Analysis of Inexact Trust–Region SQP Algorithms

Matthias Heinkenschloss

Luis N. Vicente

TR99-18

September 1999 (revised December 2000)

Department of Computational and Applied Mathematics
MS 134
Rice University
6100 Main Street
Houston, TX 77005–1892

ANALYSIS OF INEXACT TRUST–REGION SQP ALGORITHMS

MATTHIAS HEINKENSCHLOSS * AND LUíS N. VICENTE †

Abstract. In this paper we extend the design of a class of composite–step trust–region SQP methods and their global convergence analysis to allow inexact problem information. The inexact problem information can result from iterative linear systems solves within the trust–region SQP method or from approximations of first–order derivatives. Accuracy requirements in our trust–region SQP methods are adjusted based on feasibility and optimality of the iterates. Our accuracy requirements are stated in general terms, but we show how they can be enforced using information that is already available in matrix–free implementations of SQP methods. In the absence of inexactness our global convergence theory is equal to that of Dennis, El–Alem, Maciel (SIAM J. Optim., 7 (1997), pp. 177–207). If all iterates are feasible, i.e., if all iterates satisfy the equality constraints, then our results are related to the known convergence analyses for trust–region methods with inexact gradient information for unconstrained optimization.

Keywords. Nonlinear programming, trust–region methods, inexact linear systems solvers, Krylov subspace methods, optimal control

AMS subject classification. 49M37, 90C06, 90C30, 90C55

1. Introduction. In this paper we study a class of trust–region sequential quadratic programming (SQP) algorithms for the solution of minimization problems with nonlinear equality constraints. Our aim is to extend the design of these algorithms and their convergence theory to allow the use of inexact problem information that originates from inexact first–order derivative information or from the use of inexact linearized constraint equation or adjoint equation solves.

The problems we are interested in are of the form

$$\begin{aligned} \min \quad & f(y,u), \\ \text{s.t.} \quad & C(y,u) = 0, \end{aligned} \tag{1.1}$$

where $y \in I\!R^m$, $u \in I\!R^{n-m}$, $f : I\!R^n \longrightarrow I\!R$, $C : I\!R^n \longrightarrow I\!R^m$, $m < n$. Our theory assumes that f and C are at least twice continuously differentiable. Variants of the algorithms, however, require only first–order derivative information. Our research is motivated by discretized optimal control problems [16, 18, 21], parameter identification problems and inverse problems [28, 31], and design optimization [4, 24]. In these applications y represents the discretized state variables and u represents the discretized controls, parameters, or design variables, respectively, and the nonlinear constraint $C(y,u) = 0$ is the discretized state equation. For many of the above mentioned applications the solution of linear equations of the type

$$C_y(y,u)z = d \quad \text{or} \quad C_y(y,u)^T z = d, \tag{1.2}$$

where y, u and d are given and where $C_y(y,u)$ and $C_u(y,u)$ are the partial Jacobians with respect to y and u, respectively, is costly and has to be accomplished by iterative methods. In optimal control, parameter identification, or optimal design problems the equations (1.2) are related to the linearized state equations and the adjoint equations, respectively, and it is often desirable to solve such equations using application specific methods such as Krylov–subspace, multigrid, or domain decomposition methods. Hence exact solutions of linear systems (1.2) are not available; only approximate solutions with a specified residual tolerance can be obtained.

*Department of Computational and Applied Mathematics Rice University, Houston, TX 77005–1892, USA. E-Mail: heinken@caam.rice.edu. This author was supported by the DoE under Grant DE–FG03–95ER25257 and by CRPC CCR–9120008.

†Departamento de Matemática, Universidade de Coimbra, 3001–454 Coimbra, Portugal. E-Mail: lvicente@mat.uc.pt. Support for this author was provided by Centro de Matemática da Universidade de Coimbra, FCT, and Praxis XXI 2/2.1/MAT/346/94.

Composite step trust–region SQP methods are used successfully to solve large scale optimization problems. However, existing convergence theories, which are nicely reviewed in [5], rely on the exact solution of linear systems of the form (1.2). Most existing implementations of SQP methods, use dense or sparse linear algebra methods to accomplish the linear system solves. As we have mentioned before this is not feasible for several of the applications we have in mind. Our main motivation of this paper is the control of inexactness arising from iterative system solves (1.2) in composite–step trust–region SQP methods. However, our assumptions on the inexactness are more general and cover inexact first–order derivative information. The novel aspect of our work is the ability to deal with inexact first–order derivative information or inexact linearized constraint equation solves. Of course, we also allow the inexact solution of trust–region subproblems, which is a standard ingredient of trust–region convergence theories and implementations.

In the context of Newton methods for nonlinear equations and unconstrained optimization, the control of inexactness is relatively well understood. See, e.g., [2, 7, 12, 13, 14, 25]. Generalizations of the inexact Newton method concepts to the local convergence analysis of inexact SQP methods can be found, e.g., in [8, 9, 15, 22, 26]. In [23] global convergence of line-search reduced SQP methods is studied. The influence of inexact problem information on the global convergence of trust–region SQP methods, however, is to our knowledge not yet studied. Our analysis and our assumptions on inexactness are different from [23]. In particular, our bounds on the inexactness do not rely on Lipschitz constants, derivative bounds, and other quantities that are difficult to obtain in practice. Our bounds on the inexactness depend on quantities that are readily available in our algorithms.

We give a global convergence analysis of a class of composite–step trust–region SQP algorithms for (1.1), which are reviewed in [5, § 15.4] and [10, § 4]. In the absence of inexactness our global convergence theory is that of [10]. If all iterates are feasible, i.e., if all iterates satisfy $C(y_k, u_k) = 0$, then our results are related to the convergence analyses in [3, 5] for trust–region methods with inexact function and gradient information for unconstrained optimization.

This paper is organized as follows. In section 2 we will consider the reduced problem $\min f(y(u), u)$ obtained from (1.1) by eliminating the variables y. We will briefly discuss the convergence analyses in [3] and [5, §§ 8.4,10.6] for trust–region methods with inexact function or gradient information for the reduced problem. This will reveal some useful problem information and it will later motivate our assumptions on the inexactness for problem (1.1). Section 3 contains a brief review of the composite–step trust–region SQP algorithms and of their global convergence analyses given in [10]. Our inexact trust–region SQP algorithms and their global convergence analyses will be described in section 4. Assumptions on the inexactness in section 4 are stated in a general way. In section 5 we will discuss how they could be satisfied in an implementation. In the conclusions, section 6, we point to some possible extensions.

We use the following notation. We often set $x = (y, u)$ and use z_y and z_u to represent the subvectors of $z \in \mathbb{R}^n$ corresponding to the y and u components, respectively. The SQP iterates are indexed by k and the symbol of a function with subscript k is used to represent the value of that function at x_k and, possibly, λ_k. For instance, $f_k = f(x_k) = f(y_k, u_k)$. The vector and matrix norms used are the ℓ_2 norms, i.e., $\|\cdot\| = \|\cdot\|_2$. The $l \times l$ identity matrix is denoted by I_l.

2. Trust–region methods for the black–box formulation with inexactness. Under the assumptions of the implicit function theorem, the problem (1.1) can be locally reduced to an unconstrained problem in the variable u. Since the type of inaccuracies we are interested in for (1.1) relate to function and gradient inaccuracies for the reduced problem, it is worthwhile to review existing results on trust–region methods with inexact function and gradient evaluations for unconstrained problems. To simplify this presentation, we impose conditions that ensure that (1.1) is equivalent to an unconstrained problem. We suppose that for all $u \in \mathbb{R}^{n-m}$ the constraint equation $C(y, u) = 0$ has a unique solution y and that $C_y(y, u)$ is invertible for all (y, u) with $C(y, u) = 0$. In this case

the implicit function theorem guarantees the existence of a twice continuously differentiable function $u \mapsto y(u)$ defined through the solution of $C(y,u) = 0$. Instead of (1.1) we can consider the equivalent reduced problem

$$\min \widehat{f}(u) = f(y(u),u). \tag{2.1}$$

This problem is also called the black–box formulation of the optimization problem (1.1) because the solution of $C(y,u) = 0$ is treated as a black–box in the optimization algorithms for (2.1). It can be shown that

$$\nabla \widehat{f}(u) = W(y,u)^T \nabla f(y,u)|_{y=y(u)} = W(y,u)^T \nabla \ell(y,u,\lambda)|_{y=y(u),\lambda=\lambda(u)}, \tag{2.2}$$

where

$$W(y,u) = \begin{pmatrix} -C_y(y,u)^{-1} C_u(y,u) \\ I_{n-m} \end{pmatrix}, \tag{2.3}$$

and $\lambda(u)$ solves $C_y(y(u),u)^T \lambda = -\nabla_y f(y(u),u)$. For details see, e.g., [11, 19].

Now, suppose that the nonlinear equations $C(y,u_k) = 0$ can not be solved exactly for $y_k = y(u_k)$, but that an approximation $\widetilde{y}(u_k)$ of $y_k = y(u_k)$ is computed by applying an iterative method to $C(y,u_k) = 0$. In this case the function $\widehat{f}$ and its gradient can not be evaluated exactly. Gradient computation also requires the solution of a linear system of the form $C_y(y_k,u_k)^T z = -\nabla_y f(y_k,u_k)$; if such systems are solved iteratively, this will introduce another source of inexactness in the gradient. How does one need to control the inexactness in function values and gradients in trust–region methods for (2.1)? The influence of inexact gradient information is analyzed in [3], [5, § 8.4], [35] (for a detailed literature review see [5, p. 296]) and the influence of inexact function evaluations is studied in [5, § 10.6]. We want to ensure that our inexactness assumptions for the trust–region method for (1.1) are compatible with the existing inexactness assumptions for trust–region methods for (2.1) in the case that the SQP iterate (y_k,u_k) satisfies $C(y_k,u_k) = 0$. Therefore we briefly review the theory in [5, §§ 8.4,10.6].

In a trust–region method for the solution of (2.1) one computes an approximate solution of

$$\min_{\|s_u\| \le \Delta_k} \widehat{m}_k(s_u) \stackrel{\text{def}}{=} \widehat{f}_k + \widehat{g}_k^T s_u + \tfrac{1}{2} s_u^T \widehat{H}_k s_u,$$

where $\widehat{g}_k$ is an approximation of $\nabla \widehat{f}(u_k)$ and $\widehat{H}_k$ replaces $\nabla^2 \widehat{f}(u_k)$. The decision about the acceptance of $u_k + (s_u)_k$ as the next iterate and about how to update the trust–region radius is based on the ratio of actual decrease $\widehat{\text{ared}}_k = \widehat{f}(u_k) - \widehat{f}(u_k + (s_u)_k)$ and predicted decrease $\widehat{\text{pred}}_k = \widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$. Let $\eta_2 \in (0,1)$ be the constant so that the trust–region radius is reduced if and only if $\widehat{\text{ared}}_k / \widehat{\text{pred}}_k < \eta_2$ and let $\eta_1 \in (0,\eta_2]$ be the constant so that the step is rejected if and only if $\widehat{\text{ared}}_k / \widehat{\text{pred}}_k < \eta_1$.

In [5, § 8.4] it is shown that if the relative gradient error satisfies

$$\|\widehat{g}_k - \nabla \widehat{f}(u_k)\| / \|\widehat{g}_k\| \le \xi < (1-\eta_2)/2, \tag{2.4}$$

then global convergence of the trust–region algorithm to stationary points can be guaranteed. This accuracy requirement for the gradient approximation is rather weak.

Inexact evaluation of $\widehat{f}$ influences the computation of $\widehat{\text{ared}}_k$. The influence of inexact function evaluations is analyzed in [5, § 10.6]. It is sufficient that

$$\begin{aligned} |f(\widetilde{y}(u_k),u_k) - f(y(u_k),u_k)| &\le \eta_0 \widehat{\text{pred}}_k, \\ |f(\widetilde{y}(u_k+(s_u)_k),u_k+(s_u)_k) - f(y(u_k+(s_u)_k),u_k+(s_u)_k)| &\le \eta_0 \widehat{\text{pred}}_k, \end{aligned} \tag{2.5}$$

where $\eta_0 < \frac{1}{2}\eta_1$. In particular, these accuracy requirements guarantee that if the ratio of actual and predicted decreases indicates acceptance of the step, i.e., if $\widehat{\text{ared}}_k/\widehat{\text{pred}}_k \geq \eta_1$, where $\widehat{\text{ared}}_k$ is computed with the inexact function values, then one still obtains a sufficient decrease $\widehat{f}(u_k) - \widehat{f}(u_k + (s_u)_k) \geq (\eta_1 - 2\eta_0)\widehat{\text{pred}}_k$ in the exact function values. Note also that the accuracy requirement for $f(\widetilde{y}(u_k), u_k)$ depends on the trust-region step $(s_u)_k$, which is not known when $f(\widetilde{y}(u_k), u_k)$ is computed the first time. Therefore, $f(\widetilde{y}(u_k), u_k)$ might have to be recomputed if $\widehat{\text{pred}}_k$ becomes too small to meet the required accuracy requirement. For more details see [5, § 10.6].

3. Trust–Region SQP Methods. In this section we describe the class of composite–step trust–region algorithms assuming exact f and C derivative information and assuming exact solutions of linear systems of the form (1.2). Our representation follows [10, 11]. This section is needed to introduce some basic terminology and notation, as well as to describe later on what can go wrong if f or C derivative information, or linear system (1.2) solutions are inexact.

3.1. The main components of our composite–step trust–region algorithms. Given a local minimizer $x_* = (y_*, u_*)$ for problem (1.1), there exists a Lagrange multiplier λ_* such that the gradient $\nabla \ell(x_*, \lambda_*)$ of the Lagrangian function

$$\ell(y, u, \lambda) = f(y, u) + \lambda^T C(y, u)$$

is zero. If $C_y(x_*)$ is assumed to be nonsingular, then the Lagrange multiplier λ_* is determined by $\nabla_y \ell(x_*, \lambda_*) = \nabla_y f(x_*) + C_y(x_*)^T \lambda_* = 0$, and the first–order necessary optimality conditions can be written as

$$\begin{aligned} \nabla_u \ell(x_*, \lambda(x_*)) &= W(x_*)^T \nabla f(x_*) = 0, \\ \nabla_\lambda \ell(x_*, \lambda(x_*)) &= C(x_*) = 0, \end{aligned} \tag{3.1}$$

where $W(x_*)$ is given by (2.3).

Given approximations $x_k = (y_k, u_k)$ and λ_k for the solution (y_*, u_*) and the corresponding Lagrange multiplier λ_* of (1.1), SQP algorithms compute an (approximate) solution of the quadratic programming (QP) problem

$$\begin{aligned} \min \quad & q_k(s) \stackrel{\text{def}}{=} \ell(x_k, \lambda_k) + \nabla_x \ell(x_k, \lambda_k)^T s + \tfrac{1}{2} s^T H_k s, \\ \text{s.t.} \quad & C_y(x_k) s_y + C_u(x_k) s_u + C(x_k) = 0, \end{aligned} \tag{3.2}$$

where H_k is a symmetric approximation to the Hessian $\nabla^2_{xx} \ell(x_k, \lambda_k)$ of the Lagrangian at (y_k, u_k, λ_k) or the Hessian itself, and then generate a new iterate (y_{k+1}, u_{k+1}) from this QP solution and, possibly, the corresponding Lagrange multiplier λ_{k+1}. To ensure global convergence, a trust–region condition of the form $\|s\| \leq \Delta_k$ is imposed. However, the linear constraints in (3.2) and this trust-region constraint can be incompatible. To deal with the possibility of incompatible constraints, composite–step trust–region algorithms, many of which are reviewed in [5, § 15.4], [10, § 4], split the step s as a sum of two steps s^{n} and s^{t}. We assume that $C_y(x_k)$ is invertible. In this case the step decomposition takes the form

$$s = \begin{pmatrix} s_y \\ s_u \end{pmatrix} = s^{\mathsf{n}} + s^{\mathsf{t}} = \begin{pmatrix} s_y^{\mathsf{n}} \\ 0 \end{pmatrix} + \begin{pmatrix} s_y^{\mathsf{t}} \\ s_u \end{pmatrix}.$$

3.1.1. The quasi–normal step towards feasibility. First, composite–step trust–region algorithms compute a so–called quasi–normal step s_k^{n}, which is responsible to move towards feasibility. Since we assume that $C_y(x_k)$ is invertible, the y–component of s_k^{n} is an approximate solution of

$$\begin{aligned} \min \quad & \|C_y(x_k) s_y^{\mathsf{n}} + C(x_k)\|, \\ \text{s.t.} \quad & \|s_y^{\mathsf{n}}\| \leq \Delta_k \end{aligned} \tag{3.3}$$

and the u–component of s_k^{n} is given by $(s_u^{\mathsf{n}})_k = 0$. Subproblem (3.3) is not solved exactly. A rather coarse solution is sufficient to guarantee basic global convergence. The quasi–normal component s_k^{n} is required to satisfy

$$\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \geq \kappa_1 \|C_k\| \min\{\kappa_2 \|C_k\|, \Delta_k\}, \tag{3.4}$$

where κ_1 and κ_2 are positive constants independent of k.

3.1.2. The tangential step towards optimality. In a second step, composite–step trust–region algorithms compute a so–called tangential step s_k^{t}, which is responsible to move towards optimality but has to maintain linearized feasibility, i.e., has to be in the null-space of the linearized constraints. The tangential step is an approximate solution of

$$\begin{aligned} \min \quad & q_k(s_k^{\mathsf{n}} + s^{\mathsf{t}}) \\ \text{s.t.} \quad & C_y(x_k) s_y^{\mathsf{t}} + C_u(x_k) s_u = 0, \\ & \|s_u\| \leq \Delta_k. \end{aligned} \tag{3.5}$$

From the constraints in (3.5) we see that $s^{\mathsf{t}} = W_k s_u$, where W_k is defined in (2.3). Therefore we can write

$$q_k(s_k^{\mathsf{n}} + s^{\mathsf{t}}) = q_k(s_k^{\mathsf{n}}) + \left(W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\right)^T s_u + \tfrac{1}{2} s_u^T W_k^T H_k W_k s_u \tag{3.6}$$

and pose the problem (3.5) entirely in s_u:

$$\begin{aligned} \min \quad & \widehat{q}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^{\mathsf{n}}) + \left(W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\right)^T s_u + \tfrac{1}{2} (s_u)^T W_k^T H_k W_k (s_u) \\ \text{s.t.} \quad & \|s_u\| \leq \Delta_k. \end{aligned} \tag{3.7}$$

Reduced SQP algorithms do not approximate the Hessian $\nabla^2_{xx} \ell(x_k, \lambda_k)$ but the reduced Hessian $W_k^T \nabla^2_{xx} \ell(x_k, \lambda_k) W_k$. In this case $W_k^T H_k W_k$ in (3.7) is replaced by the reduced Hessian approximation $\widehat{H}_k$ and the term $H_k s_k^{\mathsf{n}}$ is approximated. The details of the latter approximation are not important in our global analysis and we refer to, e.g., [1] for more details.

The tangential step does not need to solve (3.5) or (3.7) exactly. It is sufficient that the tangential component $(s_u)_k$ satisfies a fraction of Cauchy decrease condition associated with the trust–region subproblem (3.7). In other words, $(s_u)_k$ has to provide as much decrease in the quadratic $\widehat{q}_k(s_u)$ as the decrease achieved in the direction $-\nabla \widehat{q}_k(0) = -W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ inside the trust region. It can be proved that such a condition implies

$$\widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \geq \kappa_4 \|W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \min\left\{\kappa_5 \|W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\|,\ \kappa_6 \Delta_k\right\}, \tag{3.8}$$

where κ_4, κ_5, and κ_6 are positive constants independent of k.

3.1.3. Measuring progress and evaluating the trial step. To decide about acceptance of the step $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$ we follow [10] and use the augmented Lagrangian merit function

$$L(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x) = \ell(x, \lambda) + \rho C(x)^T C(x).$$

The decision about acceptance of the step and update of the trust–region radius Δ_k is based on the ratio of actual decrease $\mathrm{ared}(s_k; \rho_k)$, given by

$$\mathrm{ared}(s_k; \rho_k) \stackrel{\text{def}}{=} L(x_k, \lambda_k; \rho_k) - L(x_k + s_k, \lambda_{k+1}; \rho_k), \tag{3.9}$$

and predicted decrease $\mathrm{pred}(s_k; \rho_k)$, given by

$$\mathrm{pred}(s_k; \rho_k) \stackrel{\text{def}}{=} L(x_k, \lambda_k; \rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T (J_k s_k + C_k) + \rho_k \|J_k s_k + C_k\|^2\right), \tag{3.10}$$

where q_k is defined in (3.2), where $J(y,u) = (C_y(y,u) \mid C_u(y,u))$ is the Jacobian of C, and where $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$. Since the tangential step lies in the null space of J_k, we have $J_k s_k^{\mathsf{t}} = C_y(x_k)(s_y^{\mathsf{t}})_k + C_u(x_k)(s_u)_k = 0$, and it can be easily seen that

$$\begin{aligned} \mathrm{pred}(s_k;\rho_k) &= \widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \\ &\quad + q_k(0) - q_k(s_k^{\mathsf{n}}) - (\Delta\lambda_k)^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) \\ &\quad + \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right). \end{aligned} \tag{3.11}$$

Recall that $\widehat{q}_k((s_u)_k) = q_k(s_k^{\mathsf{n}} + W_k(s_u)_k)$ (see (3.7)).

Because of the requirements (3.4), (3.8) on the quasi–normal step and tangential step, respectively, we have that $\widehat{q}_k(0) - \widehat{q}_k((s_u)_k) + \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right) > 0$, provided x_k does not satisfy the first–order necessary optimality conditions (3.1). To ensure that $\mathrm{pred}(s_k;\rho_k)$ is sufficiently positive the penalty parameter ρ_k is increased if necessary. In fact, the penalty parameter ρ_k will be chosen so that

$$\mathrm{pred}(s_k;\rho_k) \geq \frac{\rho_k}{2}\left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right)$$

(see step 2.6 in algorithm 3.1 below).

3.2. Statement of the algorithm. This leads to the following class of trust-region SQP algorithms. They are the same as the trust-region SQP algorithms in [10], but are adapted to our problem context and to our notation.

ALGORITHM 3.1 (Trust-Region SQP Algorithms).

1 Choose x_0 and $\Delta_0 > 0$, and calculate λ_0. Set $\rho_{-1} \geq 1$ and $\epsilon_{tol} > 0$. Choose α_1, η_1, Δ_{min}, Δ_{max}, and $\bar{\rho}$ such that $0 < \alpha_1, \eta_1 < 1$, $0 < \Delta_{min} \leq \Delta_{max}$, and $\bar{\rho} > 0$.

2 For $k = 0, 1, 2, \ldots$ do

2.1 Compute s_k^{n} satisfying (3.13) and (3.4).

2.2 Compute $W_k^T \nabla q_k(s_k^{\mathsf{n}})$.

2.3 If $\|C_k\| + \|W_k^T \nabla q_k(s_k^{\mathsf{n}})\| \leq \epsilon_{tol}$, stop and return x_k as an approximate solution for problem (1.1).

2.4 Compute $(s_u)_k$ satisfying (3.8).

2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

2.6 Update the penalty parameter.

If $\mathrm{pred}(s_k;\rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right)$, then set $\rho_k = \rho_{k-1}$.

Otherwise set

$$\rho_k = \frac{2\left(-\widehat{q}_k(0) + \widehat{q}_k((s_u)_k) - q_k(0) + q_k(s_k^{\mathsf{n}}) + \Delta\lambda_k^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k)\right)}{\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2} + \bar{\rho}.$$

2.7 Compute $(s_y^{\mathsf{t}})_k = -C_y(x_k)^{-1} C_u(x_k)(s_u)_k$ (if not already done in step 2.4).

2.8 If $\mathrm{ared}(s_k;\rho_k)/\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k) < \eta_1$, set

$$\Delta_{k+1} = \alpha_1 \max\{\|s_k^{\mathsf{n}}\|, \|(s_u)_k\|\}$$

and reject s_k.

Otherwise accept s_k and choose Δ_{k+1} such that

$$\max\{\Delta_{min}, \Delta_k\} \leq \Delta_{k+1} \leq \Delta_{max}.$$

2.9 If s_k was rejected set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Otherwise set $x_{k+1} = x_k + s_k$ and let λ_{k+1} be the vector computed in step 2.5.

REMARK 3.2. *In reduced SQP methods one uses*

$$H_k = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{H}_k \end{pmatrix}.$$

In this case $H_k s_k^{\mathsf{n}} = 0$ and steps 2.1 and 2.7 can be merged into a step 2.4a. Instead of executing steps 2.1 and 2.7, one computes in step 2.4a an approximate solution $(s_y)_k$ of

$$\begin{aligned} \min \quad & \|C_y(x_k)s_y + C(x_k)\|, \\ \text{s.t.} \quad & \|s_y\| \le \Delta_k \end{aligned} \tag{3.12}$$

which satisfies (3.13) *and* (3.4)*. In this case $(s_y^{\mathsf{n}})_k$ in steps 2.6 and 2.8 is replaced by $(s_y)_k$.*

3.3. First–order global convergence of the algorithm. Dennis, El–Alem, and Maciel [10] have proved that the class of trust-region SQP algorithms 3.1 is globally convergent. Their convergence theory requires the set of assumptions given below. For all iterations k we assume that $x_k,\ x_k + s_k \in \Omega$, where Ω is an open subset of $\mathbb{R}^n$.

A.1 The functions f, c_i, $i = 1, \ldots, m$ are twice continuously differentiable functions in Ω. Here $c_i(x)$ represents the i–th component of $C(x)$.
A.2 The partial Jacobian $C_y(x)$ is nonsingular for all $x \in \Omega$.
A.3 The functions f, ∇f, $\nabla^2 f$, C, J, $\nabla^2 c_i$, $i = 1, \ldots, m$, are bounded in Ω. The matrix $C_y(x)^{-1}$ is uniformly bounded in Ω.
A.4 The sequences $\{H_k\}$, $\{W_k\}$, and $\{\lambda_k\}$ are bounded.

Dennis, El–Alem, and Maciel [10] show that for a subsequence of the iterates the first–order necessary optimality conditions (3.1) of problem (1.1) are satisfied in the limit.

THEOREM 3.3. *Let assumptions A.1-A.4 hold. The sequences of iterates generated by the trust-region SQP algorithms 3.1 satisfy*

$$\liminf_{k \to \infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| \right) = 0.$$

We remark that inequality (3.4) and A.3 imply the existence of $\kappa_3 > 0$, independent of k, such that

$$\|s_k^{\mathsf{n}}\| \le \kappa_3 \|C_k\|. \tag{3.13}$$

In fact, using $\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| \le \|C_k\|$ and the boundedness of $\{C_y(x_k)^{-1}\}$ we find that

$$\|s_k^{\mathsf{n}}\| \le \|C_y(x_k)^{-1}\| \left(\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| + \|C_k\| \right) \le 2\|C_y(x_k)^{-1}\| \, \|C_k\|.$$

In [10] the condition (3.13) is imposed as an additional condition on the quasi-normal step, because more general quasi–normal steps are allowed.

4. Trust–region SQP methods with inexactness. Now we allow f and C derivative information, as well as linear system (1.2) solutions to be inexact. We assume, however, that the user is able to adjust the level of inexactness. We will investigate how algorithm 3.1 has to be modified to cope with this inexactness. Our aim is to devise conditions on the allowable level of inexactness that meet three criteria. First, we want our conditions to be as weak as possible to admit inexpensive problem information when the iterates (y_k, u_k) are far away from the solution. Secondly, we want

our conditions to be comparable with the conditions on inexact function and gradient information for unconstrained trust-region methods applied to the black–box formulation (2.1), which have been reviewed in section 2. Thirdly, while our conditions on the allowable level of inexactness will be general, we want them to be implementable. In particular, the conditions on the allowable level of inexactness should not depend on derivative bounds, Lipschitz constants, and other quantities that can not be computed in practice.

4.1. The main components of our composite–step trust–region algorithms with inexact problem information.

4.1.1. The quasi–normal step. The assumption (3.4) on the quasi–normal step turns out to be rather weak and can be satisfied using several algorithms that fit into our inexactness framework. This issue will be discussed in section 5.1. Notice also that assumption (3.4) is already expressed in terms of the right hand side C_k and the residual $C_y(x_k)s_y^{\mathsf{n}} + C_k$ of the linear system $C_y(x_k)s_y^{\mathsf{n}} = -C_k$.

4.1.2. The u–component of the tangential step. The computation of the tangential step s_k^{t} allowing inexact information is more complicated. Among other things, we can not assume that s_k^{t} is in the null-space of the linearized constraints. This condition, expressed as $s^{\mathsf{t}} = W_k s_u$ was used repeatedly in sections 3.1.2 and 3.1.3. It will be very useful to discuss the computation of the u–component and the computation of the y–component of the tangential step separately.

If exact derivative information and exact linearized system solves are available, then the u–component of the tangential step is the approximate solution of (3.7). Now, only approximations of $W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ and $W_k^T H_k W_k$ will be available and we compute s_u as the approximate solution of

$$\begin{aligned} \min \quad & \widehat{m}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^{\mathsf{n}}) + \widehat{g}_k^T s_u + \tfrac{1}{2} s_u^T W_k^T \widetilde{H_k} W_k s_u \\ \text{s.t.} \quad & \|s_u\|_2 \leq \Delta_k. \end{aligned} \tag{4.1}$$

In (4.1) the symbol $\widetilde{\ }$ over $W_k^T H_k W_k$ indicates that the reduced Hessian approximation may be inexact. What are the accuracy requirements on $\widehat{g}_k$ and on $W_k^T \widetilde{H_k} W_k$?

If (y_k, u_k) were feasible, i.e., if $C(y_k, u_k) = 0$, then $s_k^{\mathsf{n}} = 0$ (see (3.4)) and $\nabla \widehat{f}(u_k) = W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ (see (2.2)). In this case the theory of [5, § 8.4] for the reduced problem (2.1), which was reviewed in section 2, suggests an accuracy requirement of the form

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \leq \xi_1 \|\widehat{g}_k\| \tag{4.2}$$

with some $\xi_1 \in (0, 1)$ which is related to the parameters in the trust–region algorithm (c.f., (2.4)). We need a slightly stronger condition, namely

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \leq \xi_1 \min\{\|\widehat{g}_k\|, \Delta_k\}, \tag{4.3}$$

where $\xi_1 > 0$. In (4.3) the constant ξ_1 is not tied to the parameters in the trust–region algorithm, in particular we do not need $\xi_1 < 1$, but the absolute error in the reduced gradient approximation must be less than $\|\widehat{g}_k\|$ and Δ_k.

In section 5.2 we show how (4.3) can be enforced in practice, if errors in the reduced gradient are due to inexact linear system solves. There we will see that while (4.3) is slightly stronger than (4.2), the fact that we can give up the restriction $\xi_1 < 1$ makes it preferable from an implementation point of view.

REMARK 4.1. *Imposing the inexactness condition*

$$\|\widehat{g}_k - \nabla \widehat{f}(u_k)\| \leq \xi_1 \min\{\|\widehat{g}_k\|, \Delta_k\}, \tag{4.4}$$

where $\xi_1 > 0$, instead of (2.4) *also gives the standard* $\liminf$ *global convergence result for the unconstrained problem* (2.1)*. This may be seen using the proof in [27, Th. 4.10] and applying* (4.4) *in the estimate for $|\psi_k(s_k) - \nabla f(x_k)^T s_k|$ on page 278 of [27].*

The approximate reduced Hessian has to satisfy

$$(s_u)_k^T W_k^T \widetilde{H_k} W_k (s_u)_k \leq \xi_2 \|(s_u)_k\|^2 \tag{4.5}$$

for some $\xi_2 > 0$ independent of k. If $W_k^T H_k W_k$ is evaluated exactly, then (4.5) is implied by assumption A.4.

The approximate solution $(s_u)_k$ of (4.1) computed in step 2.4 of algorithm 3.1 must provide a fraction of Cauchy decrease on this approximate model $\widehat{m}_k$, i.e.,

$$\widehat{m}_k(0) - \widehat{m}_k((s_u)_k) \;\geq\; \kappa_4 \|\widehat{g}_k\| \min\Big\{\kappa_5 \|\widehat{g}_k\|,\; \kappa_6 \Delta_k\Big\}, \tag{4.6}$$

where, as in (3.8), κ_4, κ_5, and κ_6 are positive constants independent of k. One method to actually compute s_u satisfying (4.6) will be discussed in section 5.3.

4.1.3. Measuring progress, updating the penalty parameter, and evaluating the trial step. The reformulation (3.11) of the predicted decrease $\mathrm{pred}(s_k; \rho_k)$ defined in (3.10) is only valid if s_k^{t} is in the null-space of the linearized constraints. If this is not the case, then

$$\begin{aligned}
\mathrm{pred}(s_k;\rho_k) \;=\; & \widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \\
& + q_k(0) - q_k(s_k^{\mathsf{n}}) - (\Delta\lambda_k)^T (C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) \\
& + \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right) \\
& - (\Delta\lambda_k)^T (r_k^{\mathsf{t}}) - \rho_k \|r_k^{\mathsf{t}}\|^2 - 2\rho_k (r_k^{\mathsf{t}})^T \left(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\right),
\end{aligned}$$

where

$$r_k^{\mathsf{t}} = C_y(x_k)(s_y^{\mathsf{t}})_k + C_u(x_k)(s_u)_k. \tag{4.7}$$

Moreover, the reduced quadratic model $\widehat{q}_k$ defined in (3.2) is now replaced by $\widehat{m}_k$ defined in (4.1). We define

$$\begin{aligned}
\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) \;=\; & \widehat{m}_k(0) - \widehat{m}_k((s_u)_k) + \; q_k(0) - q_k(s_k^{\mathsf{n}}) \\
& - (\Delta\lambda_k)^T (C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) + \; \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right)
\end{aligned} \tag{4.8}$$

and

$$\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k) = -(\Delta\lambda_k)^T (r_k^{\mathsf{t}}) - \rho_k \|r_k^{\mathsf{t}}\|^2 - 2\rho_k (r_k^{\mathsf{t}})^T \left(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\right). \tag{4.9}$$

We now view

$$\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$$

as the quadratic model of the Lagrangian.

This predicted reduction $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ depends only on s_k^{n} and $(s_u)_k$ and can be readily computed. In fact, the quantities $\widehat{m}_k(0)$, $\widehat{m}_k((s_u)_k)$ and $C_y(x_k)(s_y^{\mathsf{n}})_k + C_k$ are typically already computed during the computation of the u–component of the tangential step and the computation of the quasi–normal step, respectively.

Because of the requirements (3.4) and (4.6) on s_k^{n} and $(s_u)_k$, respectively, we have that $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k) + \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right) > 0$, provided (y_k, u_k) does not satisfy the first–order necessary optimality conditions (3.1). We update the penalty parameter ρ_k, if necessary, to ensure sufficient positivity of $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$. See step i2.6 in algorithm 4.3 below.

The evaluation of the step $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$ (we will discuss the computation of $(s_y^{\mathsf{t}})_k$ in a moment) will be based on the ratio $\mathrm{ared}(s_k; \rho_k)/\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$.

4.1.4. The y–component of the tangential step. As we have noted in the previous section, the quadratic model of the Lagrangian is $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$. However, step evaluations are performed based on $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ only. To ensure that the inexactness in the tangential step $(s_y^{\mathsf{t}})_k$ does not dominate this quadratic model, we require that

$$\left|\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)\right| \;\leq\; \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k), \tag{4.10}$$

where $\eta_0 \in (0, 1 - \eta_1)$ is a given constant and η_1 is the parameter in step 2.8 of the trust–region algorithm, and that

$$\|r_k^{\mathsf{t}}\| \;\leq\; \xi_3 \Delta_k \|(s_u)_k\| \tag{4.11}$$

for some constant $\xi_3 > 0$ independent of k. If we estimate $|\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)| \leq \rho_k \|r_k^{\mathsf{t}}\|^2 + (\|\Delta\lambda_k\| + 2\rho_k \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|)\|r_k^{\mathsf{t}}\|$ and insert this upper bound into (4.10), we see that inequality (4.10) is implied by

$$\|r_k^{\mathsf{t}}\| \leq -\sigma + \sqrt{\sigma^2 + \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)/\rho_k}, \tag{4.12}$$

where $\sigma = \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\| + \|\Delta\lambda_k\|/(2\rho_k)$. Inequalities (4.10) and (4.11) are satisfied for the exact solution of $C_y(x_k)(s_y^{\mathsf{t}})_k = -C_u(x_k)(s_u)_k$. The quantity $\|r_k^{\mathsf{t}}\|$ is the residual accuracy of an inexact solution s_y^{t} of $C_y(x_k)s_y^{\mathsf{t}} = -C_u(x_k)(s_u)_k$. Since $s_k^{\mathsf{n}}, (s_u)_k$ and $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ are known, a step $(s_y^{\mathsf{t}})_k$ with (4.10) and (4.11) can be computed.

REMARK 4.2. i. *Condition* (4.10) *is motivated by* (2.5)*. We need to control the accuracy of* $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$*, whereas* (2.5) *controls the accuracy of the actual reduction. However, the effects of both conditions on the ratio of actual and predicted reduction are similar.*

ii. *Notice that* $(s_y^{\mathsf{t}})_k = -C_y(x_k)^{-1}C_u(x_k)(s_u)_k + C_y(x_k)^{-1}r_k^{\mathsf{t}}$ *and that* (4.11) *implies*

$$\|C_y(x_k)^{-1}r_k^{\mathsf{t}}\| \leq \xi_4 \Delta_k \tag{4.13}$$

for some $\xi_4 > 0$*. In other words, it implies that the norm of the residual (tangential) step* $C_y(x_k)^{-1}r_k^{\mathsf{t}}$ *is bounded by a constant time the trust–region radius. If we view* $C_y(x_k)^{-1}r_k^{\mathsf{t}}$ *as a second (tangential) step, or as a spacer (tangential) step, we then identify* (4.13) *as a condition that has already been imposed on steps of such types in the context of global convergence of trust–region algorithms for unconstrained optimization [5,* § *10.4], [6].*

We note that the amount of positivity in $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ is determined by the reductions $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ and $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$. Thus we can allow the more inaccuracy in the $(s_y^{\mathsf{t}})_k$ computation, which typically translates into less expensive $(s_y^{\mathsf{t}})_k$ computation, the larger the linearized feasibility gain $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ achieved by the quasi–normal step *and* the larger the optimality gain $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ achieved by the u–component of the tangential step. In particular, even if $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ is small, but $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ is large (which is likely the case at a point $x_k = (y_k, u_k)$ that is almost feasible, but away from being optimal) the accuracy requirement on $(s_y^{\mathsf{t}})_k$ is rather weak. Our criterion also seems to be closely aligned with the SQP philosophy which allows to trade gains in feasibility for gains in optimality. Another important point worth noting is that inaccuracy in $(s_y^{\mathsf{t}})_k$ does not enter the penalty parameter update. If it would, the penalty parameter might increase faster. Since too large penalty parameters ρ_k can slow down the convergence of SQP methods this is another benefit of our accuracy requirement.

Our initial and somewhat straight forward approach [20, 36] to deal with inaccuracy did not use the split $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$. Rather, the predicted decrease was defined by (3.10). After determination of s_k^{n} satisfying (3.4) we computed a tangential step that, among other conditions, satisfied

$$\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2 \leq \xi_5 \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right) \tag{4.14}$$

with $\xi_5 \in (0,1)$. Thus accuracy of $(s_y^{\mathsf{t}})_k$ depended only on the linearized feasibility gain $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ achieved by the quasi–normal step. Moreover, when

$$\mathrm{pred}(s_k;\rho_{k-1}) < \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2\Big),$$

where $\mathrm{pred}(s_k;\rho_k)$ is given by (3.10), we used the update

$$\rho_k = \frac{2\left(-q_k(0) + q_k(s_k) + \Delta\lambda_k^T(J_k s_k + C_k)\right)}{\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2} + \bar{\rho}. \tag{4.15}$$

The condition (4.14) often lead to very stringent accuracy requirements for $(s_y^{\mathsf{t}})_k$ and the update (4.15) often lead to large increases in the penalty parameter, especially when the current iterate (y_k, u_k) happened to be almost feasible. The approach presented in this paper represents the quadratic model of the Lagrangian as $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)$, separates the computation of the u– and the y–component of the tangential step, bases the accuracy requirement on $(s_y^{\mathsf{t}})_k$ on feasibility *and* optimality gains, and bases the penalty parameter update on quantities that are not contaminated by inaccuracies in $(s_y^{\mathsf{t}})_k$.

4.1.5. Computation of the Lagrange multiplier estimate. Finally, the computation of λ_{k+1} in step 2.5 of the exact trust-region SQP algorithms 3.1 is likely to involve inexact calculations. However, as we have seen in theorem 3.3, global convergence to a stationary point requires only boundedness from the sequence of Lagrange multipliers $\{\lambda_k\}$. This requirement is not only fairly mild from a theoretical point of view, but under assumptions A1–A4 also easy to impose computationally even when inexactness is present.

4.2. Statement of the algorithm. The inexact trust-region SQP algorithms are defined similarly as their exact counter-part, algorithm 3.1, but with steps 2.1 to 2.8 modified to accommodate the inexact calculations discussed above.

ALGORITHM 4.3 (Inexact Trust-Region SQP Algorithms).

1 The same initializations as in step 1 of algorithm 3.1.

2 For $k = 0, 1, 2, \ldots$ do

i2.1 Compute s_k^{n} satisfying (3.13) and (3.4).

i2.2 Compute an approximation $\widehat{g}_k$ to $W_k^T \nabla q_k(s_k^{\mathsf{n}})$ satisfying (4.3).

i2.3 If $\|C_k\| + \|\widehat{g}_k\| \le \epsilon_{tol}$, stop and return $x_k = (y_k, u_k)$ as an approximate solution for problem (1.1).

i2.4 Compute $(s_u)_k$ satisfying (4.6).

i2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

i2.6 Update the penalty parameter.

If $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_{k-1}) \ge \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\Big)$, then set $\rho_k = \rho_{k-1}$.

Otherwise set

$$\rho_k = \frac{2\left(-\widehat{m}_k(0) + \widehat{m}_k((s_u)_k) - q_k(0) + q_k(s_k^{\mathsf{n}}) + \Delta\lambda_k^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k)\right)}{\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2} + \bar{\rho}.$$

i2.7 Compute $(s_y^{\mathsf{t}})_k$ so that the residual vector r_k^{t} satisfies (4.10) and (4.11).

i2.8 Compute $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)$ using (4.8).

If $\mathrm{ared}(s_k;\rho_k)/\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k) < \eta_1$, set

$$\Delta_{k+1} = \alpha_1 \max\{\|s_k^{\mathsf{n}}\|, \|(s_u)_k\|\}$$

and reject s_k.
Otherwise accept s_k and choose Δ_{k+1} such that

$$\max\{\Delta_{min}, \Delta_k\} \leq \Delta_{k+1} \leq \Delta_{max}.$$

i2.9 The same step and multiplier updates as in step 2.9 of algorithm 3.1.

REMARK 4.4. *In reduced SQP methods where $H_k s_k^{\mathsf{n}} = 0$ the algorithm can be slightly reorganized to save one linear system solve with system matrix $(C_y)_k$. See also Remark* 3.2. *Steps 2.1 and 2.7 can be merged into a step 2.4a. Instead of executing steps 2.1 and 2.7, one computes in step 2.4a an approximate solution $(s_y)_k$ of* (3.12) *which satisfies* (3.13) *and* (3.4)*. In this case $(s_y^{\mathsf{n}})_k$ is replaced by $(s_y)_k$ in the remaining steps of the algorithm and $(s_y^{\mathsf{t}})_k = 0$.*

4.3. First–order global convergence of the algorithm. The global convergence property of the inexact trust-region SQP algorithms 3.1 is stated in the following theorem.

THEOREM 4.5. *Let assumptions A.1-A.4 hold. The sequences of iterates generated by the inexact trust-region SQP algorithms* 4.3 *satisfy*

$$\liminf_{k\to\infty} \left(\|\widehat{g}_k\| + \|C_k\| \right) = 0. \tag{4.16}$$

Furthermore, we have

$$\liminf_{k\to\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| \right) = 0. \tag{4.17}$$

Proof. The proof of (4.16) follows the convergence analysis given in [10] with the predicted decrease used there always replaced by $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ as defined in (4.8). Only a very few steps in the convergence analysis change and we will review them in detail.

The first modification concerns the relationship between the size of the step s_k and the trust–region radius Δ_k. The convergence analysis requires that

$$\|s_k\| \leq \kappa_7 \Delta_k$$

and, if s_k is rejected, that

$$\Delta_{k+1} \geq \kappa_8 \|s_k\|.$$

In our inexact trust–region SQP algorithms the first inequality follows from the trust–region constraints in (3.3), (4.1), and from (4.11) and assumption A.3. The second inequality is a consequence of the update of the trust–region radius in i2.8.

The second modification is in the estimates of the difference between actual decrease and predicted decrease. Since $\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$ is different from zero, the upper bounds on the difference between actual and predicted decreases given in [10, L. 7.4, 7.5] are now different. We will be able to show

$$\begin{aligned} &|\mathrm{ared}(s_k; \rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)| \\ &\quad \leq \kappa_9 \Delta_k \|s_k\| + \kappa_{10} \rho_k \|s_k\|^3 + \kappa_{11} \rho_k \|s_k\|^2 \|C_k\| \end{aligned} \tag{4.18}$$

instead of [10, L. 7.4], and

$$\left|\mathrm{ared}(s_k; \rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)\right| \leq \kappa_{12} \rho_k \Delta_k \|s_k\| \tag{4.19}$$

instead of [10, L. 7.5].

The estimates (4.18) and (4.19) can be verified as follows. Using the definitions (4.8) and (4.9) we can see that

$$\begin{aligned}
&\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k) \\
&= -\widehat{g}_k^T (s_u)_k - \frac{1}{2}(s_u)_k^T W_k^T \widetilde{H_k W_k} (s_u)_k - \nabla_x \ell_k^T s_k^{\mathsf{n}} - \frac{1}{2} {s_k^{\mathsf{n}}}^T H_k s_k^{\mathsf{n}} \\
&\quad -\Delta\lambda_k^T (J_k s_k + C_k) + \rho_k \left(\|C_k\|^2 - \|J_k s_k + C_k\|^2 \right) .
\end{aligned}$$

With the definition (3.9) of the actual decrease, the previous identity, and $W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k) = W_k^T \nabla q_k(s_k^{\mathsf{n}})$ we obtain

$$\begin{aligned}
&\mathrm{ared}(s_k; \rho_k) - \left(\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k) \right) \\
&= \ell(x_k, \lambda_k) + \rho_k \|C_k\|^2 - \ell(x_{k+1}, \lambda_{k+1}) - \rho_k \|C_{k+1}\|^2 \\
&\quad -\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k) \\
&= \ell(x_k, \lambda_k) - \ell(x_{k+1}, \lambda_k) + \ell(x_{k+1}, \lambda_k) - \ell(x_{k+1}, \lambda_{k+1}) \\
&\quad +(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)^T W_k (s_u)_k + \tfrac{1}{2}(s_u)_k^T W_k^T H_k W_k (s_u)_k + \nabla_x \ell_k^T s_k^{\mathsf{n}} + \tfrac{1}{2} {s_k^{\mathsf{n}}}^T H_k s_k^{\mathsf{n}} \\
&\quad +(\widehat{g}_k - W_k^T \nabla q_k(s_k^{\mathsf{n}}))^T (s_u)_k + \tfrac{1}{2}(s_u)_k^T W_k^T \widetilde{H_k W_k} (s_u)_k - \tfrac{1}{2}(s_u)_k^T W_k^T H_k W_k (s_u)_k \\
&\quad +\Delta\lambda_k^T (J_k s_k + C_k) - \rho_k (\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2) \\
&= -\ell(x_{k+1}, \lambda_k) + q_k(s_k) - q_k(s_k) + \widehat{q}_k((s_u)_k) \\
&\quad +(\widehat{g}_k - W_k^T \nabla q_k(s_k^{\mathsf{n}}))^T (s_u)_k + \tfrac{1}{2}(s_u)_k^T W_k^T \widetilde{H_k W_k} (s_u)_k - \tfrac{1}{2}(s_u)_k^T W_k^T H_k W_k (s_u)_k \\
&\quad +\Delta\lambda_k^T (-C_{k+1} + J_k s_k + C_k) - \rho_k \left(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2 \right) . \qquad (4.20)
\end{aligned}$$

Using Taylor expansion and the definition (3.2) of q_k gives

$$| -\ell(x_{k+1}, \lambda_k) + q_k(s_k)| \le \tfrac{1}{2} \|H_k - \nabla^2_{xx} \ell(x_k + t_k^1 s_k, \lambda_k)\| \, \|s_k\|^2 \qquad (4.21)$$

with some $t_k^1 \in (0,1)$. Using the definitions (3.2) and (3.7) of q_k and $\widehat{q}_k$, respectively, (3.6), and (4.7) we find that

$$\begin{aligned}
&| -q_k(s_k) + \widehat{q}_k((s_u)_k)| \\
&\le \|H_k s_k^{\mathsf{n}} - \nabla_x \ell(x_k, \lambda_k)\| \, \|s_k^{\mathsf{t}} - W_k (s_u)_k\| + \tfrac{1}{2} \|H_k\| \, \|s_k^{\mathsf{t}}\|^2 + \tfrac{1}{2} \|W_k^T H_k W_k\| \, \|(s_u)_k\|^2 \\
&\le \|H_k s_k^{\mathsf{n}} - \nabla_x \ell(x_k, \lambda_k)\| \, \|C_y(x_k)^{-1}\| \, \|r_k^{\mathsf{t}}\| + \tfrac{1}{2} \|H_k\| \, \|s_k^{\mathsf{t}}\|^2 + \tfrac{1}{2} \|W_k^T H_k W_k\| \, \|(s_u)_k\|^2 . \qquad (4.22)
\end{aligned}$$

With

$$\|s_k^{\mathsf{t}}\| \le \|s_k^{\mathsf{t}} - W_k (s_u)_k\| + \|W_k (s_u)_k\| \le \|C_y(x_k)^{-1}\| \, \|r_k^{\mathsf{t}}\| + \|W_k\| \, \|(s_u)_k\|$$

and (4.11), equation (4.22) implies

$$\begin{aligned}
&| -q_k(s_k) + \widehat{q}_k((s_u)_k)| \\
&\le \xi_3 \|H_k s_k^{\mathsf{n}} - \nabla_x \ell(x_k, \lambda_k)\| \, \|C_y(x_k)^{-1}\| \, \Delta_k \|(s_u)_k\| \\
&\quad +\tfrac{1}{2} \|H_k\| \, \left(\xi_3^2 \|C_y(x_k)^{-1}\|^2 \Delta_k^2 + 2\xi_3 \|W_k\| \, \|C_y(x_k)^{-1}\| \Delta_k + \|W_k\|^2 \right) \|(s_u)_k\|^2 \\
&\quad +\tfrac{1}{2} \|W_k^T H_k W_k\| \, \|(s_u)_k\|^2 . \qquad (4.23)
\end{aligned}$$

The inequalities (4.3) and (4.5) give

$$
\begin{aligned}
&(\widehat{g}_k - W_k^T \nabla q_k(s_k^{\mathsf{n}}))^T (s_u)_k + \tfrac{1}{2}(s_u)_k^T W_k^T \widetilde{H_k} W_k (s_u)_k - \tfrac{1}{2}(s_u)_k^T W_k^T H_k W_k (s_u)_k \\
&\leq \xi_1 \Delta_k \|(s_u)_k\| + \frac{1}{2}(\xi_2 + \|W_k^T H_k W_k\|)\|(s_u)_k\|^2. \qquad (4.24)
\end{aligned}
$$

Using Taylor expansion we obtain

$$
\begin{aligned}
&\Delta\lambda_k^T \left(-C_{k+1} + J_k s_k + C_k\right) - \rho_k \left(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2\right) \\
&= -\tfrac{1}{2}\sum_{i=1}^m (\Delta\lambda_k)_i s_k^T \nabla^2 c_i(x_k + t_k^2 s_k) s_k \\
&\quad -\rho_k \Big(\sum_{i=1}^m c_i(x_k + t_k^3 s_k)(s_k)^T \nabla^2 c_i(x_k + t_k^3 s_k)(s_k) \\
&\quad +(s_k)^T J(x_k + t_k^3 s_k)^T J(x_k + t_k^3 s_k)(s_k) - (s_k)^T J(x_k)^T J(x_k)(s_k) \Big),
\end{aligned}
$$

where $t_k^2, t_k^3 \in (0,1)$. Now we expand $c_i(x_k + t_k^3 s_k)$ around $c_i(x_k)$. This expansion and assumptions A.1–A.4 give

$$
\begin{aligned}
&\Delta\lambda_k^T \left(-C_{k+1} + J_k s_k + C_k\right) - \rho_k \left(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2\right) \\
&\leq \kappa_{10}\rho_k \|s_k\|^3 + \kappa_{11}\rho_k \|s_k\|^2 \|C_k\|. \qquad (4.25)
\end{aligned}
$$

If we insert (4.21)–(4.25) into (4.20) and use assumptions A.3, A.4 and (4.11), we arrive at the desired estimate (4.18) for some positive constants κ_9, κ_{10}, and κ_{11}. Inequality (4.19) is then a direct consequence of inequality (4.18) and the fact that $\rho_k \geq 1$.

We can now bound the difference between the actual and predicted decreases in the inexact context. Combining (4.18) with (4.10), yields

$$
\begin{aligned}
&|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)| \\
&\leq \left|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right| + \left|\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right| \\
&\leq \kappa_9 \Delta_k \|s_k\| + \kappa_{10}\rho_k \|s_k\|^3 + \kappa_{11}\rho_k \|s_k\|^2 \|C_k\| + \eta_0 \left|\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)\right|. \qquad (4.26)
\end{aligned}
$$

Similarly, combining (4.19) with (4.10), gives

$$
|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)| \leq \kappa_{12}\rho_k \Delta_k \|s_k\| + \eta_0 \left|\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)\right|. \qquad (4.27)
$$

The estimates (4.26) and (4.27) are used in the analysis only when rejection occurs in step i2.8. If s_k is rejected, we know that

$$
0 < 1 - \eta_1 \leq \left| \frac{\mathrm{ared}(s_k;\rho_k)}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)} - 1 \right|,
$$

which in our inexact context implies

$$
1 - \eta_1 \leq \left| \frac{\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)} \right| + \eta_0.
$$

Thus, when the estimate (4.19) is required, we obtain

$$
0 < 1 - \eta_0 - \eta_1 \leq \frac{\kappa_{12}\rho_k \Delta_k \|s_k\|}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k;\rho_k)},
$$

and the analysis in [10] remains unchanged except for the fact that a different lower bound $1 - \eta_0 - \eta_1 \in (0,1)$ is used. A similar bound is obtained when the estimate is given by (4.18).

The proof of (4.17) follows from the conjunction of (4.16) with (4.3) and (3.13). □

5. Implementation in the presence of inexactness. In this section we discuss how the requirements on the approximate reduced gradient and on the step components introduced in section 4 can be satisfied in practice. Our discussion leads to an implementable version of algorithm 4.3. However, other implementations are possible. This section is not meant to be comprehensive. Rather it is meant to support our claim made in the introduction and at the beginning of section 4 that our conditions on the allowable level of inexactness are general but implementable.

5.1. Computation of the quasi–normal component. The quasi–normal component s_k^{n} is an approximate solution of the trust–region subproblem (3.3) and it is required to satisfy the condition (3.4).

If $\|(s_y^{\mathsf{n}})_k\| \leq \Delta_k$ satisfies the fraction of Cauchy decrease condition

$$\begin{aligned} &\tfrac{1}{2}\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \\ &\leq \min\left\{\tfrac{1}{2}\|C_y(x_k)s + C_k\|^2 \;:\; s = -tC_y(x_k)^T C_k\,,\ \|s\| \leq \Delta_k\right\}\,, \end{aligned} \tag{5.1}$$

then a result due to Powell [29, Th. 4] (see also [5, § 6.3], [27, L. 4.8]) shows that (3.4) is satisfied. The papers [17], [32] describe two iterative methods based on Krylov subspaces for the computation of steps $(s_y^{\mathsf{n}})_k$ satisfying

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \geq \beta\Big(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_* + C_k\|^2\Big),$$

where $(s_y^{\mathsf{n}})_*$ is the solution of (3.3). In particular these steps also satisfy (3.4). The iterative method in [32] uses a restart technique that allows specification of storage limitations by the user, which is important for large scale problems. The iterative methods in [17] and in [32] require the evaluation of $C_y(x_k)v$ and $C_y(x_k)^T u$ for given v and u.

For some applications, the evaluation of matrix–vector products $C_y(x_k)^T v$ is more expensive than the evaluation of $C_y(x_k)v$, and therefore it may be more efficient to use methods that avoid the use of $C_y(x_k)^T v$. In this case one can apply nonsymmetric Krylov subspace methods based on minimum residual approximations, such as GMRES(l) [30]. In the context of nonlinear system solving the use of such methods is described e.g. in [2]. In that context, trust–region subproblems of the type (3.3) also have to be solved and the solvers in [2] can be applied in our situation as well. If GMRES(l) is used to project the quasi-normal step problem (3.3) onto the l–dimensional Krylov subspace and if

$$\frac{1}{2}C_k^T\Big(C_y(x_k)^T + C_y(x_k)\Big)C_k \geq \beta\|C_k\|^2 \tag{5.2}$$

holds with $\beta > 0$, then (3.4) is satisfied. The condition (5.2) is implied by the positive definiteness of the symmetric part of $C_y(x_k)$, a condition also important for the convergence of nonsymmetric Krylov subspace methods. A proof of this result and more details concerning the use of these methods can be found in [36].

Finally, we can also use the following simple procedure. Compute $\widetilde{s}_k^{\mathsf{n}}$ such that $\|C_y(x_k)\widetilde{s}_k^{\mathsf{n}} + C_k\| \leq \zeta\|C_k\|$, where $\zeta < 1$, and then scale this step back into the trust region, i.e., set

$$s_k^{\mathsf{n}} = \begin{pmatrix} \xi_k\widetilde{s}_k^{\mathsf{n}} \\ 0 \end{pmatrix}, \quad \text{where } \xi_k = \begin{cases} 1 & \text{if } \|\widetilde{s}_k^{\mathsf{n}}\| \leq \Delta_k, \\ \Delta_k/\|\widetilde{s}_k^{\mathsf{n}}\| & \text{otherwise.} \end{cases}$$

The step s_k^{n} also satisfies (3.4) (see [36]).

5.2. Computation of an approximate reduced gradient. We show how (4.3) can be enforced, if errors in the reduced gradient are due to inexact linear system solves.

If we set $d = H_k s_k^{\mathsf{n}} + \nabla_x \ell_k$ and denote the y– and u–component of d by d_y and d_u, respectively, then $W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k) = -(C_u)_k^T (C_y)_k^{-T} d_y + d_u$. We suppose that the inexactness in the computation of $W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ is due to the use of an iterative solver for the linear system $(C_y)_k^T z = -d_y$. More precisely, we assume that

$$\widehat{g}_k = (C_u)_k^T \widehat{z} + d_u, \tag{5.3}$$

where $\widehat{z}$ satisfies

$$(C_y)_k^T \widehat{z} = -d_y - e \tag{5.4}$$

with a residual error e. The following result is easy to prove.

LEMMA 5.1. *If $\widehat{g}_k$ is given by* (5.3), (5.4) *and if*

$$\|e\| \le \min\{c_1 \|(C_u)_k^T \widehat{z} + d_u\|, c_2 \Delta_k\}, \tag{5.5}$$

where $c_1, c_2 > 0$ are given, then (4.3) *is satisfied with $\xi_1 = \max\{c_1, c_2\} \|(C_u)_k^T (C_y)_k^{-T}\|$.*

Proof. Equations (5.3), (5.4) imply $\widehat{g}_k = -(C_u)_k^T (C_y)_k^{-T}(d_y + e) + d_u$ and

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| = \|(C_u)_k^T (C_y)_k^{-T} e\| \le \|(C_u)_k^T (C_y)_k^{-T}\| \, \|e\|.$$

Hence, using (5.3), (5.5),

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \le \|(C_u)_k^T (C_y)_k^{-T}\| \, \min\{c_1 \|\widehat{g}_k\|, c_2 \Delta_k\}$$

which yields the desired estimate. ∎

At first sight the inequality (5.5) seems impractical since both e and $(C_u)_k^T \widehat{z} + d_u$ depend on $\widehat{z}$. However, (5.5) can be enforced if an iterative method for the solution of $(C_y)_k^T z = -d_y$ is used and matrix–vector products of the form $(C_u)_k^T v$ for a given v can be easily computed. The latter is the case for many control problems. In fact, let $z^{(j)}$ be the jth iterate in the solution method for $(C_y)_k^T z = -d_y$ and let $e^{(j)} = -d_y - (C_y)_k^T z^{(j)}$ be the corresponding residual. If $(C_u)_k^T z^{(j)}$ can be easily computed, then we can monitor $\|(C_u)_k^T z^{(j)} + d_u\|$ and we can truncate the iterative linear system solver when

$$\|e^{(j)}\| \le \min\{c_1 \|(C_u)_k^T z^{(j)} + d_u\|, c_2 \Delta_k\}.$$

Note that the truncation criterion (5.5) for the iterative linear system solver is only applicable, because $\xi_1 > 0$ in (4.3) is not restricted. If it were required that $\xi_1 \in (0, 1)$, say, then we would need an estimate for $\|(C_u)_k^T (C_y)_k^{-T}\|$. Thus, while (4.3) is slightly stronger than (4.2), the fact that we can give up the restriction $\xi_1 < 1$ makes it preferable from an implementation point of view.

5.3. Computation of the u–component of the tangential component. An approximate solution s_u of (4.1) that satisfies (4.6) can be computed, e.g., using the conjugate gradient (cg) method with a modification as suggested by Steihaug [33] and Toint [34]. Here the cg method with starting value $s_u = 0$ is applied to the minimization of $\widehat{m}_k$. The conjugate gradient method is stopped if an approximate minimum of the quadratic model $\widehat{m}_k$ is reached, if negative curvature is detected, or if the iterates leave the trust–region bound. The first iterate in the Steihaug–Toint cg method is the Cauchy–step for the $\widehat{m}_k$ and therefore (4.6) is satisfied for the first iterate of the Steihaug–Toint cg method. If $W_k^T H_k W_k$ can be applied exactly, which is the case in a reduced SQP method where $W_k^T H_k W_k = \widehat{H}_k$, then the conjugate gradient method ensures that $\widehat{m}_k$ decreases monotonically and (4.6) remains satisfied for all Steihaug–Toint cg iterates. If $W_k^T H_k W_k$ is applied inexactly, then one has to compare the function values $\widehat{m}_k$ at the first Steihaug–Toint cg iterate s_u^1 and at the final Steihaug–Toint cg iterate s_u^f. If $\widehat{m}_k(s_u^f) \le \widehat{m}_k(s_u^1)$, then $(s_u)_k = s_u^f$; otherwise $(s_u)_k = s_u^1$.

5.4. Computation of the y–component of the tangential component. In section 4.1.4 we have already shown that (4.10), (4.11) are satisfied if $(s_y^{\mathsf{t}})_k$ satisfies $C_y(x_k)s_y^{\mathsf{t}} = -C_u(x_k)(s_u)_k + r_k^{\mathsf{t}}$ with residual

$$\|r_k^{\mathsf{t}}\| \;\leq\; \min\left\{\xi_3\Delta_k\|(s_u)_k\|, -\sigma + \sqrt{\sigma^2 + \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)/\rho_k}\right\}, \tag{5.6}$$

where $\sigma = \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\| + \|\Delta\lambda_k\|/(2\rho_k)$. Note that all quantities on the right hand side of (5.6) are known by the time $(s_y^{\mathsf{t}})_k$ needs to be computed.

6. Conclusions. In this paper we have extended the design of a class of composite–step trust–region SQP algorithms and their convergence theory to allow the use of inexact first–order derivative information or the use of inexact linearized constraint equation solves. The challenge was the formulation of accuracy requirements that are sufficient to guarantee global convergence to a point satisfying the first–order optimality conditions, but at the same time can be implemented in a practical algorithm and are not overly stringent. Our accuracy requirements are based on the structure of the composite–step trust–region SQP algorithms and they follow the SQP philosophy which allows to trade gains in feasibility for gains in optimality. The main motivation of this paper is the control of inexactness arising from iterative system solves (1.2) in trust–region SQP methods. This is important, e.g., for the solution of discretized optimal control problems governed by partial differential equations. However, our assumptions on the inexactness are not based on this particular source of inexactness and are applicable more broadly.

We focused on a specific class of problems (1.1) and on a limited class of algorithms to enhance the clarity of our presentation. An extension of our analysis of the influence of inexact first–order derivative information or the use of inexact linearized constraint equation solves to a broader range of problems and global SQP algorithms is useful. Some extensions are rather straight forward, although tedious. For example, we believe our analysis can be generalized to the affine–scaling interior–point trust–region SQP algorithms in [11], which tackle problems (1.1) with additional simple bounds on u. In fact, the predecessor [20] of this paper contains many of the technical details of such an extension, although the assumptions on the inexactness made in [20] are stronger than those in this paper.

Acknowledgments. The authors would like to thank the two anonymous referees and the associate editor for their constructive comments on the first version of this paper, which lead to significant improvements in the presentation.

REFERENCES

[1] L. T. BIEGLER, J. NOCEDAL, AND C. SCHMID, *A reduced Hessian method for large–scale constrained optimization*, SIAM J. on Optimization, 5 (1995), pp. 314–347.
[2] P. N. BROWN AND Y. SAAD, *Convergence theory of nonlinear Netwon–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.
[3] R. G. CARTER, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.
[4] E. M. CLIFF, M. HEINKENSCHLOSS, AND A. SHENOY, *Airfoil design by an all–at–once method*, International Journal for Computational Fluid Mechanics, 11 (1998), pp. 3–25.
[5] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust–Region Methods*, SIAM, Philadelphia, 2000.
[6] A. R. CONN, L. N. VICENTE, AND C. VISWESWARIAH, *Two–step algorithms for nonlinear optimization with structured applications*, SIAM J. Optimization, 9 (1999), pp. 924–947.
[7] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[8] R. S. DEMBO AND U. TULOWITZKI, *Local convergence analysis for successive inexact quadratic programming methods*, Tech. Report SOM Series B # 78, Department of Computer Science, Yale University, 1984.
[9] ———, *Sequential truncated quadratic programming*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 83–101.

[10] J. E. DENNIS, M. EL-ALEM, AND M. C. MACIEL, *A global convergence theory for general trust–region–based algorithms for equality constrained optimization*, SIAM J. Optimization, 7 (1997), pp. 177–207.
[11] J. E. DENNIS, M. HEINKENSCHLOSS, AND L. N. VICENTE, *Trust–region interior–point algorithms for a class of nonlinear programming problems*, SIAM J. Control and Optimization, 36 (1998), pp. 1750–1794.
[12] J. E. DENNIS AND H. F. WALKER, *Inaccuracy in quasi-Newton methods: Local improvement theorems*, in Mathematical Programming Study 22: Mathematical Programming at Oberwolfach II, North–Holland, Amsterdam, 1984, pp. 70–85.
[13] P. DEUFLHARD, *Global inexact Newton methods for very large scale nonlinear problems*, Impact of Computing in Science and Engineering, 4 (1991), pp. 366–393.
[14] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.
[15] R. FONTECILLA, *On inexact quasi–Newton methods for constrained optimization*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 102–118.
[16] O. GHATTAS AND J.-H. BARK, *Optimal control of two– and three–dimensional Navier–Stokes flow*, Journal of Computational Physics, (1997), pp. 231–244.
[17] G. H. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numerische Mathematik, 59 (1991), pp. 561–580.
[18] M. D. GUNZBURGER, L. S. HOU, AND T. P. SVOBOTNY, *Optimal control and optimization of viscous, incompressible flows*, in Incompressible Computational Fluid Dynamics, M. D. Gunzburger and R. A. Nicolaides, eds., Cambridge, New York, 1993, Cambridge University Press, pp. 109–150.
[19] M. HEINKENSCHLOSS, *Projected sequential quadratic programming methods*, SIAM J. Optimization, 6 (1996), pp. 373–417.
[20] M. HEINKENSCHLOSS AND L. N. VICENTE, *Analysis of inexact trust–region interior–point SQP algorithms*, Tech. Report TR95–18, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892, 1995. `http://www.caam.rice.edu/~heinken/papers/Papers.html`.
[21] K. ITO AND K. KUNISCH, *Augmented Lagrangian–SQP methods for nonlinear optimal control problems of tracking type*, SIAM J. Control and Optimization, 34 (1996), pp. 874–891.
[22] S. ITO, C. T. KELLEY, AND E. W. SACHS, *Inexact primal–dual interior–point iteration for linear programs in function spaces*, Computational Optimization and Applications, 4 (1995), pp. 189–201.
[23] H. JÄGER AND E. W. SACHS, *Global convergence of inexact reduced SQP-methods*, Optimization Methods and Software, 7 (1997), pp. 83–110.
[24] A. JAMESON, L. MARTINELLI, AND N. A. PIERCE, *Optimum aerodynamic design using the Navier-Stokes equations*, Theoretical and Computational Fluid Dynamics, 10 (1998), pp. 213–237.
[25] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
[26] F. LEIBFRITZ AND E. W. SACHS, *Inexact SQP interior–point methods and large–scale control problems*, SIAM J. Control and Optimization, 38 (2000), pp. 272–293.
[27] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming, The State of The Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer Verlag, Berlin, Heidelberg, New-York, 1983, pp. 258–287.
[28] S. OMATU AND J. H. SEINFELD, *Distributed Parameter Systems. Theory and Applications*, Oxford University Press, Oxford, New-York, Toronto, 1989.
[29] M. J. D. POWELL, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Boston, New-York, London,..., 1975, Academic Press, pp. 1–27.
[30] Y. SAAD AND M. H. SCHULTZ, *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
[31] V. SCHULZ AND G. WITTUM, *Multigrid optimization methods for stationary parameter identification problems in groundwater flow*, in Multigrid Methods V, Berlin, Heidelberg, New York, 1997, Springer Verlag. http://dom.ica3.uni-stuttgart.de/~volker/papers.html.
[32] D. C. SORENSEN, *Minimization of a large scale quadratic function subject to an spherical constraint*, SIAM J. Optimization, 7 (1997), pp. 141–161.
[33] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[34] P. L. TOINT, *Towards an efficient sparsity exploiting newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–87.
[35] ———, *Global convergence of a class of trust-region methods for nonconvex minimization in Hilbert space*, IMA Journal of Numerical Analysis, 8 (1988), pp. 231–252.
[36] L. N. VICENTE, *Trust–Region Interior–Point Algorithms for a Class of Nonlinear Programming Problems*, PhD thesis, Rice University, Department of Computational and Applied Mathematics, 1996.