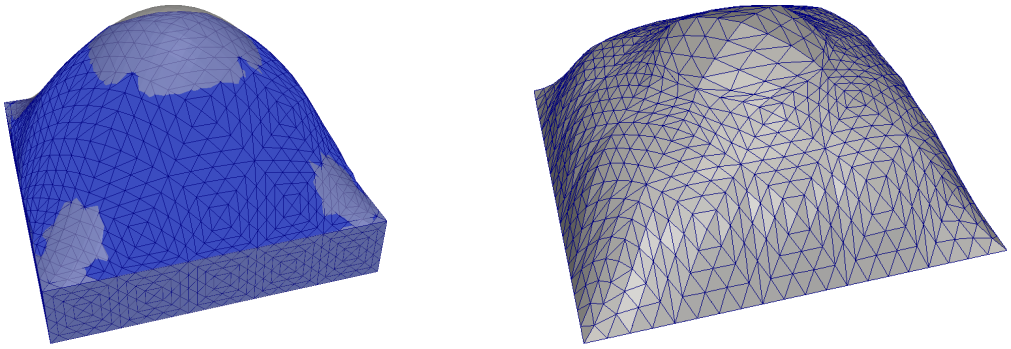
The discretization of L_{uu} is a mass matrix, rescaled with α and we choose the same matrix for the control part of the scalar product $M_u = L_{uu}$.

We consider two examples which arise as simplifications of problems of implant shape design. First we consider a simple geometry and an analytic reference shape. Then we turn to a real patient geometry and an estimated desired shape. In this case reasonable implants are relatively flat and induce only moderate deformations of the soft tissue.

Constraints from nonlinear elasticity do not fully fit into our chosen setting. The density of the set

$$Y_\infty = \{y \in W^{1,p}(\Omega) : \int_\Omega W(y) \,\mathrm{d}\mu = \infty\}$$

in $W^{1,p}(\Omega)$ may yield difficulties. In particular in the evaluation of the right hand side, for the computation of the simplified normal step, it may happen that $y_k + \delta y_k \in Y_\infty$. In this case we repeatedly adjust the normal and tangential step damping factors according to $\nu_{\text{new}} = \frac{1}{2}\nu$, resp. $\tau_{\text{new}} = \frac{1}{2}\tau$, until $y_k + \delta y_k \notin Y_\infty$, or equivalently $\det(\nabla(y_k + \delta y_k)) > 0$.



(a) Reference (transparent) and computed solution.

(b) Corresponding adjoint state on the control boundary.

Figure 4: A rubber model with pressure-type boundary conditions on a simple geometry ($\alpha = 0.1$).

An example on a simple geometry We consider the domain $\Omega =]0,1[\times]0,1[\times]0,0.2[$ with discretization as illustrated in 4a. As desired deformation on the observation boundary

$\Gamma_o = \{x \in \Omega : \ x_2 = 0.2\}$ we set

$$y_{\mathrm{ref}}(x) = \begin{pmatrix} 0 \\ 0 \\ z_{\mathrm{ref}}(x) \end{pmatrix} \qquad \text{with} \qquad z_{\mathrm{ref}}(x) = 8x_0x_1(1-x_0)(1-x_1),$$

see Fig. 4a. The regularization parameter is chosen as $\alpha = 0.1$. Computed solution, desired surface shape and the adjoint state on the control boundary are given in Fig. 4.

An example from implant shape design. We consider a more realistic example from implant shape design. Given a desired shape y_{ref} of the superficial skin the task is to compute a corresponding implant. Thus the skin determines the observation boundary, whereas the control boundary is given by the contact surface between soft tissues and bones. As illustrated in Fig. 5 we cut out the relevant part of the soft tissue. For simplicity we impose homogeneous Dirichlet boundary conditions on the artificial soft tissue boundary.

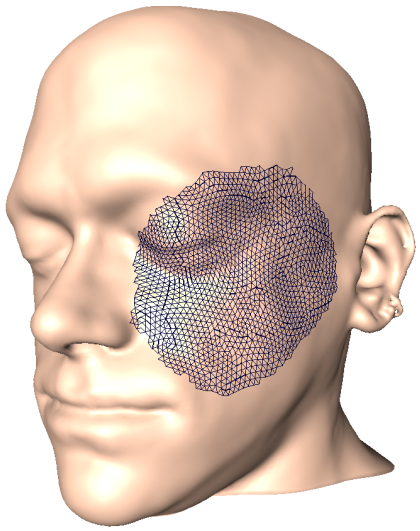
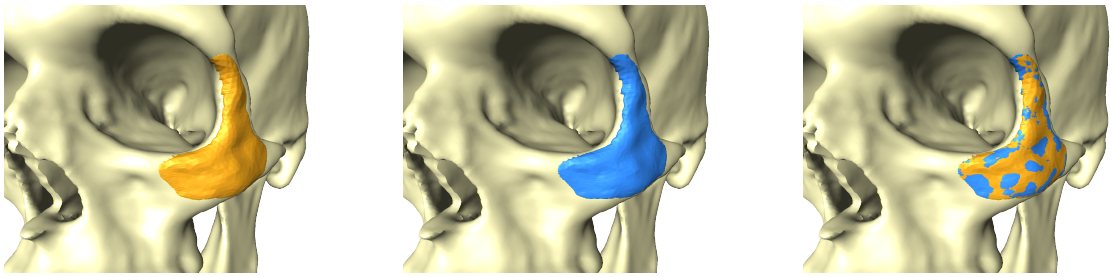


Figure 5: Patient geometry and computational domain.

In this example there is only a thin layer of soft tissue between the implant and the skin. For this reason, this problem is not too hard to solve. The reference surface was computed from the insertion of a reference implant. Given a relative tolerance of $\delta_0 = 10^{-3}$ and regularization parameter $\alpha = 0.05$, the implant was computed within 7 iterations without requiring globalization. The extraction of the implant shape and the generation of the graphics in this paragraph were done with the visualization tool Zibamira [33]. Comparing both implant shapes in Fig. 6 shows that differences in both position and shape of implant are not visible.

9 Conclusion

The composite step method considered in this work combines algorithmic features of cubic regularization algorithms and affine covariant Newton methods. Affine covariance leads to a



(a) Reference implant. (b) Computed implant. (c) Comparison of implants.

Figure 6: Computed and reference implant.

non-standard globalization scheme that does not rely on a monotonicity property, but rather on an estimate of the local Newton contraction. Finite termination of inner loops and fast local convergence of the method have been shown. A key ingredient was the double role of the simplified normal step as an indicator for Newton contraction and as a second order correction. Iterative solution techniques for the arising linear systems were discussed in the context of optimal control, and some numerical results were presented, including one arising in a medical application.

Up to now, inexactness of the step computation, in particular termination criteria for the iterative solvers, have not been discussed in detail. Different issues arise: in particular the conditions $\delta t \in \ker c'(x)$ and $\delta s, \delta n \in \ker c'(x)^{\perp}$ should be relaxed to allow for early termination of the CG-method and the use of inexact solves of the involved PDEs. Affine covariance will have a major impact also in this respect. In a similar fashion we can incorporate adaptive grid refinement into our algorithm.

From the theoretical side, a proof of global convergence is missing up to now. This will require some modifications of the algorithm. Certainly, a fraction of Cauchy decrease condition will be needed for the tangential step, but also globalization with respect to feasibility is still an open issue, even for affine covariant Newton methods for nonlinear equations.

References

[1] M. Arioli. A stopping criterion for the conjugate gradient algorithm in a finite element method framework. *Numer. Math.*, 97:1–24, 2004.

[2] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point systems. *Acta Numerica*, 14:1–137, 2005.

[3] D. Braess. *Finite Elements*. Cambridge University Press, 1997.

[4] H. Brézis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer, 2011.

[5] R.H. Byrd, F.E. Curtis, and J. Nocedal. An inexact Newton method for nonconvex equality constrained optimization. *Math. Program.*, 122(2, Ser. A):273–299, 2010.

[6] R.H. Byrd, J.C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Math. Program.*, 89(1, Ser. A):149–185, 2000.

[7] R.H. Byrd, M.E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM J. Optim.*, 9(4):877–900, 1999. Dedicated to John E. Dennis, Jr., on his 60th birthday.

[8] C. Cartis, N. Gould, and P.L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math, Prog.*, 127(2):245–495, 2011.

[9] C. Cartis, N. Gould, and P.L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Prog.*, 130(2):295–319, 2011.

[10] E. Casas and F. Tröltzsch. First- and second-order optimality conditions for a class of optimal control problems with quasilinear elliptic equations. *SIAM J. Control Optim.*, 48(2):688–718, 2009.

[11] P.G. Ciarlet. *Mathematical elasticity. Volume I: Three-dimensional elasticity*, volume 20 of *Studies in Mathematics and its Applications*. North-Holland, 1988.

[12] D. Clever, J. Lang, S. Ulbrich, and C. Ziems. Generalized multilevel SQP-methods for PDAE-constrained optimization based on space-time adaptive PDAE solvers. In *Constrained optimization and optimal control for partial differential equations*, volume 160 of *Internat. Ser. Numer. Math.*, pages 51–74. Birkhäuser/Springer Basel AG, Basel, 2012.

[13] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, 2000.

[14] F.E. Curtis, T.C. Johnson, D.P. Robinson, and A. Wächter. An inexact sequential quadratic optimization algorithm for nonlinear optimization. *SIAM J. Optim.*, 24(3):1041–1074, 2014.

[15] T.A. Davis and I.S. Duff. An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM J. Mat. Anal. and Appl.*, 18(1):140–158, 1997.

[16] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer, 2004.

[17] P. Deuflhard, A. Schiela, and M. Weiser. Mathematical cancer therapy planning in deep regional hyperthermia. *Acta Numerica*, 21:307–378, 5 2012.

[18] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91(2, Ser. A):239–269, 2002.

[19] G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numer. Math.*, 3:147–156, 1961.

[20] S. Götschel, M. Weiser, and A. Schiela. Solving optimal control problems with the Kaskade 7 finite element toolbox. In A. Dedner, B. Flemisch, and R. Klöfkorn, editors, *Advances in DUNE*, pages 101–112. Springer, 2012.

[21] N. Gould, M. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM J. Sci. Comput.*, 23(4):1376–1395, 2001.

[22] M. H. Gutknecht and S. Röllin. The Chebyshev iteration revisited. *Par. Comp.*, 28:263–283, 2002.

[23] M. Heinkenschloss and D. Ridzal. A matrix-free trust-region SQP method for equality constrained optimization. *SIAM J. Optim.*, 24(3):1507–1541, 2014.

[24] M. Heinkenschloss and L.N. Vicente. Analysis of inexact trust-region SQP algorithms. *SIAM J. Optim.*, 12(2):283–302, 2001/02.

[25] M. R. Hestenes and E. Stiefel. Methods of conjugate of gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, 49:409–436, 1952.

[26] A.D. Ioffe and Tihomirov V.M. *Theory of extremal problems*. North-Holland Publishing Company, 1979.

[27] L. Lubkoll, A. Schiela, and M. Weiser. An optimal control problem in polyconvex hyperelasticity. *SIAM J. Cont. Opt.*, 52(3):1403–1422, 2014.

[28] E. O. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Boulder, CO, USA, 1989. UMI Order No: GAX89-23520.

[29] D. Ridzal. *Trust-region SQP methods with inexact linear system solves for large-scale optimization*. ProQuest LLC, Ann Arbor, MI, 2006. Thesis (Ph.D.)–Rice University.

[30] O. Schenk and K. Gärtner. On fast factorization pivoting methods for sparse symmetric indefinite systems. *Elec. Trans. Numer. Anal*, 23:158–179, 2006.

[31] A. Schiela. A flexible framework for regularization algorithms for non-convex optimization in function space. Technical report, Technische Universität Hamburg-Harburg, 2014.

[32] J. Schöberl and W. Zulehner. Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 29(3):752–773, 2007.

[33] D. Stalling, M. Westerhoff, and H.-C. Hege. *The Visualization Handbook*, chapter Amira: a highly interactive system for visual data analysis, pages 749–767. Elsevier, 2005.

[34] Z. Strakoš and P. Tichý. Error estimation in preconditioned conjugate gradients. *BIT Numerical Mathematics*, 45(4):789–817, 2005.

[35] A. Vardi. A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM J. Numer. Anal.*, 22(3):575–591, 1985.

[36] S. Volkwein and M. Weiser. Affine invariant convergence analysis for inexact augmented Lagrangian-SQP methods. *SIAM J. Control Optim.*, 41(3):875–899 (electronic), 2002.

[37] A. Wathen and T. Rees. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *ETNA. Electr. Trans. Numer. Anal.*, 34:125–135, 2008.

[38] M. Weiser, P. Deuflhard, and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Opt. Meth. Softw.*, 22(3):414–431, 2007.

[39] E. Zeidler. *Nonlinear Functional Analysis and its Applications*, volume I. Springer, New York, 1986.

[40] J. Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact SQP methods for PDE-constrained optimization. *SIAM J. Optim.*, 21(1):1–40, 2011.